UNIVERSIDAD DE LA LAGUNA

Localización simple de servicios deseados y no deseados en redes con múltiples criterios

Autor: Colebrook Santamaría, Marcos

Director: Joaquín Sicilia Rodríguez

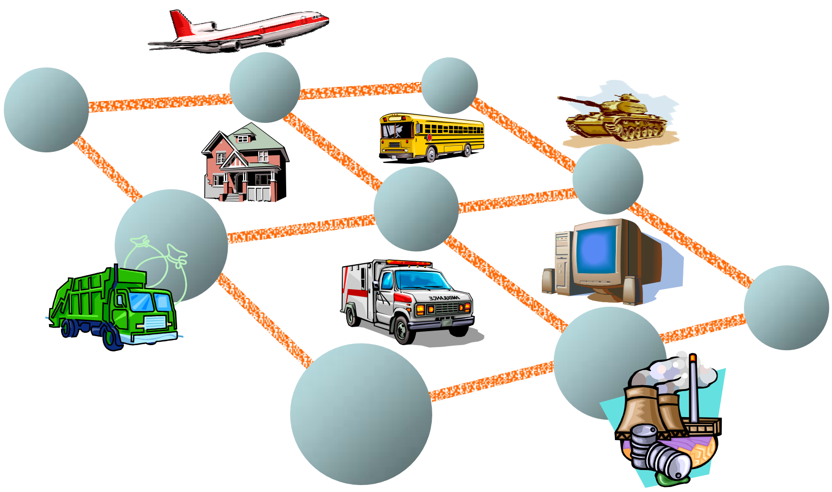
Departamento de Estadística, Investigación Operativa y Computación

UNIVERSIDAD DE LA LAGUNA

Departamento de Estadística, Investigación Operativa y Computación

Localización simple de servicios deseados y no deseados en redes con múltiples criterios

Desirable and undesirable single facility location on networks with multiple criteria



Memoria de Tesis presentada por

Marcos Colebrook Santamaría

para optar al grado de

Doctor por la Universidad de La Laguna

con

Mención de Doctorado Europeo

2003

DR. D. JOAQUÍN SICILIA RODRÍGUEZ, CATEDRÁTICO DEL DEPARTAMENTO DE ESTADÍSTICA, INVESTIGACIÓN OPERATIVA Y COMPUTACIÓN DE LA UNIVERSIDAD DE LA LAGUNA.

CERTIFICO: Que la presente memoria titulada *"Localización simple de servicios deseados y no deseados en redes con múltiples criterios (Desirable and undesirable single facility location on networks with multiple criteria)"* ha sido realizada bajo mi dirección por D. Marcos Colebrook Santamaría, constituyendo su Tesis Doctoral para optar al grado de Doctor por la Universidad de La Laguna.

Y para que conste, en cumplimiento de la legislación vigente a los efectos que haya lugar, firmo la presente.

La Laguna, a 24 de marzo de 2003.

Dedication

To my dear mother,
and in memory of my beloved father,
who encouraged my interest in science and engineering.

Índice

Capítulo VII: Problemas de localización de servicios no deseados en redes multicriterio

Lista de figuras, tablas y algoritmos (español)

Figuras

Tablas

Algoritmos

Contents

Chapter VII: Undesirable facility location problems on multicriteria networks

List of figures, tables and algorithms

Figures

Tables

Algorithms

Agradecimientos (español)

Estoy realmente en deuda con el Dr. Joaquín Sicilia Rodríguez por no sólo haber sido mi director sino también por las improvisadas lecciones en Teoría de Grafos, Localización y Optimización Combinatoria, que facilitaron la realización de esta Tesis y mejoraron considerablemente mis conocimientos de estas materias dentro de la Investigación Operativa.

Mis más sincero agradecimiento al Dr. Stefan Nickel, director del *Fraunhofer Institut für Techno– und Wirtschaftsmathematik* (ITWM) en Kaiserslautern (Alemania) por aceptarme como investigador invitado. Asimismo, quiero darle las gracias a todo el personal del ITWM, especialmente a Michael, Patricia, Teresa, Jörg y Holger, por su gentil ayuda durante mi estancia en Kaiserslautern.

Agradezco los servicios de computación proporcionados por el *Centro de Comunicaciones y Tecnologías de la Información* de la *Universidad de La Laguna* (ULL), dirigido por Félix Herrera Priano, y la asistencia técnica de José C. González González.

Debo agradecer también a todos mis queridos compañeros del *Departamento de Estadística, Investigación Operativa y Computación* (ULL) su apoyo permanente, y en especial a José Miguel, Sergio, Antonio, Macu, Rosa y Tere.

Finalmente, quiero hacer un reconocimiento póstumo al Catedrático Dr. Félix Herrera Cabello (miembro del Tribunal de mi Tesina), por infundir la curiosidad y la actitud crítica en la ciencia y la tecnología a todos los que tuvimos el honor de ser sus alumnos.

Prólogo (español)

Desde las más antiguas civilizaciones, los seres humanos han tratado siempre de buscar el mejor sitio donde vivir. Buenas condiciones meteorológicas, situaciones ambientales agradables, abundancia de comida y agua, y la seguridad ante peligros externos son algunas de las opciones más importantes que se consideran a la hora de elegir el mejor lugar donde un nuevo asentamiento debería establecerse.

Hoy en día, nos enfrentamos con innumerables situaciones en las cuales una *entidad* u *objeto* debe ser ubicado dentro de un contexto espacial. Obviamente, siempre demandamos el mejor (óptimo) emplazamiento que cumpla con nuestros propios requerimientos. Este proceso de selección implica algún tipo de toma de decisión sobre un conjunto de diferentes alternativas. En este sentido, escoger la mejor opción involucra en primer lugar la definición de objetivos cuantificables con respecto a los criterios considerados. Posteriormente, se pueden aplicar métodos adecuados para determinar las soluciones óptimas.

Dentro de la Teoría de la Localización, los modelos de localización en redes han tratado normalmente con problemas de un solo criterio, esto es, sobre redes con un peso por nodo y/o una longitud por arista. Sin embargo, para modelar adecuadamente muchos problemas reales el decisor necesita colocar más parámetros en los nodos (demanda, importancia, número de clientes, etc) y en las aristas (longitud, tiempo, costo de tránsito, etc). Es más, muchos autores han argumentado en la bibliografía que existe una gran cantidad de problemas de localización multicriterio/multiobjetivo que no se han investigado todavía, aún cuando este tema ha tomado especial relevancia en las últimas dos décadas. En esta tesis, nos centramos principalmente en modelos de localización en redes con múltiples criterios, considerando varios pesos en los nodos y varias longitudes en las aristas.

Por otro lado, la mayoría de los artículos referentes a problemas de localización tratan el asentamiento de servicios que son considerados como *deseables* por la población circundante, tales como servicios de emergencia (policía/bomberos), centros de educación, hospitales, etc. Sin embargo, debido a la gran inquietud que ha surgido en las últimas décadas sobre temas medioambientales, la localización de servicios *no deseados* (vertederos, plantas químicas, reactores nucleares, etc) está jugando un papel muy importante en la actualidad. Teniendo en cuenta estas inquietudes, hemos analizado algunos modelos de localización de servicios no deseados en redes unicriterio así como en redes multicriterio.

En los siguientes párrafos resumimos los contenidos de esta memoria.

El Capítulo I permite que el lector se familiarice con las definiciones, la notación y la bibliografía en Teoría de la Localización. A este respecto, se revisan más de 150 referencias, desde estudios recopilatorios y libros sobre problemas generales de localización, a artículos más especializados en localización multicriterio sobre redes. Además, para poder describir apropiadamente los modelos desarrollados en esta tesis, se examinan varios esquemas de clasificación para problemas de localización.

El Capítulo II analiza el problema del cent-dian en una red pesada, conexa y no dirigida desde un punto de vista biobjetivo, es decir, considerando dos longitudes (costes) por arista. El problema consiste en localizar un servicio sobre la red que minimice la combinación convexa de la distancia total y de la distancia máxima desde cualquier punto al resto de la red. Usando técnicas de Geometría Computacional, proponemos un algoritmo en tiempo polinomial que determina todos los puntos eficientes de la red. Al final del capítulo se proporcionan algunos resultados computacionales. En colaboración con R.M. Ramos, J. Sicilia y T. Ramos, una parte principal de este capítulo ha sido publicado en *Studies in Locational Analysis* (2000).

En el Capítulo III consideramos el problema de localizar un solo servicio sobre una red con $q \geq 2$ objetivos mediana representados por q conjuntos de pesos (o longitudes) sobre las aristas correspondiendo a cada uno de los objetivos. Cuando $q = 1$, se obtiene el clásico problema 1-mediana donde solamente se consideran los vértices para determinar la localización óptima. El capítulo examina el caso cuando $q \geq 2$, y proporciona un método para determinar el conjunto no-dominado de puntos de localización del servicio. En colaboración con R.M. Ramos, J. Sicilia y T. Ramos, un artículo referente al problema de localización 1-mediana multiobjetivo ha aparecido en *Annals of Operations Research* (1999).

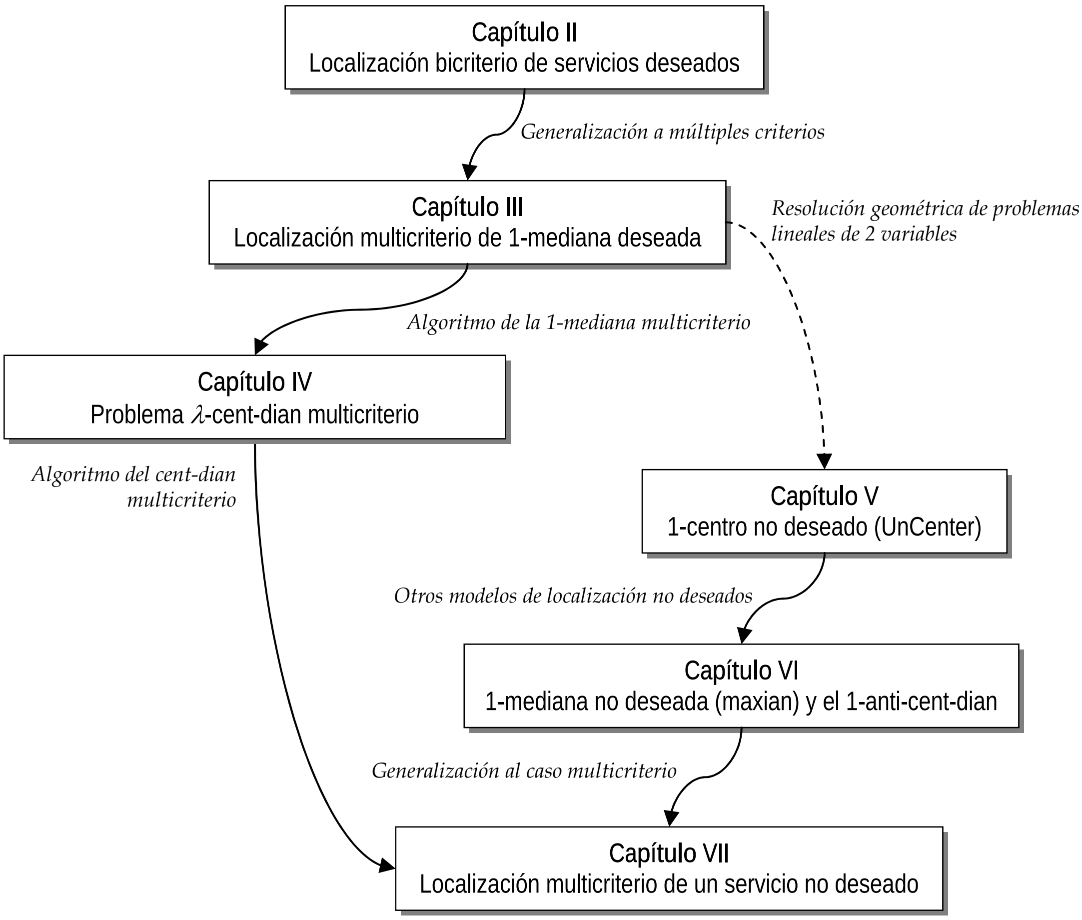
Considerando redes con varios pesos en los nodos y varias longitudes en las aristas, en el Capítulo IV presentamos un algoritmo polinomial para solucionar el problema λ-cent-dian en redes multicriterio. De este modo, podemos obtener fácilmente la solución al problema del centro multicriterio y al problema de la mediana multicriterio, el cual generaliza el modelo presentado en el capítulo anterior.

Trabajos recientes han desarrollado algoritmos eficientes para la localización de un centro no deseado en redes generales con n nodos y m aristas. Aunque la complejidad teórica de estos algoritmos es adecuada, el tiempo de cómputo requerido para conseguir la solución puede disminuirse usando una formulación diferente del modelo y mejorando levemente las cotas superiores. Así, en el Capítulo V presentamos un nuevo algoritmo en $O(mn)$ más simple y computacionalmente más rápido que los anteriores. Se proporcionan resultados computacionales comparando los métodos previos con el algoritmo propuesto. En colaboración con J. Gutiérrez, S. Alonso y J. Sicilia, una parte del contenido de este capítulo ha sido publicada en *Journal of the Operational Research Society* (2002).

El problema de localizar un servicio no deseado en una red para maximizar su distancia total a todos los nodos se trata en el Capítulo VI. Proponemos una nueva cota superior al problema. Asimismo, desarrollamos un algoritmo en tiempo $O(mn)$ que actualiza dinámicamente esta nueva cota superior. Mostramos resultados computacionales en redes de densidad baja y alta, así como en redes planares. Un trabajo en colaboración con J. Gutiérrez y J. Sicilia referente a la nueva cota y al nuevo algoritmo para el problema maxian está aceptado para publicación en *Computers and Operations Research*. En este capítulo también analizamos el problema del anti-cent-dian, el cual consiste en una combinación convexa del problema del centro no deseado y del problema de la mediana no deseada. Se propone un algoritmo eficiente en tiempo $O(mn)$ que mejora un método previo de complejidad $O(mn \log n)$.

El Capítulo VII está dedicado a la localización de servicios no deseados en redes multicriterio. En primer lugar, analizamos los modelos del centro no deseado y de la mediana no deseada, desarrollando resultados básicos que caracterizan las soluciones eficientes. Posteriormente, por medio de una combinación convexa de estas dos últimas funciones, analizamos el problema del λ-anti-cent-dian, presentando un algoritmo que soluciona dicho problema junto con una regla que elimina las aristas ineficientes.

La memoria acaba con algunas conclusiones y observaciones, así como con la bibliografía utilizada. En la siguiente figura, ilustramos la relación entre los diversos capítulos.



Capítulo I (Resumen)

Introducción a la Teoría de Localización

"Las tres cosas más importantes en el negocio inmobiliario son:
localización, localización y localización"
PROVERBIO INMOBILIARIO

I.1 ¿Qué significa "localización"?

En un sentido muy amplio, los problemas de localización consisten en encontrar el sitio adecuado donde uno o más servicios deberían ubicarse, de forma que se optimice (minimice o maximice) algunos criterios específicos, que están usualmente relacionados con la distancia (medida de rendimiento) existente entre los servicios y los punto de demanda (clientes).

Los problemas de localización surgen muy frecuentemente en nuestras vidas. Esto fue ilustrado en una viñeta recogida en el prólogo de Mirchandani y Francis (1990), y que también es el proverbio que encabeza este capítulo. Como dice la señora de la viñeta, los tres principios fundamentales en el negocio inmobiliario son *localización, localización y localización*. La casa ofertada a la pareja es considerada un buen emplazamiento, ya que la distancia de recorrido a los servicios circundantes es insignificante.

Existen cientos de referencias y páginas web en Internet describiendo cómo localizar el mejor lugar para vivir. La mayoría de los requisitos de los potenciales dueños cumplen los siguientes criterios: proximidad del colegio, distancia mínima al lugar de trabajo, acceso rápido al transporte público, servicios médicos/emergencia cercanos y centros comerciales colindantes. El criterio clave parece estar siempre directamente relacionado con la distancia recorrida.

Además de su indiscutible papel en el mercado inmobiliario, la teoría de la localización ha tenido también una gran inquietud en el establecimiento de nuevos negocios privados y en el desarrollo de servicios públicos. Por ejemplo, en el sector privado, los franquiciadores consideran los siguientes criterios, entre otros, como los más destacados en la implantación de una nueva franquicia:

- Información demográfica: densidad y tipo de la población circundante.
- Tráfico y accesibilidad: cantidad de coches y peatones que pasan por la futura franquicia.
- Competidores: ¿quiénes son? ¿dónde están situados?

Salvaneschi (1996), antiguo Presidente de *Blockbuster Video*, Vicepresidente de *McDonald's* y Vicepresidente Superior de *Kentucky Fried Chicken* (tres de las mayores franquicias en el mundo), afirma que la localización es una de los materias más cruciales en el desarrollo de una nueva franquicia.

Asimismo, cuando se trata de establecer un nuevo negocio, los comerciantes han tratado de situarlos tan cerca como fuera posible de los potenciales clientes. Resumimos esta idea básica en la siguiente ley de mercado: *cuanto más cercana esté la oferta a la fuente de demanda, más rentable será el negocio*. Otros problemas de localización de negocios privados surgen también en el establecimento de plantas de producción y ensamblaje, almacenes, nuevas oficinas y centros de distribución.

Por otro lado, el sector público también requiere enfoques óptimos en la localización de servicios de emergencia (ambulancias y estaciones de policía/bomberos), recursos públicos (agua y electricidad), o incluso de servicios no deseados (vertederos, plantas de tratamiento de residuos y reactores nucleares).

Daskin (1995) declaró de forma breve que *"el éxito o fracaso de los servicios privados y públicos depende de las localizaciones elegidas para esos servicios"*. Más aún, en muchas circunstancias las localizaciones resultan ser bastante críticas. Por ejemplo, en la asistencia a personas que sufran ataques de corazón, el mal emplazamiento de las ambulancias provocará un incremento en el tiempo promedio de respuesta, con el incremento asociado en la probabilidad de fallecimiento (Handler y Mirchandani, 1979; Daskin, 1995).

La localización también se aplica en el campo militar, involucrando el emplazamiento de servicios de recursos tales como comedores, almacenes de armas y munición, y suministros médicos. Además, tanto la localización de instalaciones militares como la de almacenes o silos para misiles son consideradas como problemas de localización de servicios no deseados.

La disciplina matemática que estudia los problemas de localización, construye los modelos matemáticos apropiados y deriva los métodos para resolverlos se denomina *Teoría de Localización*. Siendo una rama del marco de la *Investigación Operativa*, esta materia proporciona a los decisores herramientas cuantitativas para encontrar buenas soluciones a problemas de decisión de localización reales. También, la moderna Teoría de Localización ha captado la atención de profesionales como economistas, geógrafos, planificadores regionales y arquitectos, así como investigadores en campos diversos como la *Ingeniería Industrial*, las *Ciencias de la Gestión* o la *Informática*.

Con respecto a la taxonomía de la teoría de la localización, los problemas de localización se ajustan a uno de los siguientes tres tipos:

- *Localización continua*: se permite que las localizaciones estén en cualquier lugar dentro de un espacio *d* dimensional.
- *Localización discreta*: se especifican a priori un número finito de posibles localizaciones en el espacio. A veces también se denomina localización-asignación.
- *Localización en redes*: tipo especial de problemas de localización que se modelan en redes o árboles.

La sección I.4 describirá de forma más precisa la clasificación de los modelos de localización. En esta tesis, nos centramos en problemas de localización en redes. Este tipo de problemas pueden modelar problemas reales de localización en redes fluviales, redes aéreas (pasillos aéreos), redes marítimas (líneas navieras); redes de autopistas, carreteras, avenidas y calles; y redes de comunicaciones y de ordenadores. La literatura en localización en redes está llena de aplicaciones reales intrínsecas. Mencionamos brevemente algunas de ellas:

- Localización de centros de conmutación en una red de comunicaciones para minimizar los costos de transmisión, o la localización de servicios informáticos o programas en una red de ordenadores para minimizar el almacenamiento anual y los costes de transmisión (Handler y Mirchandani, 1979).
- Diseño de la red de tratamiento de aguas de una ciudad o pueblo. Las aguas no tratadas emanan de diferentes fuentes en la ciudad. Se localiza un servicio de tratamiento de aguas para minimizar la longitud total de las tuberías necesarias para conducir el agua no tratada con el servicio de tratamiento (Brandeau y Chiu, 1989).
- Se intenta localizar una unidad de servicio de emergencia en un área rural para minimizar el tiempo máximo de intervención a los centros poblados (Labbé, Peeters y Thisse, 1995).

Tal y como ha sido ilustrado en ejemplos previos, las decisiones sobre problemas reales involucran, en la mayoría de las ocasiones, más de un criterio. Muchos investigadores en varias excelentes recopilaciones y libros, (por ejemplo, Cohon y Shobrys, 1981a,b; Ross y Soland, 1980; Krarup y Pruzan, 1990; Current, Min y Schilling, 1990; Daskin, 1995), han hecho intenso hincapié en la importancia de considerar varios objetivos en el Análisis de Localización. Algunos otros autores van incluso más allá (Erkut y Neuman, 1989; Daskin, 1995; Zhang, 1996), indicando explícitamente no sólo la necesidad de incluir múltiples criterios en los problemas de localización de servicios no deseados, sino también el hecho de que se ha investigado escasamente en este prometedor campo.

La tesis actual está primordialmente centrada en problemas de localización bicriterio y multicriterio de servicios deseados y no deseados sobre redes. Sin embargo, también hemos obtenido nuevos resultados en problemas unicriterio de localización no deseada sobre redes.

A pesar de que la mayoría de estos problemas de localización parecen estar relacionados directamente con el mundo contemporáneo, en realidad algunos de ellos fueron propuestos originalmente hace siglos. Esto se describe en la siguiente sección, donde presentamos una breve reseña histórica, así como una revisión detallada de la bibliografía sobre Análisis de Localización en Redes. Luego, introducimos la notación general y los conceptos básicos en Teoría de Localización. Estos conceptos son usados para describir la clasificación de problemas de localización en la última sección.

I.2 Breve reseña histórica y revisión de la bibliografía

Los problemas de localización han coexistido casi simultáneamente a la vida normal de los seres humanos. Así, nuestros ancestros debían decidir cual era el mejor emplazamiento donde deberían habitar para refugiarse de los peligros, teniendo en cuenta también la cercanía a fuentes de riqueza natural tales como ríos y tierras fértiles.

La primera referencia de la que tenemos constancia data del siglo XVII, cuando el matemático P. Fermat propuso el siguiente problema: *"Dados tres puntos en el plano, encontrar el cuarto punto tal que su distancia al resto es mínima"*.

En 1640, Torricelli observó que este problema tenía una solución geométrica basada en tres círculos circunscritos. En 1834, Heinen demostró que la propiedad de Torricelli no era general. Antes de esto, en 1750 Simpson generalizó el problema para obtener el punto que minimiza la suma pesada de distancias desde los tres puntos dados.

En 1857, Sylvester propuso el siguiente problema descrito en una línea: *"Se requiere encontrar el círculo más pequeño que contenga a un conjunto de puntos en el plano"*. Este es el equivalente de un problema de localización bajo el criterio *minimax*, o a veces descrito como el problema del *centro*.

El origen de la teoría de localización moderna se atribuye a A. Weber (1909), quien incorporó el problema original de Fermat al Análisis de Localización en su influyente tratado sobre la teoría de la localización industrial *"Über den Standort der Industrien"* (Teoría de la localización de industrias), traducido posteriormente por Friedrich (1929). El problema consistía en determinar la localización óptima de una fábrica que debía abastecer a un solo mercado y con dos fuentes diferentes de material. El criterio considerado para tal localización era la minimización de los costos de transporte (distancia a recorrer). Este fue el comienzo de los problemas de localización *minisum*, usualmente conocidos como problemas *mediana* o simplemente problemas de *Weber* (Wesolowsky, 1993).

Todas las referencias de arriba tratan sobre problemas de localización en el plano. Sin embargo, algunos problemas son modelados sobre redes. De este modo, Jordan (1869) obtuvo una caracterización del conjunto mediana de un árbol. Con respecto a los problemas de localización sobre redes generales, debemos hacer mención a Hakimi (1964), quien introdujo los problemas de la mediana y el centro sobre redes pesadas, y de esta forma, su trabajo inicial estableció los cimientos para el desarrollo de sucesivos problemas de localización en redes.

La bibliografía en Análisis de Localización es extremadamente grande y bastante entrelazada. Uno de los primeros y más extensos compendios es debido a Domschke y Drexl (1985), quienes recopilaron una bibliografía de más de 1800 artículos. En un libro más reciente, Drezner (1995) aporta más de 1200 referencias. Trevor Hale (1998) mantiene una página web con una lista de más de 3000 referencias en ciencia de la localización, localización de servicios y trabajos relacionados. ¡Y este número sigue aumentando!

A continuación, citamos algunas recopilaciones, estudios y libros interesantes sobre problemas de localización.

I.2.1 Estudios, recopilaciones y libros sobre problemas de localización

En esta sección podemos destacar las recopilaciones y estudios de Francis, McGinnis y White (1983), Hansen, Peeters y Thisse (1983), Hansen, Labbé y Thisse (1987a), Brandeau y Chiu (1989), Eiselt (1992), Chhajed, Francis y Lowe (1993), Marsh y Schilling (1994), Eiselt y Laporte (1995), Labbé (1998), Hale (1999) y Drezner (2002).

Asimismo, existen varios números especiales en revistas de investigación prestigiosas referentes a la teoría de localización, tales como Osleeb y Ratick (1986), Louveaux, Labbé y Thisse (1989), y Drezner (1992) en *Annals of Operations Research*, Current (1988) en *Environment and Planning B*, Current y Schilling en *Geographical Analysis* (1990) e *INFOR* (1991), Boffey y Karkazis (1991) en *RAIRO*, y Current y Ratick (1992) en *Papers in Reginal Science*.

Se dispone de una amplia variedad de excelentes libros sobre problemas de localización en general (tanto discreta, continua o en redes), entre los que podemos citar a Thisse y Zoller (1983), Arnott (1986) y Hansen *et al* (1987), Love, Morris y Wesolowsky (1988), Hurter y Martinich (1989), Mirchandani y Francis (1990), Francis, McGinnis y White (1992). Drezner (1995), Puerto (1996), y Drezner y Hamacher (2002).

El objetivo primordial de esta tesis es el de estudiar, desarrollar y, en algunos casos, mejorar varios algoritmos de localización en redes. Por consiguiente, en las siguientes secciones revisamos, en orden cronológico, las más destacadas referencias en localización de servicios deseados/no deseados sobre redes considerando un solo criterio y varios criterios.

I.2.2 Localización simple de servicios deseados en redes

Como ya comentamos en la sección previa, Jordan (1869) fue el primero en estudiar un problema de localización en redes tipo árbol. Sin embargo, Hakimi (1964) es considerado el precursor del Análisis de Localización en Redes. En su influyente artículo, los conceptos del centro y la mediana vértice de un grafo son generalizados al *centro absoluto* y a la *mediana* absoluta de una red. Esto condujo a la famosa propiedad de Hakimi: *la mediana absoluta de una red estará siempre localizada en un nodo*. De este modo, la mediana absoluta coincide con la mediana.

Cabe citar los trabajos posteriores de Hakimi (1965), Goldman (1969), Hakimi y Maheshwari (1972), Goldman (1971), Goldman (1972), Handler (1973), Halfin (1974), Minieka (1977), Hakimi, Schmeichel y Pierce (1978), Kariv y Hakimi (1979a,b), Minieka (1980), Minieka (1981), Cuninghame-Green (1984), Hansen, Thisse y Wendell (1986b), Tamir (1987), Chiu (1987), Batta y Palekar (1988), Hansen, Labbé y Nicolas (1991), Sforza (1990), Tamir (1992), Burkard, Çela y Woeginger (1995), Nickel y Puerto (1999), y Kalcsics, Nickel, Puerto y Tamir (2002).

La mayoría de todas estas referencias están relacionadas con el problema del centro o de la mediana. No obstante, algunos autores se dieron cuenta que la combinación de ambos criterios podría producir modelos reales muy interesantes. Citamos a Halpern (1976), Halpern (1978), Halpern (1980), Halpern y Maimon (1983), Handler (1985), Hansen, Labbé y Thisse (1991), Berman y Yang (1991), Carrizosa, Conde, Fernández y Puerto (1994), Ogryczak (1997a), y Averbakh y Berman (1999).

Antes de concluir esta sección, vamos a mencionar brevemente algunas revisiones y libros sobre localización en redes: Tansel, Francis y Lowe (1983a,b), Moon y Chaudhry (1984), Hansen, Labbé, Peeters y Thisse (1987b), Hooker, Garfinkel y Chen (1991), Labbé, Peeters y Thisse (1995), Labbé y Louveaux (1997), y Current, Daskin y Schilling (2002). En cuanto a los libros cabe destacar a Handler y Mirchandani (1979), Daskin (1995), y Miller, Friesz y Tobin (1996).

I.2.3 Problemas de localización de servicios no deseados en redes

No existen muchos trabajos dedicados a la localización de servicios no deseados. Esta materia emergió tímidamente a mediados de los 1970, y gradualmente ha atraído la atención de los investigadores debido a temas medioambientales. Este tipo de problemas son los opuestos a los problemas clásicos del centro (*minimax*) y de la mediana (*minisum*), y por tanto, son generalmente modelados usando los criterios *maximin* y *maxisum*.

Los trabajos más importantes a reseñar son debidos a Slater (1975), Church y Garfinkel (1978), Minieka (1983), Ting (1984), Kuby (1987), Moon (1989), Tamir (1988), Labbé (1990), Tamir (1991), Stowers y Palekar (1993), Kincaid y Berger (1994), Drezner y Wesolowsky (1995), Berman, Drezner y Wesolowsky (1996), Moreno-Pérez y Rodríguez-Martín (1999), Moon y Chaudhry (1984), Tamir (1988, 2001), Melachrinoudis y Zhang (1999), Berman y Drezner (2000), Salhi, Welch y Cuninghame-Green (2000), Burkard, Dollani, Lin y Rote (2001), Burkard y Dollani (2003), López-de-los-Mozos y Mesa (2001), y Carrizosa y Conde (2002).

Con respecto a los estudios y recopilaciones sobre localización no deseada, podemos citar a Moon y Chaudhry (1984), Erkut y Neuman (1989), Erkut y Verter (1995), Verter y Erkut (1995), Plastria (1996), Carrizosa y Plastria (1999), Murray, Church, Gerrard y Tsui (1998), y Cappanera (1999).

No conocermos libros dedicados exclusivamente a la localización de servicios no deseados, salvo las aportaciones hechas dentro de Daskin (1995) y Puerto (1996).

I.2.4 Localización multicriterio de servicios deseados sobre redes

A pesar de su amplia aplicabilidad en problemas reales, los modelos de localización multicriterio en redes no han sido investigados tanto como los problemas unicriterio. Aunque se han desarrollado nuevas líneas de investigación en los últimos años, bastante trabajo queda todavía por llevarse a cabo.

Así, podemos citar los trabajos de Warszawski (1973), Lowe (1978), Schilling (1980), Ross y Soland (1980), Tansel, Francis y Lowe (1980), Nijkamp y Spronk (1981), Hultz, Klingman, Ross y Soland (1981), Tansel, Francis y Lowe (1982), Hansen, Thisse y Wendell (1986a), Buhl (1988), Mirchandani (1990), Puerto y Fernández (1994), Malczewski and Ogryczak (1995), Malczewski y Ogryczak (1996), Krumke, Noltemeier, Ravi y Marathe (1996), Ogryczak (1997b), Ramos, Sicilia and Ramos (1997), Hamacher, Labbé y Nickel (1999), y Ogryczak (1999).

A pesar de la carencia de literatura en problemas de localización multicriterio en redes, en décadas pasadas se han desarrollado muchas aplicaciones prácticas. Aún cuando puede que no estén modelados sobre redes, los citamos simplemente por su aplicabilidad real o su probable uso en un futuro cercano. Así, destacamos a Cohon *et al* (1980), Mladineo, Margeta, Brans y Mareschel (1987), Min (1987), Min (1988), Fortenberry, Mitra y Willis (1989), Barda, Dupuis y Lencioni (1990), Current y Storbeck (1994), Badri, Mortagy y Alsayed (1998), Mahmoud, Fahmy y Labadie (2002).

Con respecto a los estudios y libros en esta materia podemos citar a ReVelle, Cohon y Shobrys (1981a,b), Current, Min y Schilling (1990), Ehrgott y Gandibleux (2000), Handler y Mirchandani (1979), Daskin (1995), y Current, Daskin y Schilling (2002).

Por último, citamos tres tesis interesantes en localización multiobjetivo: Oudjit (1981), Carrizosa (1992), y Zhang (1996).

I.2.5 Localización multicriterio de servicios no deseados sobre redes

Asombrosamente, la literatura en la localización multicriterio en redes comienza a finales de los 1980. La inquietud por la localización de instalaciones indeseables ha crecido bastante en los últimos años, junto con el uso de herramientas multiobjetivo/multicriterio para modelar y solucionar tales problemas.

Así, destacamos a Ratick y White (1988), List y Mirchandani (1991), Erkut y Neuman (1992), Rahman y Kuby (1995), Giannikos (1998), Zhang y Melachrinoudis (2001), Skriver y Andersen (2001), y Hamacher, Labbé, Nickel y Skriver (2002).

Una vez más, los siguientes trabajos se citan por su aplicabilidad real, aunque los problemas considerados, en general, no sean estudiados sobre redes: Melachrinoudis, Min y Wu (1995) y Hokkanen y Salminen (1997). Finalmente, citamos a Saameño (1992) y Skriver (2001) como dos tesis en localización no deseada multicriterio.

I.3 Definiciones básicas y notación

I.3.1 Redes estándar

El concepto matemático de *grafo* puede modelar innumerables problemas reales tales como redes de carreteras, redes de transporte sobre ríos/aire/océanos o redes de comunicación/ordenadores. Todas estas redes son, salvo excepciones, grafos simples (sin lazos ni múltiples aristas), conexos y no dirigidos, con pesos en los vértices y etiquetas en las aristas.

Así, sea $N=(V,E)$ una red no dirigida con tales características, donde $V=\{v_1,v_2,\ldots,v_n\}$ denota el conjunto de vértices o nodos, y $E=\{(v_s,v_t):v_s,v_t\in V\}$ el conjunto de aristas, con $n=|V|$ y $m=|E|$. Los nodos representan los puntos de demanda, fuente o cruce donde se sitúan los servicios o los clientes ya existentes, mientras que las aristas corresponden a las líneas de transporte, caminos, líneas de ferrocarril o canales de comunicación.

Cada nodo $v_i\in V$ se fija con un peso positivo como sigue:

$$\begin{array}{rcl} w: \quad V & \longrightarrow & \mathbb{R}_+ \\ v_i\in V & \longrightarrow & w(v_i)=w_i>0 \end{array}$$

Este peso w_i puede representar ratios de demanda, tiempo/costo/pérdida por unidad de distancia, número de clientes, probabilidad de que una demanda ocurra en el nodo v_i, o incluso la importancia de un daño potencial. Obviamente, los pesos son positivos ya que un peso $w_i=0$ significa nula demanda, tiempo, etc, y por lo tanto no tiene ningún sentido.

Por otra parte, cada arista $e=(v_s,v_t)$ se etiqueta con un número positivo l_e en términos de la siguiente función de longitud:

$$\begin{array}{rcl} l: \quad E & \longrightarrow & \mathbb{R}_+ \\ e=(v_s,v_t)\in E & \longrightarrow & l(e)=l_e>0 \end{array}$$

Así, un punto x dentro de una arista e se mueve en el intervalo $[0,l_e]$. Esta longitud representa el tiempo/costo del recorrido, la confiabilidad o cualquier otra cualidad del recorrido. Las longitudes son positivas ya que cualquier $l_e=0$ implica una distancia nula entre v_s y v_t, y por lo tanto, puede ser descartada.

Además, cada arista se asume que puede ser rectificable, en el sentido de que hay una correspondencia uno a uno entre cada arista y el intervalo $[0,1]$. Por lo tanto, dada cualquier arista $e=(v_s,v_t)\in E$ de longitud l_e y un punto interno $x\in e$, existe un número único $t_e(x)\in[0,1]$ tal que $t_e(x)l_e$ y $(1-t_e(x))l_e$ son las longitudes a lo largo de la arista *e* entre v_s y *x*, y *x* y v_t, respectivamente.

Un *camino* es una secuencia de aristas adyacentes, con cada una de las aristas adyacentes compartiendo un nodo común. Entonces, para cada par de nodos $v_a,v_b\in V$ definimos la *distancia* $d(v_a,v_b)$ entre estos dos nodos como la longitud de cualquier camino más corto en *N* que una v_a y v_b. Por otra parte, dado cualesquiera dos puntos $x,y\in N$, la distancia $d(x,y)$ es la longitud del camino más corto entre *x* e *y*. Dada una arista $e=(v_s,v_t)$, a veces es posible que $d(v_s,v_t)<l_e$ puesto que la arista puede que no proporcione el camino más corto entre los nodos v_s and v_t. Esta función de distancia $d(\cdot,\cdot)$ satisface las siguiente *características métricas* para cualesquiera $x,y\in N$:

1. *No negatividad*: $d(x,y)\geq 0$, con $d(x,y)=0$ si $x=y$.
2. *Simetría*: $d(x,y)=d(y,x)$.
3. *Desigualdad triangular*: $d(x,y)\leq d(x,z)+d(z,y)$, para cualquier $z\in N$.

Llegados a este punto, el tema principal a destacar es que *los modelos de localización en redes se basan generalmente en la suposición de que las distancias corresponden a longitudes de los caminos más cortos*. En este sentido, dada cualquier arista $e=(v_s,v_t)\in E$, un nodo $v_i\in V$ y un punto interno $x\in e$, definimos la distancia entre el punto *x* y nodo v_i como:

$$d(x,v_i)=\min\{x+d(v_s,v_i),l_e-x+d(v_t,v_i)\}$$

El punto sobre *e* donde $d(x,v_i)$ alcanza su equilibrio, esto es $x+d(v_s,v_i)=l_e-x+d(v_t,v_i)$, se denomina *punto cuello de botella* b_i, con

$$b_i=\frac{d(v_t,v_i)+l_e-d(v_s,v_i)}{2}$$

Una característica fundamental de las distancias en redes es la *propiedad de linealidad por trozos y la propiedad de concavidad*. Esta característica indica que la función distancia en $x\in e=(v_s,v_t)$ definido por $d(x,v_i)$:

1. Es continua en toda la arista *e*.
2. A medida que *x* varía desde v_s a v_t en la arista *e*, o
   - crece linealmente con pendiente w_i, o
   - decrece linealmente con pendiente $-w_i$, o
   - primero crece linealmente y luego decrece linealmente con un punto de inflexión en b_i.
3. Es *cóncava*, en el sentido de que un segmento de línea que una dos puntos cualesquiera de la gráfica de la función permanecerá sobre ella o por debajo.

Éstos son los conceptos básicos en redes estándares. En la siguiente sección introducimos las nociones básicas en redes con múltiples criterios, a saber, considerando varios pesos en cada nodo así como varias longitudes en cada arista.

I.3.2 Redes con múltiples parámetros en los nodos y las aristas

La mayoría de la extensa literatura en problemas de la localización en redes se ocupa de la optimización de un *solo criterio*. Este criterio se asocia generalmente a la distancia pesada desde un cierto punto al resto de nodos, por ejemplo, la minimización de la distancia pesada total de un servicio a los clientes.

Sin embargo, hay muchas aplicaciones en las cuales varios parámetros necesitan ser considerados en cada nodo y en cada arista. Así, varios pesos en cada nodo pueden representar diversos criterios que pueden ser considerados por el decisior(es), por ejemplo, índice de demanda, importancia, número de potenciales clientes, etc. Por otra parte, varias longitudes (costes del recorrido) en cada arista pueden significar distancia, tiempo del recorrido, congestión del tráfico, tarifa de peaje, coste del recorrido, impacto ambiental, etc.

En este sentido, en cada nodo $v_i \in V$, la función de peso anterior es ahora substituida por la siguiente:

$$\begin{array}{rccc} w: & V & \longrightarrow & \mathbb{R}^p \\ & v_i \in V & \longrightarrow & w(v_i) = w_i = (w_i^1, \dots, w_i^p) \end{array}$$

donde p es el número de pesos por nodo. Para cualquier vector de pesos w_i, cada componente w_i^r es un número no negativo para $r = 1, \dots, p$, y asumimos que no todos son iguales a cero.

Asimismo, a cada arista se le asocia un vector de longitudes (costes), tal y como sigue:

$$\begin{array}{rccc} l: & E & \longrightarrow & \mathbb{R}^q \\ & e = (v_s, v_t) \in E & \longrightarrow & l(e) = l_e = (l_e^1, \dots, l_e^q) \end{array}$$

en la cual q es el número de longitudes. Una vez más, asumimos que cada componente l_e^r es no negativa para cualquier vector l_e, y no todos $l_e^r = 0$, para $r = 1, \dots, q$.

Sea r un índice de longitud, con $1 \le r \le q$, y sea $x \in e = (v_s, v_t)$ un punto dentro de la arista e. Entonces, $c_e^r(x, v_s)$ se define como el trozo del segmento de línea entre x y v_s considerando la longitud r. Obviamente, se tiene que $0 \le c_e^r(x, v_s) \le l_e^r$, con $c_e^r(x, v_t) = l_e^r - c_e^r(x, v_s)$.

Para cada par de nodos $v_a, v_b \in V$, se define la *distancia* $d^r(v_a, v_b)$ entre estos dos nodos como la longitud de cualquier camino más corto dentro de N que enlace v_a y v_b considerando la longitud r. Asimismo, dado cualesquiera dos puntos $x, y \in N$, la distancia $d^r(x, y)$ es la longitud del camino más corto entre x e y. Éstas q funciones de distancia también cumplen con las características métricas indicadas en la sección precedente.

Dado cualquier nodo $v_i \in V$, tenemos que

$$d^r(x, v_i) = \min\{c_e^r(x, v_s) + d(v_s, v_i), c_e^r(x, v_t) + d(v_t, v_i)\}$$

denota la distancia entre un punto y un nodo considerando la longitud r, siendo $b_i^r = (d^r(v_t, v_i) + l_e^r - d^r(v_s, v_i))/2$ el punto de cuello de botella referido al nodo v_i. Éstas r funciones de distancia satisfacen también las propiedades de linealidad por trozos y la de concavidad.

Finalmente, introducimos la teoría básica de optimización multicriterio/multiobjetivo. Generalmente, los modelos *multicriterio* son los que realizan una optimización simultánea de varios objetivos inconmensurables, por ejemplo, minimizando el recorrido máximo y minimizando el coste total del recorrido. Por otra parte, un concepto bastante relacionado es el de *optimización vectorial*, que determina las soluciones no dominadas a un problema multicriterio.

En este sentido, sean $f = (f_1, f_2, \ldots, f_k)$ y $g = (g_1, g_2, \ldots, g_k)$ dos vectores que pertenecen a $\mathbb{R}^k$. El vector *f* se dice que *domina* al vector *g*, y se denota por $f \prec g$, si y solamente si:

$$f_i \leq g_i, \forall i = 1, \ldots, k \quad \text{and} \quad \exists j \in \{1, \ldots, k\} : f_j < g_j$$

Entonces, dado el subconjunto de vectores $U \subseteq \mathbb{R}^k$, un vector $f \in U$ se llama *no-dominado*, *eficiente* o *Pareto óptimo* (Pareto, 1896) con respecto a subconjunto *U* si no existe otro vector $g \in U$ tal que $g \prec f$. El conjunto de todos los vectores no-dominados con respecto a *U* se denota por U_{ND}. Para un conocimiento más amplio en optimización multicriterio, se remite al lector a Steuer (1986).

I.4 Clasificación de problemas

Como comentamos en la sección I.2, puede haber actualmente más de 3000 referencias en localización. Esta literatura tan amplia debía ser clasificada de alguna manera. Por consiguiente, varios autores han propuesto algunos esquemas de clasificación para indicar concisamente y describir sin ambigüedad los modelos de localización.

La primera tentativa fue hecha por Handler y Mirchandani (1979). Luego vinieron los trabajos de Brandeau y Chiu (1989), Krarup y Pruzan (1990), Francis, McGinnis y White (1992), Eiselt y Laporte (1995), Daskin (1995), y Hale (1999).

Todas estas últimas clasificaciones fueron desarrolladas para describir una amplia gama de modelos. Sin embargo, algunos investigadores han sugerido otros esquemas para modelos más particulares de localización, como los descritos en Moon y Chaudhry (1984), Erkut y Neuman (1989), Eiselt, Laporte y Thisse (1993), y Carrizosa, Conde, Muñoz y Puerto (1995).

Recientemente, Hamacher y Nickel (1998) propusieron un esquema de 5 posiciones que se puede utilizar no solamente para las clases de modelos específicos de localización, sino también para cubrir todos los modelos de localización. Ha estado en uso desde 1992, y ha demostrado ser muy útil en investigación, desarrollo de software y en docencia universitaria. Por lo tanto, decidimos seguir esta taxonomía para definir y describir los modelos de localización desarrollados en esta tesis.

El esquema de clasificación tiene cinco posiciones, descritas como

Pos1 / Pos2 / Pos3 / Pos4 / Pos5

Si no existe ninguna suposición especial, se indica con un •. Para una explicación más detallada, el lector es remitido a Hamacher y Nickel (1998).

Capítulo II (Resumen)

Localización bicriterio de un servicio deseado en redes

"En muchos problemas del mundo real, la función objetivo es una mezcla de dos diferentes, y posiblemente adversos objetivos, el centro y la mediana"
J. HALPERN

II.1 Introducción

Los problemas de localización del centro y de la mediana, que consideran un costo por arista, fueron introducidos por Hakimi (1964). El objetivo del problema del centro es localizar un punto en la red de modo que la distancia al nodo más lejano sea mínima. En el mundo real, este tipo de función se asocia con frecuencia a la localización de servicios de emergencia tales como estaciones de ambulancias, bomberos y comisarías de policía.

Por otra parte, el problema de la mediana se refiere a la localización de un punto en la red que minimiza la distancia total (la suma de todas las distancias) desde este punto a todos los nodos. Los problemas reales relacionados con la mediana se presentan en la localización de los puntos del servicio que se dedican a la distribución de personas y mercancías (reparto de productos, transporte escolar, servicio de correos, etc).

Sin embargo, estos dos conceptos se combinan a veces. Por ejemplo, la localización de un supermercado debe considerar la función centro, de modo que no esté muy lejos para los clientes, y la función mediana, de modo que el reparto de mercancía sea mucho más rápido.

La combinación convexa de estas dos funciones (centro y mediana) se llama *función cent-dian*, y el punto que minimiza tal función se llama el *cent-dian* de una red. Esta combinación convexa fue propuesta inicialmente a mediados de los 1970 por Halpern (1976), que acuñó el término *cent-dian*, e independientemente por Handler (1976, 1985) que propuso el término *medi-center*.

Sin embargo, en muchas situaciones, la determinación del cent-dian de una red se debe realizar considerando varios criterios. Así, usando el ejemplo del supermercado introducido anteriormente, podríamos definir dos parámetros por arista: su longitud, y el coste de transporte implicado (mantenimiento del vehículo, gasolina, peaje, etc).

Siguiendo los trabajos hechos en Ramos, Sicilia y Ramos (1992, 1997), nos centramos en este capítulo en el estudio del problema del cent-dian con dos funciones objetivo asociadas, y presentamos un método para encontrar el conjunto de todos los posible puntos eficientes del cent-dian. Los algoritmos que proponemos para obtener estos puntos hacen uso de técnicas de geometría computacional.

II.2 Notación y formulación del modelo

Sea $N=(V,E)$ una red finita, simple, no dirigida y conexa, con $V=\{v_1,v_2,\ldots,v_n\}$ siendo el conjunto de nodos (vértices) y $E=\{(v_s,v_t):v_s,v_t\in V\}$ como el conjunto de aristas, $m=|E|$. Se asocia un peso positivo w_i a cada nodo $v_i\in V$, y en cada arista $e=(v_s,v_t)\in E$ se colocan dos parámetros o costes independientes (longitudes) l_e^r, con $r=1,2$, que pueden representar la longitud de la arista *e*, el tiempo de recorrido entre v_s y v_t, el coste de envío de una unidad de cierto material a lo largo de la arista *e*, etc.

Formalmente, para $r=1,2$, la *función centro* puede ser formulada como

$$f_{\max}^r(x)=\max_{v_i\in V} w_i d^r(x,v_i),\quad \forall x\in N$$

y un punto $x_c\in N$ es *centro* (absoluto) para el *r*-ésimo costo si $f_{\max}^r(x_c)=\min_{x\in N} f_{\max}^r(x)$.

Por otra parte, *función mediana* se define como la distancia mínima total desde un punto (mediana) de la red al conjunto de nodos. La formulación de esta función es:

$$f_{\text{sum}}^r(x)=\sum_{v_i\in V} w_i d^r(x,v_i),\quad \forall x\in N$$

y un punto $x_m\in N$ es una *mediana* para el *r*-ésimo costo cuando $f_{\text{sum}}^r(x_m)=\min_{x\in N} f_{\text{sum}}^r(x)$.

La función del cent-dian se compone de la combinación convexa de las dos funciones anteriores:

$$f_{\text{cd}}^r(\lambda,x)=\lambda f_{\max}^r(x)+(1-\lambda)f_{\text{sum}}^r(x)=\lambda\max_{v_i\in V}\{w_i d^r(x,v_i)\}+(1-\lambda)\sum_{v_i\in V} w_i d^r(x,v_i)$$
$$\forall x\in N,\quad 0\le\lambda\le 1,\quad r=1,2$$

Dado un índice de coste *r*, el λ-cent-dian es el punto de la red que minimiza la combinación convexa de los dos objetivos. El valor de λ refleja la importancia atribuida a la distancia pesada máxima comparada con la distancia pesada total.

Sin embargo, uno puede todavía observar una gran discrepancia en los valores de las funciones $f_{\max}^r$ y f_{sum}^r, debido al hecho de que los valores de la segunda función son siempre más grandes que la primera. Esto parece contradecir cualquier idea intuitiva de equidad distribucional entre los criterios, justificando así otra combinación convexa para construir la función del cent-dian. Por lo tanto, varios autores utilizan la *función centro no pesada* y la *función mediana pesada* dividida por la suma de pesos. Véase por ejemplo Halpern (1978), Hansen, Labbé y Thisse (1991), Labbé, Peeters y Thisse (1995).

De acuerdo con estos autores, quitamos los pesos de la función centro como sigue

$$f_{\max}^r(x)=\max_{v_i\in V} d^r(x,v_i),\quad \forall x\in N$$

Entonces, tenemos la siguiente función objetivo:

$$F_{\text{cd}}^r(\lambda,x)=\lambda\max_{v_i\in V} d^r(x,v_i)+\frac{(1-\lambda)}{W}\sum_{v_i\in V} w_i d^r(x,v_i)=\lambda f_{\max}^r(x)+(1-\lambda)f_{\text{sum}}^r(x)/W$$
$$\forall x\in N,\quad 0\le\lambda\le 1,\quad r=1,2$$

donde $W = \sum_{v_i \in V} w_i$ representa la suma de pesos. Así, el problema a solucionar puede ahora ser formulado como sigue: encontrar un punto x en N tal que

$$\min_{x \in N}(F^1_{\text{cd}}(\lambda, x), F^2_{\text{cd}}(\lambda, x)), \quad 0 \le \lambda \le 1$$

Para solucionar este problema, un orden en $\mathbb{R}^2$ debe ser definido. Consideramos el orden de Pareto, es decir, dados dos vectores $f, g \in \mathbb{R}^2$, el orden componente a componente se define como

$$f = (f_1, f_2) \le (g_1, g_2) = g \Leftrightarrow f_i \le g_i,\ i = 1,2$$

Si por lo menos una de las últimas desigualdades es estricta, se utiliza entonces la expresión $f \prec g$, y f se dice que *domina* a g.

Sea $U = \{(F^1_{\text{cd}}(\lambda, x), F^2_{\text{cd}}(\lambda, x)), x \in N, \lambda \in [0,1]\}$ el conjunto de todos los posibles vectores asociados con todos los puntos x en la red N. El conjunto de todos los vectores no dominados se denota por U_{ND}.

El conjunto de todas las localizaciones x en N tales que $(F^1_{\text{cd}}(\lambda, x), F^2_{\text{cd}}(\lambda, x)) \in U_{ND}$ es denotado por L, y un punto $x \in L$ se denomina *no dominado* o *punto eficiente de localización*. El resto del capítulo se dedica a encontrar estos puntos eficientes de localización en el problema del cent-dian biobjetivo.

II.3 Propiedades del cent-dian

Dado un índice de coste (longitud) r y una arista $e = (v_s, v_t) \in E$, sea P^r_e el conjunto de puntos que contienen los nodos $v_s, v_t \in V$ y los mínimos locales de $f^r_{\max}(x)$, para cualquier punto x en N. Las características siguientes del cent-dian fueron establecidas y demostradas en Halpern (1978):

Propiedad II.1. *Dados r, λ y un punto interior x en la arista e, la función*

$$F^r_{\text{cd}}(\lambda, x) = \lambda f^r_{\max}(x) + (1-\lambda) f^r_{\text{sum}}(x) / W$$

es continua, lineal a trozos y con un número finito de valores mínimos locales de $F^r_{\text{cd}}(\lambda, x)$ en la arista e, los cuales se alcanzan todos en puntos miembros de P^r_e.

Propiedad II.2. *Dado r, la función $F^r_{\text{cd}}(\lambda, x) = \min\{F^r_{\text{cd}}(\lambda, x) : x \in N\}$ es una función continua, lineal a trozos y cóncava para λ, $0 \le \lambda \le 1$.*

Propiedad II.3. *Dado el r-ésimo costo, si $x_{\text{cd}}(\lambda) \in N$ es un punto cent-dian para un λ dado, entonces la función $f^r_{\max}(x_{\text{cd}}(\lambda))$ es una función no creciente de λ y la función $f^r_{\text{sum}}(x_{\text{cd}}(\lambda)) / W$ es una función no decreciente de λ.*

Propiedad II.4. *Dados r y λ, el cent-dian de una red está en el camino mínimo que conecta el centro y la mediana de la red.*

La primera característica indica que el conjunto de localizaciones del λ-cent-dian están en el conjunto $P^r_N = \{P^r_e : e \in E\}$, es decir, necesitamos solamente evaluar la función objetivo $F^r_{\text{cd}}(\lambda, x)$ en los nodos de la red y en los mínimos locales de $f^r_{\max}(x)$ a lo largo de todas las aristas. El conjunto P^r_N es siempre finito. Este resultado ha sido utilizado por varios autores para obtener algoritmos en $O(mn\log n)$ para determinar el λ-cent-dian en una red para un valor dado de λ.

Sin embargo, el conjunto de localizaciones eficientes para el problema del cent-dian biobjetivo puede ser infinito y no numerable. Considerando la última propiedad del λ-cent-dian para el caso uniobjetivo, podemos preguntarnos si es posible encontrar un resultado similar para el caso biobjetivo: ¿Está el cent-dian biobjetivo de una red sobre el camino más corto que conecta cualquier centro uniobjetivo con cualquier mediana uniobjetivo de la red? Desafortunadamente, la respuesta es negativa.

Otra pregunta podría ser si el conjunto de puntos eficientes está en cualquier camino mínimo que conecte los cent-dians de los dos objetivos. La respuesta también es negativa.

Por tanto, no es posible generalizar alguna de las propiedades obtenidas para el λ-cent-dian uniobjetivo. Para caracterizar el conjunto de localizaciones eficientes debemos analizar todas las aristas de la red. La siguiente sección presenta una regla simple para eliminar aristas no eficientes, esto es, aristas que contengan sólo puntos no eficientes.

II.4 Eliminación de aristas

Los algoritmos que presentamos en las siguientes secciones realizan los cálculos sobre las aristas de la red para obtener los puntos eficientes. Por esta razón, es muy importante que el número de aristas a examinar no sea muy grande. Aquí mostramos una regla simple para quitar las aristas que no contendrán puntos eficientes.

Dado un valor de λ, $0 \le \lambda \le 1$, y para todas las aristas $e=(v_s,v_t)\in E$, comprobar si alguna de estas condiciones es verificada para algunos nodos:

a)
$$\lambda \min\{d^1(v_s,v_k),d^1(v_t,v_k)\} + (1-\lambda)\sum_{v_i\in V} w_i \min\{d^1(v_s,v_k),d^1(v_t,v_k)\}/W \ge F^1_{\text{cd}}(\lambda,v^1_{\text{cd}})$$
y
$$\lambda \min\{d^2(v_s,v_l),d^2(v_t,v_l)\} + (1-\lambda)\sum_{v_i\in V} w_i \min\{d^2(v_s,v_l),d^2(v_t,v_l)\}/W \ge F^2_{\text{cd}}(\lambda,v^1_{\text{cd}})$$

o

b)
$$\lambda \min\{d^1(v_s,v_k),d^1(v_t,v_k)\} + (1-\lambda)\sum_{v_i\in V} w_i \min\{d^1(v_s,v_k),d^1(v_t,v_k)\}/W \ge F^1_{\text{cd}}(\lambda,v^2_{\text{cd}})$$
y
$$\lambda \min\{d^2(v_s,v_l),d^2(v_t,v_l)\} + (1-\lambda)\sum_{v_i\in V} w_i \min\{d^2(v_s,v_l),d^2(v_t,v_l)\}/W \ge F^2_{\text{cd}}(\lambda,v^2_{\text{cd}})$$

En las fórmulas de arriba v^1_{cd} y v^2_{cd} son los vértices cent-dian de la red para cada objetivo. Estos vértices cent-dian pueden ser calculados usando el algoritmo presentado en Halpern (1978). Los valores $F^1_{\text{cd}}(\lambda,\cdot)$ y $F^2_{\text{cd}}(\lambda,\cdot)$ son los valores de la función cent-dian sobre estos nodos. Si a) o b) se verifica, entonces la arista e es eliminada. Si no, la arista es examinada.

II.5 Calculando las funciones centro y mediana

Como se indicó en las secciones anteriores, la función del cent-dian $F^r_{\text{cd}}(\lambda,x)$ para el r-ésimo objetivo se forma como combinación convexa de las funciones centro y mediana. Así, para calcular cada $F_{\text{cd}}^{r}(\lambda, x)$ debemos primero desarrollar los algoritmos que permitan calcular estas dos funciones.

Suponiendo que la matriz de distancias está ya calculada, la función centro se puede calcular en tiempo $\mathrm{O}(mn + n^2 \log n)$, mientras que calcular la función mediana toma tiempo $\mathrm{O}(mn \log n)$.

II.6 Determinando el cent-dian biobjetivo

Proponemos ahora un algoritmo exacto en $\mathrm{O}(mn \log n)$ que determina los puntos cent-dian biobjetivo. Para obtener los vectores no dominados que corresponden a los puntos eficientes, podemos utilizar el algoritmo de Hershberger (1989) para calcular la envoltura inferior de los segmentos en el espacio objetivo en tiempo $\mathrm{O}(S \log S)$, donde S es el número de segmentos.

Dada una arista, hay $\mathrm{O}(n)$ segmentos de recta que unen los pares de valores $(F_{\text{cd}}^{1}, F_{\text{cd}}^{2})$, por lo que habrá a lo sumo $S = mn$ segmentos de recta. Ya que la complejidad del paso final es más grande o igual que la complejidad de los pasos anteriores, la complejidad de tiempo total del algoritmo es $\mathrm{O}(mn \log n)$.

II.7 Resultados computacionales

El método seguido para probar la calidad de este algoritmo ha sido la generación de grafos planares aleatorios con un número de nodos n entre 10 y 100, y un número de aristas $m = 3n - 6$. El valor de λ varía de 0 a 1, con un incremento de 0.1.

Para cada par (n, λ), se han solucionado diez problemas. Queremos remarcar que los tiempos promedio no aumentan a medida que n lo hace. Esto es debido al número de aristas restantes después de aplicar la regla de eliminación descrita en la sección II.4.

Por otra parte, los tiempos promedio mínimos se alcanzan para $\lambda = 1$. Asimismo, en todos los casos los tiempos promedio son menores de un minuto y medio.

II.8 Conclusiones

En este capítulo hemos propuesto un algoritmo en $\mathrm{O}(mn \log n)$ para solucionar el problema del cent-dian biobjetivo. Este procedimiento también permite resolver dos interesantes casos particulares: para $\lambda = 0$ se obtienen los puntos eficientes para el problema de la mediana biobjetivo, y para $\lambda = 1$ se determinan los puntos eficientes para el problema del centro biobjetivo.

Debemos comentar que el conjunto de puntos eficientes para localizar el λ-cent-dian es infinito, en comparación con el caso uniobjetivo donde el λ-cent-dian está situado en el conjunto de nodos o en el conjunto de mínimos locales de la función centro.

Capítulo III (Resumen)

Localización multicriterio de un servicio 1-mediana en redes

"La mayoría de los problemas de localización son por naturaleza inherentemente multiobjetivo"
M. DASKIN

III.1 Introducción

En este capítulo ponemos el énfasis sobre la localización de un servicio deseado sobre una red y nuestro objetivo será el problema 1-mediana. Consideramos que los puntos de demanda corresponden a los vértices, e intentamos localizar el punto en la red tal que se minimice la suma de distancias a todas los vértices. Estudiaremos este problema considerando múltiples objetivos, es decir, la red toma múltiples longitudes en las aristas, lo cual implica considerar múltiples funciones de distancia.

El problema de la 1-mediana simple fue resuelto por Hakimi desde 1964, cuando demostró que la localización óptima debía estar en los vértices de la red. Aunque los investigadores han prestado mucha atención a la 1-mediana, asombrosamente ciertas generalizaciones de estos problemas, que tienen en cuenta varias consideraciones de la vida real, no se han estudiado a fondo.

Pocos investigadores han estudiado algunas generalizaciones. Handler y Mirchandani (1979) dieron una lista de varias generalizaciones naturales que pudieran ocurrir, y que incluyen la consideración de demandas y de costes probabilísticas, costes del transporte no lineales y multi-atributo, múltiples materias y múltiples objetivos. Obviamente, es imposible estudiar todas estas generalizaciones aquí. En su lugar, solamente será analizado el problema 1-mediana con múltiples objetivos.

En este sentido, supongamos que la demanda de ciertas mercancías está concentrada en diversas ciudades representadas por los vértices en una red de caminos. Asumimos que es posible considerar varias longitudes en cada arista de la red. Estas longitudes pueden representar el tiempo necesario para cruzar la arista, el coste del recorrido, consecuencias para el medio ambiente, etc. Así, se expresan los múltiples criterios como la minimización del recorrido total, de la suma del coste del recorrido, de la suma de las consecuencias para el medio ambiente, etc. Deseamos localizar un servicio deseado en la red tal que los múltiples criterios se optimicen (el problema 1-mediana multiobjetivo).

Oudjit (1981) estudió el problema 1-mediana multiobjetivo en árboles. En este trabajo presentaremos un procedimiento para calcular los puntos eficientes de localización para el problema 1-mediana multiobjective en cualquier red. Un trabajo relacionado con este capítulo se presenta en Hamacher, Labbé y Nickel (1999), quienes trataron el problema mediana multicriterio considerando varios pesos en los nodos.

Consideraremos una red conexa y no dirigida $N(V,E)$, sin lazos ni aristas múltiples, donde $V=\{v_1,v_2,\ldots,v_n\}$ es el conjunto de vértices (nodos) y $E=\{(v_s,v_t):v_s,v_t\in V\}$ es el conjunto de aristas de la red. Esta condición no implica pérdida de generalidad, porque los puntos localizados nunca podrían estar en los lazos. La razón estriba en que el vértice relacionado con cualquier lazo sería siempre un punto mejor de localización.

Utilizaremos la siguiente notación:

$n=|V|$: número de vertices.
$m=|E|$: número de aristas.
q: número de criterios u objetivos.
l_e^r: longitud de la arista e bajo el criterio $r=1,2,\ldots,q$.
$d^r(v_i,v_j)$: distancia del camino más corto desde v_i a v_j bajo el criterio r.
$(l_e^1,l_e^2,\ldots,l_e^q)$: vector de longitudes para los diferentes criterios en cada arista e.

Para cualquier criterio r y cada punto x en N, definimos

$$f^r(x)=\sum_{v_i\in V} d^r(x,v_i)$$

Si x_m es un punto en N de modo que $f^r(x_m)=\min_{x\in N} f^r(x)$, entonces x_m es una 1-mediana para el objetivo r.

Dado los puntos $x,y\in N$, decimos que x domina a y si, y solamente si, $f^r(x)\le f^r(y)$ para todos r, y $f^r(x)<f^r(y)$ para al menos un r. El conjunto de puntos eficientes es el conjunto de todos los puntos de la red que no estén dominados.

Recalcamos que las funciones objetivo son cóncavas en cada arista, entonces para cualquier $x\in N$ y cualquier criterio r, existe un vértice $v_i\in V$ tal que $f^r(v_i)\le f^r(x)$. En este capítulo veremos que no todas las 1-medianas multiobjetivo están situadas en los vértices. Así, nuestro problema es más interesante, puesto que los posibles puntos de localización se podrían situar sobre toda la red.

III.2 Algunos ejemplos y observaciones

Hemos comentado previamente que Hakimi demostró que el problema 1-mediana se convierte en el problema del vértice 1-mediana usando la característica de concavidad de la función objetivo. Nos podríamos preguntar si los puntos eficientes de localización para el caso multiobjetivo están siempre en los vértices del grafo. La respuesta es negativa.

Podemos también podríamos pensar que todos los puntos en los caminos mínimos que enlazan vértices mediana deben ser puntos eficientes en una red multiobjetivo. Existen puntos eficientes en estos caminos mínimos, pero también podemos encontrar algunos puntos no eficientes o dominados.

Ahora, podemos preguntarnos si todos los puntos eficientes están solamente en los caminos mínimos que enlazan los vértices 1-mediana o si algunos de estos puntos eficientes se podrían encontrar también fuera. La última cuestión se refiere a si los puntos eficientes deben estar solamente en esos vértices que contengan cualquier vértice 1-mediana. La respuesta es negativa para ambas.

III.3 Puntos eficientes para el problema 1-mediana multiobjetivo

Para simplificar la búsqueda de los puntos eficientes, proponemos ahora una regla simple para eliminar aristas de la red. Puesto que las funciones objetivo son cóncavas en cada arista, una arista $e=(v_s,v_t)\in E$ puede eliminarse si se satisface la condición siguiente:

$$f^r(v_s)\geq f^r(v_m) \quad \text{and} \quad f^r(v_t)\geq f^r(v_m), \quad \forall r=1,2,\ldots,q$$

donde v_m es cualquier vértice mediana para algún criterio *r*.

Si no, es posible comprobar si existen puntos eficientes en esta arista. Por lo tanto, las aristas que enlazan los vértices 1-mediana nunca se eliminarán.

A continuación explicamos el procedimiento de búsqueda de puntos eficientes en una red multiobjetivo. Este procedimiento será aplicado a las aristas restantes. Se basa en dos algoritmos. El primer algoritmo determina las funciones de distancia para cada objetivo. Estas funciones son poligonales cóncavas y serán totalmente caracterizadas cuando calculemos, para cada objetivo *r*, los puntos de inflexión de las líneas poligonales donde la pendiente cambia su valor.

Dado la matriz de la distancia *d*, la complejidad de este algoritmo es $\mathrm{O}(mqn\log n)+\mathrm{O}(qmn+qn^2\log n)$, donde *m* es el número de aristas, *q* es el número de objetivos y *n* el número de vértices. El cómputo de las matrices de distancia para los *q* objetivos requiere tiempo $\mathrm{O}(qmn+qn^2\log n)$ usando Fredman y Tarjan (1987), mientras que la ordenación de los puntos b_i^r es realizado a lo máximo en tiempo $\mathrm{O}(n\log n)$.

El segundo algoritmo utiliza los puntos de inflexión de las funciones objetivo para dividir las aristas en segmentos según los valores máximos de los objetivos. Se determinan así los puntos no dominados para cada segmento, y los vectores de valores de los puntos obtenidos se comparan para quitar los dominados. Para ello, se define el conjunto de puntos *P* y el conjunto de segmentos *S*. Estos conjuntos se comparan para obtener los puntos no dominados o eficientes en la red. Para ello, se realiza una comparación directa, respectivamente, entre todos los puntos y entre los puntos y los segmentos, almacenando los no dominados.

La comparación entre los segmentos almacenados en el conjunto *S* no es tan sencilla como cabría esperar. Por lo tanto, será explicada a fondo en una sección posterior.

La complejidad total es $\mathrm{O}(m^2q^3)$. Esta complejidad se calcula como sigue. En cada arista hay como máximo $q+1$ segmentos. El número de segmentos y de puntos a comparar será $\mathrm{O}(mq)$. Puesto que se realizan como máximo $\binom{mq}{2}$ comparaciones, y cada comparación requiere tiempo $\mathrm{O}(q)$, entonces la complejidad total es $\mathrm{O}(m^2q^3)$.

III.4 Comparación segmento contra segmento

Primero, cada segmento $[x_i, x_{i+1}]$ del conjunto S se divide en segmentos $[x_i, b_j] \cup [b_j, b_k] \cup \cdots \cup [b_p, x_{i+1}]$ con una sola línea de función objetivo sobre ellos, donde $b_j, b_k, \ldots, b_p$ son los puntos de inflexión de las q funciones objetivo con respecto al primer objetivo.

Dados cualesquiera dos segmentos $X = [x_0, x_1] \in S$ e $Y = [y_0, y_1] \in S$, y dos puntos internos $x \in X$ e $y \in Y$, las q funciones objetivo son de la forma

$$f_X^r(x) = f_X^r(x_0) + m_X^r(x - x_0), \quad f_Y^r(x) = f_Y^r(y_0) + m_Y^r(y - y_0), \quad \forall r = 1, \ldots, q$$

Si X domina Y ($X \prec Y$), entonces las siguientes desigualdades deben ser satisfechas:

$$\begin{array}{l} f_X^1(x) \le f_Y^1(y) \Rightarrow f_X^1(x_0) + m_X^1(x - x_0) \le f_Y^1(y_0) + m_Y^1(y - y_0) \\ f_X^2(x) \le f_Y^2(y) \Rightarrow f_X^2(x_0) + m_X^2(x - x_0) \le f_Y^2(y_0) + m_Y^2(y - y_0) \\ \quad \vdots \qquad\qquad\qquad\qquad \vdots \\ f_X^j(x) < f_Y^j(y) \Rightarrow f_X^j(x_0) + m_X^j(x - x_0) < f_Y^j(y_0) + m_Y^j(y - y_0) \\ \quad \vdots \qquad\qquad\qquad\qquad \vdots \\ f_X^q(x) \le f_Y^q(y) \Rightarrow f_X^q(x_0) + m_X^q(x - x_0) \le f_Y^q(y_0) + m_Y^q(y - y_0) \end{array}$$

Por tanto, para cualquier desigualdad i obtenemos

$$f_X^i(x_0) + m_X^i(x - x_0) \le f_Y^i(y_0) + m_Y^i(y - y_0) \Rightarrow y \ge \frac{f_X^i(x_0) - f_Y^i(y_0) - m_X^i x_0 + m_Y^i y_0}{m_Y^i} + \frac{m_X^i}{m_Y^i} x \quad \text{(III.1)}$$

Sea $p^i = (f_X^i(x_0) - f_Y^i(y_0) - m_X^i x_0 + m_Y^i y_0) / m_Y^i$ y $q^i = m_X^i / m_Y^i$. Entonces, (III.1) se rescribe como $y \ge p^i + q^i x$. De acuerdo a estos valores p^i y q^i, surgen los siguientes tipos de desigualdades:

- Si $m_X^i = 0 \Rightarrow q^i = 0 \Rightarrow \begin{cases} y \le p^i : \text{ tipo } \boxed{e} \\ y \ge p^i : \text{ tipo } \boxed{f} \end{cases}$
- Si $q^i > 0 \Rightarrow \begin{cases} y \le p^i + q^i x : \text{ tipo } \boxed{a} \\ y \ge p^i + q^i x : \text{ tipo } \boxed{b} \end{cases}$
- Si $q^i < 0 \Rightarrow \begin{cases} y \le p^i + q^i x : \text{ tipo } \boxed{c} \\ y \ge p^i + q^i x : \text{ tipo } \boxed{d} \end{cases}$
- En el caso particular en el que $m_Y^i = 0 \Rightarrow q^i = \infty$, y así, la desigualdad se mantiene con respecto a x, esto es $\begin{cases} x \le u^i : \text{ tipo } \boxed{g} \\ x \ge u^i : \text{ tipo } \boxed{h} \end{cases}$, con $u^i = (f_Y^i(y_0) - f_X^i(x_0) + m_X^i x_0) / m_X^i$.

Sea T el conjunto de todas las desigualdades, siendo $T_a, T_b, \ldots, T_h$ los conjuntos de de las diferentes desigualdades con $T = T_a \cup T_b \cup \cdots \cup T_h$. Todas estas desigualdades forman una región R donde $X \prec Y$. Cada desigualdad se denota con la letra del tipo al que pertenece, a saber $a \in T_a$, etc. Obviamente, si hay dos desigualdades $a \in T_a$ y $b \in T_b$ tal que $a(x) < b(x)$, $\forall x \in [x_0, x_1]$, entonces la región R es vacía, y por lo tanto $X \nprec Y$. El siguiente Lema III.1 indica este resultado para todas las desigualdades dentro de T.

Lema III.1. *Si existen desigualdades* $a \in T_a, b \in T_b, \ldots, h \in T_h$, *tal que* $a(x) < b(x)$, $c(x) < d(x)$, $e(x) < f(x)$ *o* $g(y) < h(y)$, *para todos los puntos* $x \in X$ *e* $y \in Y$, *entonces* $X \nprec Y$.

Es obvio que hay una conexión directa entre la región convexa R definido por el sistema de desigualdades T y un problema de programación lineal de dos variables. Este hecho podía conducirnos a solucionar la comparación de segmentos usando algoritmos de programación lineal como el simplex. Sin embargo, como demostramos en las siguientes páginas, este problema se puede solucionar fácilmente con técnicas de geometría computacional.

Una vez que hayamos clasificado las desigualdades, procedemos a encontrar los puntos en el segmento X que dominan a puntos en el segmento Y. Esto es, todos los puntos $x \in [x_{\min}, x_{\max}]$ que dominen a todos los puntos $y \in [y_{\min}, y_{\max}]$, o lo que es lo mismo, $[x_{\min}, x_{\max}] \prec [y_{\min}, y_{\max}]$. Nuestra meta es encontrar estos dos valores en el segmento Y.

En el análisis posterior primero computamos $y_{\max}$, y por medio de un resultado clásico, obtendremos $y_{\min}$. Cuando algún conjunto de desigualdades en T es vacío, el valor de $y_{\max}$ es calculado fácilmente, según lo indicado en el siguiente resultado.

Lema III.2. *Si* $T_a = \emptyset$ *y* $T_c = \emptyset$ *entonces* $y_{\max} = y_1$. *Cuando* $T_a = \emptyset$ *obtenemos* $y_{\max} = \min_{c \in T_c} c(x_0)$, *con* $x_{\max} = x_0$. *Asimismo, si* $T_c = \emptyset$, $y_{\max} = \min_{a \in T_a} a(x_0)$, *con* $x_{\max} = x_1$.

En otro caso, $T_a \neq \emptyset$ y $T_c \neq \emptyset$, y por lo tanto, el valor $y_{\max}$ es alcanzado en el punto de intersección entre dos desigualdades de T_a y T_c. En este sentido, dadas dos desigualdades $a \in T_a$ y $c \in T_c$ se define $x = I(a,c) \in X$ como el punto de intersección entre ellos. Sean $Q = \{I(a,c) : \forall a \in T_a, \forall c \in T_c\}$ todos los puntos de intersección entre todas las desigualdades en T_a and T_c. Sea $F(x) = \{a(x) : \forall a \in T_a\}$ el conjunto de desigualdades con pendiente positiva.

Asumimos que por lo menos existe una intersección entre una desigualdad de T_a y otra de T_c. Si no, significa que todas las desigualdades en T_a están por debajo de T_c, o viceversa, y por lo tanto el valor $y_{\max}$ puede ser obtenido usando el Lema III.2.

Además, asumimos que el punto de intersección se produce por debajo de y_1. Si no, el siguiente resultado establece el valor de $(x_{\max}, y_{\max})$.

Lema III.3. *Si todas las intersecciones entre las desigualdades en* T_a *y* T_c *se producen por encima de* y_1, *entonces* $y_{\max} = y_1$, *siendo* $[x^0_{\max}, x^1_{\max}]$ *el intervalo donde se alcanza este máximo, con* $x^0_{\max} = \max\{x \in X : t(x) = y_1, t \in T_a\}$ *y* $x^1_{\max} = \min\{x \in X : t(x) = y_1, t \in T_c\}$.

Teniendo en cuenta estas últimas suposiciones, debe existir un punto $z \in Q$ tal que $F(z) = \min_{x \in Q} F(x)$. Por tanto, $x_{\max} = z$ y $y_{\max} = F(z)$. El siguiente Lema demuestra este resultado.

Lema III.4. $y_{\max} = F(z)$ *y por lo tanto* $x_{\max} = z$.

De este resultado podemos derivar inmediatamente la siguiente consecuencia.

Corolario III.1. *Suponiendo que todas los puntos de intersección caen dentro de* $X \times Y$, *sea* $a \in T_a$ *y* $c \in T_c$, *con* $x = I(a,c)$ *e* $y = F(x)$. *Si* $a_m(x) < y$ *entonces* $F(I(a_m, c)) < y$, *y si* $c_m(x) < y$ *entonces* $F(I(a, c_m)) < y$.

Este último resultado será usado posteriormente en el algoritmo para acelerar el proceso de búsqueda de $(x_{\max}, y_{\max})$. Finalmente, una vez que hemos computado el valor $y_{\max}$, si $y_{\max} < y_0$ la región R se hace vacía, y consecuentemente $X \not\prec Y$.

Para obtener el valor mínimo $y_{\min}$ podemos aplicar el Lema III.2, Lema III.3 y el Lema III.4 sobre las desigualdades T_d y T_b, y el clásico resultado de optimización que establece

$\min(y) = -\max(-y)$. Sean d_m y b_m las desigualdades cuya intersección produce $x_{\min} = I(d_m, b_m)$, con $y_{\min} = d_m(x_{\min})$.

Tan pronto como hemos obtenido los valores $y_{\min}$ y $y_{\max}$, el segmento X no dominará al segmento Y si $y_{\min} > y_{\max}$. En otro caso, podemos comprobar si estos valores máximos y mínimos pueden ser alcanzados con la intersección de desigualdades T_a y T_b o T_c y T_d, respectivamente.

Antes de calcular los nuevos valores de $y'_{\min}$ e $y'_{\max}$, el siguiente resultado establece la condición por la cual el segmento X no dominará al segmento Y.

Lema III.5. *Si* $x_{\min} < x_{\max}$ *y* $a_m(x_{\min}) < y_{\min}$ *y* $b_m(x_{\max}) > y_{\max}$, *entonces* $X \nprec Y$.

Ahora buscamos el nuevo punto máximo $y'_{\max}$ entre los puntos de intersección de las desigualdades T_a y T_b. En primer lugar, eliminamos de T_b todas las desigualdades que no nos sirven:

$$T'_b = T_b / \{b \in T_b : b(x_{\max}) \leq y_{\max}\}$$

Definimos un nuevo conjunto T'_b porque las desigualdades en T_b se usan luego para obtener $y'_{\min}$. Si $T'_b = \varnothing$ entonces no existe ningún $b \in T_b$ tal que $b(x_{\max}) > y_{\max}$, y de este modo, el punto máximo $(x_{\max}, y_{\max})$ permanece sin alterar.

En otro caso, procedemos a obtener el nuevo punto máximo $(x'_{\max}, y'_{\max})$ por medio de un resultado similar al Lema III.4. Sea $Q' = \{I(a,b) : \forall a \in T_a, \forall b \in T'_b, \text{pendiente}(a) < \text{pendiente}(b)\}$ el conjunto de puntos de intersección entre las desigualdades T_a y T'_b, donde pendiente(a) y pendiente(b) denota las pendientes de los segmentos de línea de cada desigualdad. Este requerimiento es importante ya que queremos que las desigualdades T'_b crucen a las de T_a tan alto como sea posible. Por tanto, como $T'_b \neq \varnothing$, debe existir al menos un punto $z' \in Q'$ tal que $F(z') = \min_{x \in Q'} F(x)$, y por consiguiente establecemos el siguiente resultado.

Lema III.6. $y'_{\max} = F(z')$ *y por tanto* $x'_{\max} = z'$.

Como en el Corolario III.1, se puede derivar un resultado que mejora la búsqueda de $(x'_{\max}, y'_{\max})$.

Corolario III.3. *Asumiendo que todas los puntos de intersección caen dentro de* $X \times Y$, *sea* $a \in T_a$ *y* $b \in T'_b$, *con* $x = I(a,b)$ *e* $y = F(x)$. *Si* $a'_m(x) < y$ *entonces* $F(I(a'_m, b)) < y$, *y si* $b'_m(x) > y$ *entonces* $F(I(a, b'_m)) < y$.

Tratamos ahora de ajustar el valor $y_{\min}$ buscando un nuevo valor $y'_{\min}$ en los puntos de intersección entre las desigualdades T_a y T_b. Inicialmente, nos deshacemos de todas las desigualdades inservibles en T_a.

$$T_a = T_a / \{a \in T_a : a(x_{\min}) \geq y_{\min}\}$$

En este caso, no hay necesidad de crear un nuevo conjunto T'_a, puesto que T_a no será utilizado más adelante. Si $T_a = \varnothing$, no hay ninguna desigualdad en T_a que pueda mejorar $y_{\min}$. Si no, el nuevo valor mínimo $y'_{\min}$ puede ser obtenido de una forma muy similar al Lema III.6 junto al hecho de que $\min(y) = -\max(-y)$.

Lema III.7. *Si* $x_{\min} > x_{\max}$ *y* $c_m(x_{\min}) < y_{\min}$ *y* $d_m(x_{\max}) > y_{\max}$, *entonces* $X \nprec Y$.

Si este resultado no se cumple, continuamos para obtener el nuevo valor máximo $y'_{\max}$. Como hicimos arriba, primero eliminamos de T_d todas las desigualdades que están por debajo de $y_{\max}$:

$$T'_d = T_d / \{d \in T_d : d(x_{\max}) \leq y_{\max}\}$$

Ahora, si $T'_d \neq \varnothing$ buscamos $y'_{\max}$ en los puntos de intersección entre las desigualdades T_c y T'_d. Después de que esto es llevado a cabo, eliminamos todas las desigualdades inservibles de T_c:

$$T_c = T_c / \{c \in T_c : c(x_{\min}) \geq y_{\min}\}$$

y buscamos el valor $y'_{\min}$ en los puntos de intersección de T_d y T_c.

Todos los resultados anteriores establecen los puntos mínimos y máximos dentro de *R* donde *X* domina a *Y*. Tales resultados se basan en la comparación de valores sobre los puntos de intersección entre las desigualdades. En el caso del Lema III.4 las desigualdades consideradas son T_a y T_c. El conjunto de intersecciones forma el conjunto *Q*. Si hay una sola desigualdad para cada objetivo $r = 1,\ldots,q$, habrá como máximo $O(q)$ desigualdades en *R*. Por lo tanto, $|Q| \leq q^2$. Sin embargo, más adelante demostraremos que $(x_{\min}, y_{\min})$ y $(x_{\max}, y_{\max})$ pueden calcularse en tiempo $O(q)$.

Empezamos analizando el cálculo de $(x_{\max}, y_{\max})$. Sea $M = \{(a,c) : a \in T_a, c \in T_c\}$ el conjunto de pares (emparejamientos) de las desigualdades T_a y T_c tal que $|M| = \max\{|T_a|, |T_c|\}$, con $|M| \leq |Q|$. Por ejemplo, si tenemos los siguientes conjuntos $T_a = \{a_1, a_2, a_3, a_4\}$ y $T_c = \{c_1, c_2\}$, entonces $M = \{(a_1,c_1),(a_2,c_2),(a_3,c_1),(a_4,c_2)\}$, con $|M| = 4 = |T_a|$.

Cada par de desigualdades $(a,c) \in M$ produce un punto $x = I(a,c)$ y un valor $y = F(x)$. Sea $x_m \in X$ un punto tal que $F(x_m) = \min_{(a,c)\in M} F(I(a,c))$. Este punto x_m puede que sea el óptimo, con $x_{\max} = x_m$, $y_{\max} = y_m = F(x_m)$, y siendo a_m y c_m las desigualdades que se cruzan en este máximo. Por consiguiente, todas las desigualdades en T_a y T_c son entonces eliminadas. De lo contrario, existen algunas desigualdades por debajo de a_m y c_m.

Sea $a^* \in T_a : a^*(x_m) = \min_{a \in T_a} a(x_m)$ y $c^* \in T_c : c^*(x_m) = \min_{c \in T_c} c(x_m)$ las desigualdades más inferiores por debajo de $F(x_m)$. Sea $x_m = I(a^*, c^*)$ e $y_m = F(x_m)$. Este valor es el nuevo punto óptimo. Es más, ahora podemos eliminar de *M*, en el peor caso, una desigualdad *a* o *c* de cada pareja $(a,c) \in M$. En efecto, cada pareja puede tener sólo una desigualdad bajo y_m, esto es, o $a(x_m) < y_m$ o $c(x_m) < y_m$. Ambas desigualdades no pueden estar por debajo porque contradice el hecho de que (x_m, y_m) es el punto mínimo. Por tanto, al menos $|M|/2 = \max\{|T_a|, |T_c|\}/2$ desigualdades son eliminadas. Este análisis demuestra el siguiente resultado.

Lema III.8. *En cada búsqueda del punto óptimo* (x_m, y_m) *podemos eliminar al menos* $|M|/2$ *desigualdades de M.*

Finalmente, el siguiente teorema establece la complejidad teórica del algoritmo.

Teorema III.1. *El algoritmo que calcula el punto máximo (mínimo) dentro de R se ejecuta en tiempo* $O(q)$.

Megiddo (1982) y Dyer (1984) propusieron algoritmos en $O(q)$ para calcular, respectivamente, los valores mínimo y máximo de un problema de programación lineal de dos variables. Sin embargo, la complejidad de tiempo de estos métodos está acotada por $4q$, mientras que el nuevo método está acotado por $2q$.

III.5 Un ejemplo para ilustrar los algoritmos

En esta sección se presenta un ejemplo aplicando los algoritmos propuestos en las secciones previas. Consideramos una red compuesta por 9 vértices y 16 aristas.

III.6 Conclusiones

En este capítulo, se ha estudiado el problema de localizar un servicio en una red con múltiples objetivos tipo mediana que consisten en minimización de la suma de las distancias o longitudes del punto de localización a los vértices de la red. Aunque este problema, conocido como 1-mediana, es fácil para el caso uniobjetivo (Hakimi, 1964), su extensión al caso multiobjetivo no lo es tanto.

Hemos demostrado en este capítulo que los puntos eficientes no necesitan estar solamente en los vértices de la red, ni en los caminos más cortos que enlazan vértices mediana correspondiendo a cada objetivo tipo mediana. Por lo tanto, la búsqueda de puntos eficientes no se restringe a los vértices o a una parte específica de la red, sino que debe ser ampliada a todas las aristas de la red. Para simplificar esta búsqueda, hemos propuesto una regla simple para quitar los vértices de la red que nunca contendrán puntos eficientes. Para determinar los puntos eficientes de localización hemos presentado un método que consiste en dos algoritmos. El primer calcula para cada arista los puntos de inflexión donde la pendiente cambia. Las funciones objetivo son obtenidas usando estos puntos de inflexión. El segundo divide cada arista en varios segmentos considerando los puntos máximos de las funciones objetivo. Estos segmentos se comparan entonces para obtener los puntos eficientes de localización.

Capítulo IV (Resumen)

Extendiendo el marco de la localización multiobjetivo en redes al problema cent-dian

"Racionalmente hablando, no existen criterios para la elección de criterios"
J. KRARUP & P.M. PRUZAN

IV.1 Introducción

En el capítulo anterior analizábamos un problema de la localización en redes con varios objetivos mediana, y propusimos un algoritmo polinomial para solucionarlo. Sería razonable ahora estudiar el problema de localización del centro multiobjetivo en redes. Sin embargo, consideramos más notable analizar el problema λ-cent-dian en redes con no solamente varias longitudes en las aristas, sino también varios pesos en los nodos. Así, siguiendo el modelo del Capítulo II, para $\lambda = 0$ podemos solucionar el problema mediana, mientras que para $\lambda = 1$ obtenemos la solución al problema del centro.

Según lo indicado en el Capítulo I, el problema de localización del centro fue propuesto y solucionado por Hakimi (1964). Este problema concierne cuestiones de *equidad*, y se utiliza para localizar servicios de emergencia tales como bomberos, policía, servicios de la ambulancia o de rescate, etc.

Por otra parte, si deseamos minimizar la distancia total (agregada o promedio pesado), entonces se plantea el problema mediana (Hakimi, 1964). La mediana se vincula a *eficacia espacial*, y es conveniente para localizar los servicios que implican la distribución de personas o de mercancías, estos es, colegios, centros comerciales, servicio de correo, etc.

Sin embargo, puesto que la mediana se basa en calcular un promedio, puede discriminar áreas remotas y de densidad de población baja, contra áreas centralmente situadas y de la alta densidad de población, lo cual implica ninguna equidad (Hansen, Labbé y Thisse, 1991; Ogryczak, 1997).

Por otra parte, la localización de un servicio en el centro puede causar un gran aumento en la distancia total, los cual significa ninguna eficacia espacial (Hansen, Labbé y Thisse, 1991; Ogryczak, 1997). Halpern (1976) introdujo el λ-cent-dian como compromiso entre el centro y la mediana, por medio de una combinación convexa. Este modelo permite explotar en común las ventajas principales de cada problema.

Ya hemos comentado que la mayoría de la extensa literatura en análisis de localización en redes considera sólo un criterio en cada nodo (peso) y/o un criterio en cada arista (longitud). Sin embargo, hay muchas aplicaciones en las cuales varios criterios necesitan ser considerados. Por ejemplo, varios pesos pueden representar demanda, importancia social y política, número de potenciales servicios complementarios, etc. Asimismo, varios costes (longitudes) pueden significar distancia, tiempo, congestión del tráfico, peaje, el etc.

Siguiendo los trabajos hechos en los Capítulos II e III, analizamos el problema λ-cent-dian en redes, considerando varios pesos en los nodos y varias longitudes en las aristas.

IV.2 Definiciones y formulación del modelo

Sea $N=(V,E)$ una red simple (sin lazos ni múltiples aristas), conexa y no dirigida, siendo $V=\{v_1,v_2,\ldots,v_n\}$ el conjunto de nodos, y $E=\{(v_s,v_t):v_s,v_t\in V\}$ el conjunto de aristas. Sea *p* el número de pesos en cada nodo, y *q* el número de longitudes (costos) en cada arista. Así, para cada nodo en *V*, definimos la siguiente función de peso

$$\begin{array}{llll} w: & V & \longrightarrow & \mathbb{R}^p \\ & v_i\in V & \longrightarrow & w(v_i)=w_i=(w_i^1,\ldots,w_i^p) \end{array}$$

Asimismo, sobre cada arista en *E* definimos la siguiente función de longitud

$$\begin{array}{llll} l: & E & \longrightarrow & \mathbb{R}^q \\ & e=(v_s,v_t)\in E & \longrightarrow & l(e)=l_e=(l_e^1,\ldots,l_e^q) \end{array}$$

Sea *r* un índice de longitud, con $1\le r\le q$, y $x\in e=(v_s,v_t)$ un punto interno. Definimos $c_e^r(x,v_s)$ como la longitud del segmento de línea entre *x* y v_s con respecto a la longitud *r*, con $0\le c_e^r(x,v_s)\le l_e^r$ y $c_e^r(x,v_t)=l_e^r-c_e^r(x,v_s)$.

Para cada par de nodos v_a y v_b, la distancia entre tales nodos, denotada por $d^r(v_a,v_b)$, se define como la longitud de cualquier camino mínimo en *N* uniendo v_a y v_b con respecto a la longitud *r*. De la misma forma, dado cualquier punto $x\in N$ y cualquier nodo $v_i\in V$, sea

$$d^r(x,v_i)=\min\{c_e^r(x,v_s)+d(v_s,v_i),c_e^r(x,v_t)+d(v_t,v_i)\}$$

la distancia entre el punto *x* y el nodo v_i considerando la longitud *r*.

Tal y como hicimos en el Capítulo II, definimos ahora la función centro no pesada (Hansen, Labbé and Thisse, 1991) como

$$f_{\max}^r(x)=\max_{v_i\in V}d^r(x,v_i),\quad \forall x\in N,\ r=1,\ldots,q$$

y un punto $x_c\in N$ en un *centro* (absoluto) para la longitud *r* si $f_{\max}^r(x_c)=\min_{x\in N}f_{\max}^r(x)$.

Por otro lado, la función mediana (Hansen, Labbé and Thisse, 1991) se define como

$$f_{\text{sum}}^{sr}(x)=\frac{1}{W^s}\sum_{v_i\in V}w_i^s d^r(x,v_i),\quad \forall x\in N,\ s=1,\ldots,p,\ r=1,\ldots,q$$

donde $W^s=\sum_{v_i\in V}w_i^s$ representa la suma de pesos para un cierto índice de peso *s*. Un punto $x_m\in N$ en una *mediana* para un índice de peso dado *s* y un cierto índice de longitud *r* cuando $f_{\text{sum}}^{sr}(x_m)=\min_{x\in N}f_{\text{sum}}^{sr}(x)$.

Finalmente, la función λ-cent-dian surge de la combinación convexa de estas dos últimas funciones, esto es

$$F_{\text{cd}}^{sr}(\lambda,x)=\lambda\max_{v_i\in V}d^r(x,v_i)+\frac{(1-\lambda)}{W}\sum_{v_i\in V}w_i^s d^r(x,v_i)=\lambda f_{\max}^r(x)+(1-\lambda)f_{\text{sum}}^{sr}(x)$$
$$\forall x\in N,\quad 0\le\lambda\le 1,\quad s=1,\dots,p\quad r=1,\dots,q$$

Las propiedades de la función λ-cent-dian fueron establecidas y comentadas en el Capítulo II.

Sea $F(\lambda,x)=(F_{\text{cd}}^{11}(\lambda,x),F_{\text{cd}}^{12}(\lambda,x),\dots,F_{\text{cd}}^{pq}(\lambda,x))\in\mathbb{R}^{p\times q}$. Para un valor dado de λ, $0\le\lambda\le 1$, el problema consiste en encontrar el conjunto $x_{\text{cd}}\in N$ tal que

$$F(\lambda,x_{\text{cd}})=\min_{x\in N}F(\lambda,x)$$

Sea $k=p\times q$, y sea $g=(g^1,g^2,\dots,g^k)$ y $h=(h^1,h^2,\dots,h^k)$ dos vectores en $\mathbb{R}^k$. Se dice que el vector g domina al vector h, denotado como $g\prec h$, si y solo si $g^i\le h^i,\forall i$ y $g^i<h^i$ para al menos un i. Sea $U=\{(F^1(\lambda,x),F^2(\lambda,x),\dots,F^k(\lambda,x)):\forall x\in N\}$ el conjunto de todos los posibles valores de los vectores en N. Un vector $F\in U$ es *no dominado* o *eficiente* si $\nexists G\in U$ tal que $G\prec F$. El conjunto de todos los vectores no dominados se denota por U_{ND}.

Por tanto, sea $L=\{x\in N:(F^1(\lambda,x),\dots,F^k(\lambda,x))\in U_{\text{ND}}\}$. Un punto $x\in L$ se dice *no dominado* o *eficiente*. Nuestro objetivo es encontrar el conjunto U_{ND}, y de este modo, el conjunto de puntos eficientes de localización L en N. La siguiente sección presenta el algoritmo que determina el conjunto L.

IV.3 El algoritmo

Teniendo en cuenta el enfoque al problema de la mediana multiobjetivo, ahora presentamos el algoritmo que soluciona el problema λ-cent-dian multicriterio.

Como se comentó en el Capítulo II, para una arista dada $e\in E$ y para todos los puntos interiores $x\in e$, la función λ-cent-dian $F_{\text{cd}}^{sr}(\lambda,x)$, con $1\le s\le p$ and $1\le r\le q$, no es ni convexa ni cóncava. Debido a esto, debemos dividir las $p\times q$ λ-cent-dian funciones de acuerdo con sus puntos de inflexión. Posteriormente, el algoritmo procede de una forma muy similar al procedimiento de la mediana multiobjetivo.

Una importante diferencia entre este algoritmo y el de la mediana multiobjetivo yace en la división en segmentos y puntos de las $k=p\times q$ funciones λ-cent-dian. Este proceso se lleva a cabo en a lo máximo $\text{O}(kn)$ pasos, ya que pueden haber como mucho $\text{O}(n)$ puntos de inflexión en cada una de las k funciones. Por tanto, el número de segmentos y puntos generados por todas las aristas es $\text{O}(mnk)$. La comparación dos-a-dos de todos estos elementos lleva $\text{O}(m^2n^2k^2)$ pasos, y cada comparación se realiza en tiempo $\text{O}(k)$. Así, suponiendo que todas las k matrices de distancia ya están calculadas, el algoritmo del λ-cent-dian multicriterio se ejecuta en $\text{O}(m^2n^2k^3)$.

IV.4 Un breve ejemplo

Se generó aleatoriamente una red con $n=5$ nodos y $m=9$ aristas. Se asociaron a cada nodo dos pesos, mientras que cada arista lleva dos longitudes. El valor del parámetro λ es 0.5.

Siguiendo las directrices del algoritmo, primero se computa las q funciones centro no pesadas. A continuación, se calculan las $p \times q$ funciones mediana pesada. Una vez tengamos todas las funciones centro y mediana, se procede a construir las funciones λ-cent-dian a través de la combinación convexa de estas dos últimas funciones. Posteriormente, se dividen estas funciones λ-cent-dian para obtener el conjunto de puntos P y el conjunto de segmentos S. De aquí en adelante solo resta comparar los segmentos en S y los puntos en P.

Antes de comentar las conclusiones del capítulo, en la siguiente sección presentamos los resultados computacionales del algoritmo λ-cent-dian multicriterio.

IV.5 Resultados computacionales

Se generaron redes planares aleatorias ($m = 3n - 6$) con $n = 10$ hasta 100 nodos. El número de pesos p y el número de longitudes q varían de 1 a 3. El valor de los pesos varía uniformemente entre 1 y 10, mientras que los valores de las longitudes están uniformemente distribuidos de 1 a 50. El parámetro λ varía de 0 a 1 con un incremento de 0.25. Se generaron diez ejemplos para cada problema.

Dada una combinación fija de n, p y q, los tiempos de computación permanecen iguales independientemente del valor del parámetro λ. Los tiempos de computación crecen proporcionalmente al número de pesos p y al número de longitudes q.

IV.6 Conclusiones

Siguiendo el modelo presentado en el Capítulo II, y teniendo en cuenta el enfoque del problema de la mediana multiobjetivo propuesto en el Capítulo III, hemos desarrollado un algoritmo polinomial que resuelve el problema del λ-cent-dian multicriterio para un valor dado de λ.

Este modelo permite obtener la solución al problema del centro no pesado multicriterio en el caso de $\lambda = 1$. Sin embargo, el modelo puede ser ligeramente modificado para adecuarse al problema del centro pesado multicriterio. Por otro lado, cuando $\lambda = 0$, se resuelve el problema de la mediana pesada multicriterio, el cual es una generalización del modelo presentado en el capítulo anterior.

En los siguientes capítulos, estudiamos varios modelos para la localización de servicios no deseados con respecto a un solo criterio así como múltiples criterios.

Capítulo V (Resumen)

El problema de localización de un centro no deseado en redes

"Las cosas deberían hacerse tan simples como sea posible,
pero no más simples"
A. EINSTEIN

V.1 Introducción

Usualmente, los servicios a ser localizados son deseables, esto es, los potenciales clientes (nodos) tratan de atraerlos tan cerca como sea posible. Por ejemplo, servicios tales como policía/bomberos, hospitales, escuelas o incluso centros comerciales son típicos servicios deseables.

Sin embargo, algunas veces los servicios pueden ser considerados no deseables por la población circundante, tales como reactores nucleares, instalaciones militares, plantas contaminantes, prisiones, centros correccionales y vertederos. Erkut y Neuman (1989) distinguen entre servicios *perjudiciales* (dañinos, letales) y *detestables* (molestos, insoportables). Para mayor claridad, los denominamos *no deseables*.

No hay muchos trabajos dedicados a la localización no deseada en redes. Minieka (1983) propuso el *anticentro* (maxmax) y la *antimediana* (maxsum). Según Erkut y Neuman (1989) y Cappanera (1999), no existía ningún trabajo referente a la localización de un centro no deseado (*maximin*) en la literatura hasta ahora. El primer algoritmo en $O(mn)$ para el problema 1-maximin fue brevemente sugerido por Tamir (1988) usando Megiddo (1982) y Dyer (1984). Más recientemente, Melachrinoudis y Zhang (1999) han propuesto otro procedimiento en $O(mn)$ basado en cotas superiores y en una pequeña modificación de Dyer (1984). El trabajo más reciente con respecto a este problema es debido a Berman y Drezner (2000), quienes proporcionaron un enfoque de programación lineal en tiempo $O(mn)$. El algoritmo que presentamos mejora computacionalmente estos enfoques anteriores.

El principal propósito de este capítulo es doble. Primero, ajustamos las cotas superiores ya propuestas, reduciendo aún más el número de aristas a ser procesadas y, sobre cada arista, el número de operaciones para obtener el punto óptimo. En segundo lugar, proponemos un nuevo algoritmo en tiempo $O(mn)$ para el 1-centro no deseado en redes. Este nuevo método se basa en la intersección de las líneas de las funciones de distancia con signo de pendiente opuesto, y evitando el emparejamiento de líneas superfluas. Aunque la complejidad teórica es idéntica a los métodos ya reportados, los tiempos de cómputo del nuevo algoritmo son menores.

V.2 Notación y formulación del modelo

Sea $N=(V,E)$ una red simple (sin lazos ni múltiples aristas), no dirigida y conexa, siendo $V=\{v_1,v_2,\ldots,v_n\}$ el conjunto de nodos, y $E=\{(v_s,v_t):v_s,v_t\in V\}$ el conjunto de aristas, con $|E|=m$. En cada nodo v_i, asociamos un peso positivo (demanda) w_i como una función $w:V\to\mathbb{R}_+$, $v_i\in V\to w(v_i)=w_i>0$.

Además, cada arista $e=(v_s,v_t)$ está etiquetada con una longitud positiva (costo del recorrido) l_e. Así, tenemos una función de longitud $l:E\to\mathbb{R}_+$, $e=(v_s,v_t)\in E\to l(e)=l_e>0$.

Para cada par de nodos $v_i,v_j\in V$ definimos la *distancia* entre dos nodos $d(v_i,v_j)$ como la longitud del camino mínimo entre v_i y v_j. Dada cualquier arista $e=(v_s,v_t)\in E$, $v_i\in V$ y un punto $x\in e$, definimos la distancia entre x y un nodo v_i como $d(x,v_i)=\min\{x+d(v_s,v_i),l_e-x+d(v_t,v_i)\}$. El punto donde $d(x,v_i)$ alcanza su equilibrio (es decir, $x+d(v_s,v_i)=l_e-x+d(v_t,v_i)$) se llama *punto cuello de botella*:

$$b_i=\frac{d(v_t,v_i)+l_e-d(v_s,v_i)}{2} \tag{V.1}$$

Dado cualquier punto $x\in N$ definimos $f(x)=\min_{v_i\in V} w_i d(x,v_i)$. Entonces, el problema consiste en

$$\max_{x\in N}\min_{v_i\in V} w_i d(x,v_i)=\max_{x\in N} f(x) \tag{V.2}$$

y un punto $x_N\in N$ es un 1-centro no deseado si y solo si $f(x_N)=\max_{x\in N} f(x)$.

Este problema es el opuesto al problema del 1-centro (minimax), por lo que podría llamarse *anti-centro*. Desafortunadamente, este término ya fue acuñado por Minieka (1983) para definir el problema *maxmax*. Nosotros proponemos en su lugar el término 1-*uncenter* para definir el punto óptimo de localización.

Algunas propiedades interesantes surgen para este problema, todas definidas y demostradas en Melachrinoudis y Zhang (1999) y en Berman y Drezner (2000). Sea x_e un punto en la arista $e=(v_s,v_t)\in E$ tal que $f(x_e)=\max_{x\in e} f(x)$. Este punto x_e se llama el 1-*uncenter local* sobre la arista *e*.

V.3 Nuevas propiedades para el problema 1-uncenter pesado

Reformulamos el problema 1-uncenter sobre cada arista $e=(v_s,v_t)\in E$ como sigue: $x_N\in N$ es un punto 1-uncenter si y solo si $f(x_N)=\max_{e\in E} f(x_e)$.

Como el punto 1-uncenter local es el máximo valor de una función objetivo $f(x)$, debería estar localizado en la intersección de dos líneas de las funciones de distancia que tengan pendiente de signo opuesto. Nuestro objetivo es encontrar estas dos líneas y el punto de intersección entre ellas.

Así, dada $e=(v_s,v_t)\in E$ y para todos $v_i\in V$ podemos obtener las siguientes relaciones:

$$\begin{aligned}&b_i>0 \Leftrightarrow \text{línea de la función distancia del vértice } v_i \text{ es creciente a la izquierda de } b_i.\\&b_i<l_e \Leftrightarrow \text{línea de la función distancia del vértice } v_i \text{ es decreciente a la derecha de } b_i.\end{aligned} \tag{V.3}$$

Reemplazamos b_i en (V.3), y sea $d_i=d(v_s,v_i)-d(v_t,v_i)$. Entonces:

$$\begin{aligned} d_i < l_e &\Leftrightarrow \text{línea de la función distancia creciente.} \\ -d_i < l_e &\Leftrightarrow \text{línea de la función distancia decreciente.} \end{aligned} \tag{V.4}$$

Dividimos el conjunto de nodos V en dos conjuntos, dependiendo si la función distancia crece ($L = \{v_k \in V : d_k < l_e\}$) o decrece ($R = \{v_k \in V : -d_k < l_e\}$) desde v_s, con $|L| + |R| \leq 2n$. Para cualquier nodo $v_i \in V$, definimos ahora las funciones $F_i^L(x)$ y $F_i^R(x)$ como:

$$F_i^L(x) = w_i(x + d(v_s, v_i))$$
$$F_i^R(x) = w_i(l_e - x + d(v_t, v_i))$$

Para cualquier par de vértices $v_i \in L, v_j \in R$ también definimos

$$X(v_i, v_j) = \frac{w_j(l_e + d(v_t, v_j)) - w_i d(v_s, v_i)}{w_i + w_j}$$

el cual calcula el punto de intersección entre dos líneas de funciones distancia con pendiente de signo opuesto. Para el caso especial donde $v_i = v_j$, obtenemos el punto cuello de botella b_i.

Al existir como máximo n líneas de funciones de distancia en los conjuntos L y R, habrán como mucho n^2 posibles puntos de intersección. Sea P_e el conjunto que contiene tales puntos de intersección para una arista $e \in E$:

$$P_e = \{X(v_i, v_j) : \forall v_i \in L,\ \forall v_j \in R\}, \quad |P_e| \leq n^2$$

y sea P_N el conjunto obtenido al unir, para cada arista, todos los puntos que pertenecen a P_e, esto es

$$P_N = \bigcup_{e \in E} P_e, \quad |P_N| \leq mn^2$$

Melachrinoudis y Zhang (1999) establecieron que el *Conjunto Finito Dominante* (*Finite Dominating Set*, FDS) para el problema 1-maximin en redes con pesos positivos es $V \cup B_A \cup B_C$. Sin embargo, esto es incorrecto, y debe ser subsanado. El siguiente resultado determina el FDS correcto.

Lema V.1. *El FDS para el problema del 1-uncenter pesado en redes es* P_N.

Teniendo en cuenta estos últimos resultados, podemos obtener una nueva formulación del problema 1-uncenter (V.2) como sigue.

Dada $e = (v_s, v_t) \in E$, sea $F(x) = \{F_i^L(x) : \forall v_i \in L\}$ (o $F(x) = \{F_i^R(x) : \forall v_i \in R\}$) el conjunto de de funciones de distancia pesadas izquierdas (derechas) sobre la arista *e*. Definimos el punto z_e en *e* tal que $F(z_e) = \min_{x \in P_e} F(x)$.

Lema V.2. *El punto 1-uncenter* x_e *en la arista e es* z_e.

Denotando F_e como el valor $F(x_e) = F(z_e)$, el problema original es equivalente al siguiente.

Teorema V.1. *El problema del 1-uncenter en redes puede ser expresado como*

$$\max_{e \in E} \min_{x \in P_e} F(x)$$

y un punto $x_N \in N$ *es un punto 1-uncenter si y solo si* $F(x_N) = \max_{e \in E} F_e$.

Teniendo en cuenta el resultado previo, el problema continuo inicial 1-uncenter (V.2) en redes se convierte en un problema discreto. Finalmente recalcamos que, a pesar de que el tamaño del conjunto P_N es como máximo mn^2, el punto 1-uncenter puede ser encontrado en una red en tiempo $O(mn)$.

V.4 Enfoques recientes y nuevas cotas

Uno de los últimos algoritmos en tiempo $O(mn)$ ha sido presentado por Melachrinoudis y Zhang (1999). Su método se basa en tres cotas superiores. Dada una arista $e=(v_s,v_t)\in E$, la primera cota superior se define como $x_{UB1}=X(v_s,v_t)$ y $F_{UB1}=F_s^L(x_{UB1})=F_t^R(x_{UB1})$. Esta cota no se puede mejorar. Sin embargo, las siguientes dos cotas pueden ajustarse. Sea

$$v_g \in V : F_g^L(0)=\min_{\substack{v_k\in V\\ v_k\neq v_s}} F_k^L(0), \qquad v_h \in V : F_h^R(l_e)=\min_{\substack{v_k\in V\\ v_k\neq v_t}} F_k^R(l_e) \tag{V.5}$$

los nodos en los cuales la función distancia alcanza el valor mínimo en ambos lados. La segunda cota superior es $x_{gh}=X(v_g,v_h)$ y $F_{gh}=F_g^L(x_{gh})=F_h^R(x_{gh})$.

Esta cota superior puede mejorarse ligeramente en dos casos especiales. Así, introducimos un nuevo punto z y su ordenada definidos por:

$$(z,F_z)=\begin{cases}(X(v_s,v_h),F_s^L(X(v_s,v_h))) & \text{if } F_s^L(x_{gh})\le F_{gh} \quad \text{(Figure 3a)}\\ (X(v_g,v_t),F_t^R(X(v_g,v_t))) & \text{if } F_t^R(x_{gh})\le F_{gh} \quad \text{(Figure 3b)}\\ (0,\infty) & \text{otherwise}\end{cases} \tag{V.6}$$

Proponemos una nueva cota $F_{UB2}=\min\{F_{gh},F_z,F_{UB1}\}$, y por tanto, x_{UB2} es igual a x_{gh}, z o x_{UB1}.

Asimismo, la tercera cota superior se define considerando

$$v_p \in V : F_p^L(l_e)=\min_{\substack{v_k\in V\\ v_k\neq v_s}} F_k^L(l_e), \qquad v_q \in V : F_q^R(0)=\min_{\substack{v_k\in V\\ v_k\neq v_t}} F_k^R(0) \tag{V.7}$$

con $x_{pq}=X(v_p,v_q)$ y $F_{pq}=F_p^L(x_{pq})=F_q^R(x_{pq})$.

Esta cota puede ser también mejorada estableciendo un nuevo punto y, y su ordenada, los cuales vienen definidos por:

$$(y,F_y)=\begin{cases}(X(v_s,v_q),F_s^L(X(v_s,v_q))) & \text{if } F_s^L(x_{pq})\le F_{pq}\\ (X(v_p,v_t),F_t^R(X(v_p,v_t))) & \text{if } F_t^R(x_{pq})\le F_{pq}\\ (0,\infty) & \text{otherwise}\end{cases} \tag{V.8}$$

Proponemos una nueva cota $F_{UB3}=\min\{F_{pq},F_y,F_{UB1}\}$ y x_{UB3} se actualiza a x_{pq}, y o x_{UB1}.

Por otro lado, la más reciente contribución al problema 1-uncenter es debida a Berman y Drezner (2000), quienes presentaron un breve trabajo sobre la localización de un servicio no deseado en una red. Estudiaron este problema desde un punto de vista de programación lineal, haciendo uso del algoritmo dado en Megiddo (1982) para obtener un procedimiento en tiempo $O(mn)$. Sin embargo, este enfoque no es muy rápido ya que se debe procesar cada arista para encontrar el valor óptimo.

V.5 El algoritmo

El algoritmo tiene dos partes: la primera computa las tres cotas superiores; la segunda busca el mejor punto en el conjunto de líneas de funciones de distancia.

La función *UnCenter* necesita sólo dos entradas: la red $N = (V, E)$ y la matriz de distancias *d*, que puede ser computada en tiempo $O(mn + n^2 \log n)$ usando Fredman y Tarjan (1987). La salida es F_N y el conjunto de puntos *S* donde se alcanza este valor.

El cálculo de la primera cota es sencillo. La segunda se calcula usando (V.5) y (V.6), mientras que (V.7) y (V.8) calculan la tercera cota superior.

El par (x_e, F_e) se iguala a la mejor cota superior. A continuación, dividimos el conjunto *V* en dos conjuntos *L* y *R*. Las líneas de las funciones de distancia que pertenecen a estos conjuntos son emparejadas, de forma que el número de emparejamientos debe ser igual a $\max\{|L|,|R|\}$. En cada emparejamiento, se calcula el punto de intersección entre las dos líneas y su valor de ordenada relacionado. El punto de intersección con mínimo valor de función se almacena en (x_e, F_e).

El valor x_e se proyecta sobre la función objetivo (envoltura inferior), y de este modo, obtenemos un nuevo valor de (x_e, F_e). Todos las líneas por encima de F_e son eliminadas de *L* y *R*. El algoritmo continúa hasta que $F_e < F_N$, esto es, esta arista no puede mejorar el óptimo global, o ambos *L* y *R* están vacíos.

El emparejamiento máximo asegura un máximo de *n* líneas emparejadas, lo cual esencial para eliminar tantas líneas como sea posible. El siguiente Lema establece este resultado.

Lema V.3. *En cada iteración del bucle 'while', al menos* $(\max\{|L|,|R|\})/2$ *nodos son eliminados de L y R.*

Dada la matriz de distancia, el siguiente teorema demuestra que la complejidad total del nuevo algoritmo 1-uncenter es $O(mn)$.

Teorema V.2. *El algoritmo anterior resuelve eficientemente el problema del 1-uncenter pesado en tiempo* $O(mn)$.

V.6 Un ejemplo

La red tiene $n = 8$ nodos y $m = 18$ aristas. Los pesos de los nodos varían aleatoriamente de 1 a 9, mientras que las longitudes varían aleatoriamente de 1 a 49.

El algoritmo procesa sólo 6 de las 18 posibles aristas, con sólo 5 emparejamientos. Para el mismo ejemplo, el algoritmo maximin de Melachrinoudis y Zhang (1999) necesita procesar 7 aristas, y computa 26 emparejamientos. Aunque estas cifras no parezcan importantes, serán bastante relevantes cuando la red aumenta de tamaño en el número de nodos y aristas.

V.7 Resultados computacionales

Los tiempos dados por el método de Berman y Drezner (2000) son extremadamente altos, ya que se tiene que ejecutar sobre todas las aristas existentes. Al incluir las cotas propuestas en este capítulo, se reducen drásticamente el número de aristas procesadas, y por tanto, el tiempo total de cómputo. El nuevo algoritmo alcanza tiempos de computación más rápidos incluso que la versión con cotas superiores de Berman y Drezner.

Debido a las cotas superiores más ajustadas, se procesan menos aristas en el algoritmo 1-uncenter que en el procedimiento maximin. Además, el número de líneas emparejadas es mucho menor en nuestro algoritmo. Asimismo, el algoritmo 1-uncenter gana al maximin en todos los tiempos de cómputo. El algoritmo 1-uncenter se comporta incluso mejor que el procedimiento maximin cuando el número de aristas m es $O(n)$. En este caso particular, la diferencia entre ambos algoritmos es bastante grande.

En todos los casos, el número de aristas procesadas y el número de emparejamientos en nuestro algoritmo es menor que el de Melachrinoudis y Zhang, alcanzando en algunos casos una reducción del 50%. Como consecuencia directa de todo esto, los tiempos de cómputo del nuevo algoritmo son mejores, alcanzando en algunos casos una reducción del 80%. Además, la reducción aumenta a medida que aumenta el número de nodos n.

V.8 Observaciones finales

Se ha estudiado la localización de un servicio no deseado bajo el criterio max-min. Como se estableció en la introducción, existen pocos trabajos dedicados a este problema en la literatura. Uno de los más recientes es debido a Melachrinoudis y Zhang (1999), quienes propusieron un algoritmo en tiempo $O(mn)$ basado en tres cotas superiores y en una modificación del procedimiento de Dyer (1984). Sin embargo, hemos mostrado que sus cotas superiores pueden ser ajustadas, y que el emparejamiento de líneas superfluas no es necesario. El trabajo de Berman y Drezner (2000) enfoca el problema desde un punto de vista de programación lineal. Aunque tiene la misma complejidad teórica, sus tiempos de cómputo son muy elevados debido a que el algoritmo tiene que procesara cada arista.

Por tanto, usando cotas más ajustadas y eliminando el emparejamiento de líneas superfluas por medio de una formulación del problema más adecuada, proponemos un nuevo algoritmo en tiempo $O(mn)$. Como resultado de todo esto, el algoritmo propuesto es más directo y sus tiempos de cómputo son más rápidos que los ya reportados por Melachrinoudis y Zhang (1999).

Capítulo VI (Resumen)

Los problemas de localización en redes de la mediana no deseada y del anti-cent-dian

"Los trabajos sobre modelos de localización de servicios no deseados representan uno de los mayores campos de investigación en la actualidad"
H.A. EISELT & G. LAPORTE

VI.1 Introducción

Normalmente, los servicios a ser localizados se consideran "deseables" para los clientes, por ejemplo, los centros comerciales, los servicios de emergencia, los colegios, etc. Sin embargo, hay algunos servicios que no son tan deseables, y pueden ser considerados como una molestia (*desagradable*), por ejemplo vertederos, plantas petrolíferas o prisiones. Algunos de ellos pueden ser incluso dañinos (*nocivo*) para la población circundante, por ejemplo, reactores nucleares, industrias químicas y plantas contaminantes. De todos modos, los consideramos todos "indeseables".

La literatura en localización no deseada en redes empezó a mediados de los 1970 con Church y Garfinkel (1978), quienes definieron y resolvieron el problema 1-maxisum (maxian) en tiempo $O(mn \log n)$, siendo n el número de nodos y m el número de aristas. Más tarde, Minieka (1983) estudió el *anti-centro* (maxmax) y la *anti-mediana* (maxsum), la cual es un enfoque similar al caso no pesado descrito en Church y Garfinkel (1978). Poco después, Ting (1984) desarrolló un algoritmo en tiempo lineal para el problema 1-maxisum en árboles.

Tamir (1991) sugirió brevemente que el problema 1-maxisum podía ser resuelto en tiempo $O(mn)$ usando el algoritmo de Zemel (1984). Sin embargo, no tenemos referencia en la literatura de ningún trabajo que describa directamente tal algoritmo para el problema del 1-maxisum en redes. En este capítulo presentamos un nuevo algoritmo que resuelve este problema en tiempo $O(mn)$.

VI.2 Notación y propiedades generales

Sea $N = (V, E)$ una red simple (sin lazos ni múltiples aristas), no dirigida, finita y conexa con n nodos (vértices) $V = \{v_1, v_2, \ldots, v_n\}$, y m aristas $E = \{(v_s, v_t) : v_s, v_t \in V\}$, con $|E| = m$. Se define una función $w : V \to \mathbb{R}$, $w(v_i) = w_i \geq 0$, que denota el número de clientes situados en v_i.

Asumimos que no todos $w_i = 0$. Por otros lado, definimos una función $l : E \to \mathbb{R}_+$, $l(e) = l_e > 0$ que indica la longitud de la arista *e*.

Dado cualquier par de nodos $v_i, v_j \in V$, la *distancia* entre estos nodos $d(v_i, v_j)$ se define como la longitud del camino más corto entre v_i y v_j. Entonces, para cualquier $e = (v_s, v_t) \in E$ y dado un punto interno $x \in e$, la distancia entre *x* y un nodo v_i es

$$d(x, v_i) = \min\{x + d(v_s, v_i), l_e - x + d(v_t, v_i)\} \tag{VI.1}$$

El punto en *e* donde $d(x, v_i)$ alcanza su equilibrio, esto es, $x + d(v_s, v_i) = l_e - x + d(v_t, v_i)$, se denomina un *punto cuello de botella*:

$$b_i = \frac{d(v_t, v_i) - d(v_s, v_i) + l_e}{2} \tag{VI.2}$$

Sea $B_e = \bigcup_{v_i \in V} b_i$ el conjunto de todos los puntos cuello de botella sobre la arista *e*, y sea $B_N = \bigcup_{e \in E} B_e$ el conjunto de todos los puntos cuello de botella en la red *N*.

Dado cualquier punto *x* en la red *N*, definimos

$$f(x) = \sum_{v_i \in V} w_i d(x, v_i) \tag{VI.3}$$

como la suma de distancias pesadas desde el punto *x* a todos los nodos de la red.

El *problema maxian* se expresa como

$$\max_{x \in N} f(x) \tag{VI.4}$$

y un punto $x_N \in N$ es un punto *maxian* si y solo si $f(x_N) = \max_{x \in N} f(x)$. Las propiedades de este problema fueron descritas en Church y Garfinkel (1978).

El problema (VI.4) puede ser formulado sobre cada arista *e* como sigue:

$$f(x_e) = \max_{x \in e} f(x) \tag{VI.5}$$

y un punto $x_N \in N$ es un punto maxian si y solo si $f(x_N) = \max_{e \in E} f(x_e)$.

Una evaluación directa de (VI.3) sobre todos estos puntos se puede llevar a cabo en tiempo $O(mn^2)$. A pesar de esto, en este capítulo presentamos un algoritmo que eficientemente resuelve el problema (VI.4) en tiempo $O(mn)$.

VI.3 Un nuevo enfoque

Dada una arista $e = (v_s, v_t) \in E$, para todos los nodos $v_i \in V$, sea $d_i = d(v_t, v_i) - d(v_s, v_i)$ la diferencia de las distancias pesadas desde los nodos v_s y v_t al nodo v_i. Obviamente de (VI.2) se cumple que $-l_e \le d_i \le l_e$. Usando d_i tenemos $b_i = (d_i + l_e)/2$. En particular, para $d = -l_e$, obtenemos $b_i = 0 = v_s$, mientras que para $d = l_e$, tenemos que $b_i = l_e = v_t$.

Definimos los siguientes conjuntos

$$A = \{v_i \in V : -l_e < d_i \le l_e\}, \quad B = \{v_i \in V : d_i = -l_e\}$$
$$C = \{v_i \in V : -l_e \le d_i < l_e\}, \quad D = \{v_i \in V : d_i = l_e\}$$

Remarcar que $B \subseteq C$, $D \subseteq A$ y $A \cup B = C \cup D = V$.

Sea $W = \sum_{v_i \in V} w_i$ la suma de todos los pesos, y sea W_s la pendiente derecha de la función $f(x)$ en el nodo v_s, esto es

$$W_s = \sum_{v_i \in A} w_i - \sum_{v_i \in B} w_i = W - 2\sum_{v_i \in B} w_i = 2\sum_{v_i \in A} w_i - W \tag{VI.6}$$

Asimismo, sea W_t el valor de la pendiente izquierda con signo opuesto de $f(x)$ en v_t,

$$W_t = \sum_{v_i \in C} w_i - \sum_{v_i \in D} w_i = 2\sum_{v_i \in C} w_i - W = W - 2\sum_{v_i \in D} w_i \tag{VI.7}$$

Obviamente, $W_s, W_t \le W$. Cuando $W_s \le 0$ o $W_t \le 0$, el problema (VI.5) se resuelve fácilmente usando el siguiente resultado.

Teorema VI.1. *Dada la arista $e = (v_s, v_t) \in E$, obtenemos una solución a* (VI.5) *en los siguientes casos:*

a) *Si $W_s = W_t = 0$, la solución es el intervalo $[v_s, v_t]$.*
b) *Si $W_s = 0$ y $W_t \ne 0$, la solución es el intervalo $[v_s, \min_{b_i \ne 0} b_i]$.*
c) *Si $W_t = 0$ y $W_s \ne 0$, la solución es el intervalo $[\max_{b_i \ne l_e} b_i, v_t]$.*
d) *Si $W_s < 0$ y $W_t \ne 0$ el punto óptimo es v_s.*
e) *Si $W_t < 0$ y $W_s \ne 0$ el punto óptimo es v_t.*

Desafortunadamente, cuando W_s y W_t son ambos estrictamente positivos, el problema (VI.5) no es tan sencillo de resolver. Sin embargo estos dos valores pueden ser usados para definir una nueva cota superior, lo cual permitirá simplificar la búsqueda.

VI.4 Cotas inferiores y superiores

En cualquier arista $e = (v_s, v_t) \in E$, una cota inferior simple $LB(e) = \max(f(v_s), f(v_t))$ fue propuesta por Church y Garfinkel (1978). También dieron una cota superior para el problema maxian no pesado, que puede ser usado para derivar una cota superior para el problema maxian pesado como sigue

$$UB(e) = \frac{f(v_s) + f(v_t) + Wl_e}{2} \tag{VI.8}$$

Esta cota se computa en tiempo $O(n)$. No obstante, esta cota se puede mejorar en el mismo tiempo computacional como sigue.

Consideramos W_s y W_t estrictamente positivos. Ahora, computamos el punto de intersección z tal que $f(v_s) + zW_s = f(v_t) + W_t(l_e - z)$, y su valor de ordenada.

$$z = \frac{f(v_t) - f(v_s) + W_t l_e}{W_s + W_t}, \qquad y(z) = \frac{W_s f(v_t) + W_t f(v_s) + W_s W_t l_e}{W_s + W_t}$$

Sea $NUB(e) = y(z)$ la nueva cota superior. Obviamente, como $f(x)$ es una función cóncava, $f(x) \le NUB(e), \forall x \in e$. Para demostrar que la nueva cota es tan buena como (VI.8) necesitamos primero establecer el siguiente Lema.

Lema VI.1. $f(v_t) \le f(v_s) + W_s l_e$.

Proposición VI.1. *Para cualquier arista $e = (v_s, v_t) \in E$, $NUB(e) \le UB(e)$.*

A pesar de que estas dos cotas $UB(e)$ y $NUB(e)$ son iguales cuando $W_s = W_t = W$, existe un caso especial en cual podemos determinar la mínima diferencia entre ambas. Si la distancia entre los nodos de una arista es igual a su longitud, entonces podemos establecer el siguiente resultado.

Corolario VI.1. *Dada cualquier arista $e = (v_s, v_t) \in E$ tal que $d(v_s, v_t) = l_e$, la mínima diferencia entre $NUB(e)$ y $UB(e)$ es*

$$\frac{(w_s - w_t)(f(v_s) - f(v_t)) + (W(w_s + w_t) - 4w_s w_t) l_e}{2(W - w_s - w_t)}$$

El método descrito en la siguiente sección hace uso de esta nueva cota $NUB(e)$. Más aún, esta cota será actualizada en cada iteración del procedimiento de búsqueda. Por tanto, podemos definir la nueva cota superior en la arista *e* como una función G_{UB} de cinco parámetros:

$$G_{UB}(e, F_j, W_j, F_k, W_k) = \frac{W_j F_k + W_k F_j + W_j W_k l_e}{W_j + W_k}$$

Así, $NUB(e) = G_{UB}(e, f(v_s), W_s, f(v_t), W_t)$.

VI.5 El método propuesto cuando W_s y W_t son estrictamente positivos

En esta sección veremos cómo obtener los puntos óptimos en tiempo $O(mn)$ cuando W_s y W_t son estrictamente positivas. Sea $e = (v_s, v_t) \in E$. Comenzamos reemplazando(VI.1) en (VI.3) para obtener

$$f(x) = \sum_{v_i \in V} w_i \min\{x + d(v_s, v_i), l_e - x + d(v_t, v_i)\}$$

Dado un punto *x* en *e*, los siguientes dos conjuntos son definidos:

$$L(x) = \{v_i \in V : b_i < x\}, \qquad R(x) = \{v_i \in V : b_i \geq x\}$$

La función $f(x)$ se divide entonces en dos sumandos

$$f(x) = \sum_{L(x)} w_i (l_e + d(v_t, v_i)) + \sum_{R(x)} w_i d(v_s, v_i) + x\left(\sum_{R(x)} w_i - \sum_{L(x)} w_i\right) \tag{VI.9}$$

La diferencia $\sum_{R(x)} w_i - \sum_{L(x)} w_i$ son los diferentes valores de las sucesivas pendientes de $f(x)$. Sea $W_L(x) = \sum_{L(x)} w_i$ y como $\sum_{R(x)} w_i + \sum_{L(x)} w_i = W$ entonces $\sum_{R(x)} w_i = W - W_L(x)$. Reemplazamos este valor en (VI.9), y sea $H(x)$ igual a los dos primeros sumandos,

$$f(x) = H(x) + x(W - 2W_L(x))$$

Para cualquier $x \in e$, la función $H(x)$ es siempre positiva. Podemos evaluar $W - 2W_L(x)$ en varios puntos particulares *x* para comprobar si $f(x)$ crece, decrece o permanece constante. Los puntos a evaluar son el conjunto de puntos cuello de botella B_e.

Sea $l = 1$ y $r = n$ los índices inferior y superior en B_e, respectivamente. Sea d_q el valor mediana de todas las diferencias d_i ($l \leq i \leq r$). Sean b_q y w_q, respectivamente, el punto cuello de botella y el peso relacionado con d_q.

Sea $W_L(b_q) = \sum_{L(b_q)} w_i$. Como $b_i < b_q$ para $l \le i < q$, podemos hacer $W_L = W_L(b_q) = \sum_{i=l}^{q-1} w_i$. Asimismo, sea $W_R = \sum_{i=q+1}^{r} w_i = W - W_L - w_q$. Siguiendo el análisis anterior, alcanzamos el siguiente resultado.

Teorema VI.2. *Existe una solución a* (VI.5) *en los siguientes tres casos:*

a) Si $W_L + w_q = W_R$, entonces la solución es $[b_q, \min_{q<i\le r} b_i]$.

b) *Si* $W_L = W_R + w_q$, *entonces la solución es* $[\max_{l\le i<q} b_i, b_q]$.

c) *Si* $W_R - w_q < W_L < W_R + w_q$, *entonces la solución es el punto* b_q.

Estos tres casos son mutuamente excluyentes, aunque existen dos alternativas más en las cuales la solución no es obtenida porque el valor máximo no se alcanza en el punto b_q.

d) $W_L + w_q < W_R$: la función $f(x)$ es creciente en el punto b_q. Todos los puntos b_i tal que $l \le i \le q$ pueden ser descartados. La búsqueda continúa con $l = q+1$.

e) $W_L > W_R + w_q$: implica que la función $f(x)$ es decreciente y, por tanto, todos los puntos b_i con $q \le i \le r$ pueden ser eliminados. La búsqueda continua con $r = q-1$.

Además, introducimos una mejora mediante la actualización dinámica de la nueva cota superior $NUB(e)$ sobre el punto b_q en cada iteración. De esta forma, el proceso de búsqueda puede terminarse tan pronto como el valor de $NUB(e)$ es menor que el óptimo global ya almacenado.

VI.6 El nuevo algoritmo

El cálculo dinámico de la nueva cota usando el punto b_q es llevado a cabo por la función $G_{UB}(e, F_j, W_j, F_k, W_k)$. Los valores F_j y F_k dependen de $f(b_q)$, el cual puede ser obtenido de (VI.9). Reemplazando $f_L(b_q) = \sum_{L(b_q)} w_i d(v_t, v_i)$ y $f_R(b_q) = \sum_{R(b_q)} w_i d(v_s, v_i)$ obtenemos

$$f(b_q) = f_L(b_q) + f_R(b_q) + b_q(W - W_L(b_q)) + (l_e - b_q)W_L(b_q)$$

Si la nueva mediana es calculada en la siguiente iteración, por ejemplo d_p con punto cuello de botella b_p, entonces el valor de $f(b_p)$ puede ser determinado de $f(b_q)$ de la siguiente forma:

- Si $b_p < b_q$, entonces $f_L(b_p) = f_L(b_q) - \sum_{i=p}^{r} w_i d(v_t, v_i)$ y $f_R(b_p) = f_R(b_q) + \sum_{i=p}^{r} w_i d(v_s, v_i)$, por lo que $f_L(b_p) + f_R(b_p) = f_L(b_q) + f_R(b_q) + \sum_{i=p}^{r} w_i (d(v_s, v_i) - d(v_t, v_i)) = f_L(b_q) + f_R(b_q) - \sum_{i=p}^{r} w_i d_i$.
- Si $b_p > b_q$, entonces $f_L(b_p) = f_L(b_q) + \sum_{i=l}^{p-1} w_i d(v_t, v_i)$ y $f_R(b_p) = f_R(b_q) - \sum_{i=l}^{p-1} w_i d(v_s, v_i)$, por lo que $f_L(b_p) + f_R(b_p) = f_L(b_q) + f_R(b_q) + \sum_{i=l}^{p-1} w_i (d(v_t, v_i) - d(v_s, v_i)) = f_L(b_q) + f_R(b_q) + \sum_{i=l}^{p-1} w_i d_i$.

El cálculo de $W_L(b_p)$ se realiza de la misma forma:

$$W_L(b_p) = W_L(b_q) + \begin{cases} -\sum_{i=p}^{r} w_i, & \text{if } b_p < b_q \\ \sum_{i=l}^{p-1} w_i, & \text{if } b_p > b_q \end{cases}$$

Finalmente, cuando se cumplen los casos d) o e), los valores F_j, W_j y F_k, W_k deben ser actualizados como corresponde:

- Si se cumple d), actualizar $W_j = W - 2(W_L(b_q) + w_q)$ y $F_j = f(b_q) - W_j b_q$. Además $W_L = W_L + w_q$ y $f(b_q) = f(b_q) + w_q d_q$.
- En otro caso, actualizar $W_k = 2W_L(b_q) - W$ y $F_k = f(b_q) - W_k(l_e - b_q)$, dejando W_L y $f(b_q)$ intactos.

Asimismo, los valores de *l* y *r* también son actualizados, y de este modo, podemos eliminar la mitad de los valores d_i. Este resultado demuestra el siguiente Lema.

Lema VI.2. *En cada iteración del bucle 'while', se eliminan* $q = (l + r)/2$ *puntos de* B_e.

Este Lema ayuda a demostrar la complejidad total del nuevo algoritmo.

Teorema VI.3. *Asumiendo que la matriz de distancias ya está calculada, el nuevo algoritmo resuelve el problema de la 1-mediana en redes en tiempo* $O(mn)$.

VI.6.1 El caso no pesado

Cuando todos los nodos v_i tienen el mismo peso $w_i = w$, la red subyacente puede ser considerada como no pesada. El siguiente resultado establece que el nuevo algoritmo resuelve directamente el caso no pesado también en tiempo $O(mn)$.

Proposición VI.2. *Si todos los pesos* w_i, $\forall v_i \in V$ *son iguales, entonces o se cumple el Teorema VI.1, o sólo los casos b) o c) del Teorema VI.2 se cumplen en la primera iteración del bucle 'while'.*

VI.7 Un ejemplo

Consideramos una red con $n = 7$ nodos y $m = 15$ aristas. Los pesos de los nodos son enteros aleatoriamente generados entre 1 y 9, mientras que las longitudes de las aristas varían entre 1 y 25. El peso total *W* es igual a 24.

Debido a la nueva cota superior $NUB(e)$, sólo se han procesado 8 de las 15 aristas totales. Si hubiésemos usado $UB(e)$ el algoritmo hubiera procesado 13 aristas. Esta mejora permite un substancial ahorro de tiempo con respecto al algoritmo de Church y Garfinkel.

VI.8 Resultados computacionales

A pesar de que Tamir (1991) comentó brevemente que se podía obtener una solución al problema 1-maxisum sobre redes en tiempo $O(mn)$ usando los algoritmos generales propuestos por Zemel (1984), el procedimiento no está directamente descrito. Por ello, decidimos comparar el nuevo algoritmo con el propuesto por Church y Garfinkel (1978). Además, le añadimos la versión de la cota pesada $UB(e)$ para hacer la comparación lo más justa posible.

En todos los casos, el nuevo algoritmo es mucho más rápido que el de Church y Garfinkel. Con respecto al número de aristas, los valores iniciales de las cotas $UB(e)$ y $NUB(e)$ son prácticamente iguales. Esto significa que, en el comienzo, descartan el mismo número de aristas. En el caso de redes planares, el valor inicial de $NUB(e)$ es mucho mejor que el de $UB(e)$, y por tanto, el nuevo algoritmo descarta muchas más aristas.

VI.9 Combinando el uncenter con el maxian: un algoritmo mejorado para el problema anti-cent-dian

En las secciones previas hemos estudiado el problema 1-uncenter (maximin) y el problema 1-maxian (maxisum) en redes. Ahora vamos a combinar estos dos objetivos para obtener el modelo denominado *anti-cent-dian*.

El modelo anti-cent-dian en redes considera la combinación convexa de los criterios maximin y maxisum. Moreno-Pérez y Rodríguez-Martín (1999) desarrollaron dos algoritmos que proporcionan, respectivamente, la localización óptima para un valor fijo de λ, el cual determina la combinación convexa, y el conjunto de localizaciones óptimas para todas las combinaciones convexas. Los dos se ejecutan en tiempo $O(mn \log n)$. En las siguientes secciones mostramos que la complejidad del primer algoritmo puede reducirse a $O(mn)$.

VI.9.1 Notación y propiedades

Sea $N=(V,E)$ una red simple, no dirigida y conexa con n nodos (vértices) $V=\{v_1,v_2,\ldots,v_n\}$, y m aristas $E=\{(v_s,v_t):v_s,v_t \in V\}$, con $|E|=m$. Para mayor simplicidad, seguimos la misma notación introducida en la sección VI.2.

Sea Q_e el conjunto de puntos $x \in e$ tal que, para cualesquiera dos nodos distintos $v_i,v_j \in V$, $d(x,v_i)=d(x,v_j)$ y además, $d(x,v_i)$ y $d(x,v_j)$ no decrecen simultáneamente cuando x es perturbado ligeramente en cualquier dirección. Sea $Q_N=\bigcup_{e\in E} Q_e$.

Ahora definimos la función uncenter no pesado (maximin) y la función maxian (maxisum). Dado un punto x en la red N, definimos

$$f_{\min}(x)=\min_{v_i \in V} d(x,v_i)$$

como la mínima distancia no pesada desde del punto x al resto de nodos de la red. Recuérdese que un punto $y_N \in N$ es un punto uncenter si y solo si $f_{\min}(y_N)=\max_{x\in N} f_{\min}(x)$. Cuando todos los pesos de los nodos w_i son iguales, el punto y_N para una arista $e=(v_s,v_t)$ es $y_e=l_e/2$, y de aquí $f_{\min}(y_e)=l_e/2$.

Por otro lado, dado $W=\sum_{v_i\in V} w_i$ y un punto $x\in N$, definimos

$$f_{\text{sum}}(x)=\frac{1}{W}\sum_{v_i\in V} w_i d(x,v_i)$$

como la suma promedio de las distancias pesadas desde el punto x al resto de nodos de la red. El punto maxian local en la arista e se denota por z_e.

Finalmente, la función *anti-cent-dian* se define como

$$f_{\mathrm{acd}}(\lambda, x) = \lambda f_{\min}(x) + (1-\lambda) f_{\mathrm{sum}}(x) \qquad \text{(VI.10)}$$

y cualquier punto $x_N \in N$ maximizando $f_{\mathrm{acd}}(\lambda, x)$ para un valor particular de λ, $0 \le \lambda \le 1$, se llama un punto *λ-anti-cent-dian*. En particular, si $\lambda = 0$, el anti-cent-dian es igual al maxian; mientras que para $\lambda = 1$, obtenemos el uncenter.

Combinando las propiedades de los problemas uncenter y maxian obtenemos las propiedades originalmente establecidas en Moreno-Pérez y Rodríguez-Martín (1999). Como las funciones objetivo maxisum y maximin son ambas cóncavas, podemos derivar una nueva propiedad referente al conjunto de puntos candidatos dentro de una arista.

Propiedad VI.8. *Sea $e = (v_s, v_t) \in E$, y_e el punto uncenter en la arista e y $[a,b]$ los puntos maxian. Dado un valor de λ, $0 \le \lambda \le 1$, los puntos anti-cent-dian de dicha arista residen dentro del intervalo* $[\min(y_e, a), \max(y_e, b)]$.

El problema (VI.10) puede ser formulado sobre cada arista e como sigue:

$$f_{\mathrm{acd}}(\lambda, x_e) = \max_{x \in e} f_{\mathrm{acd}}(\lambda, x) \qquad \text{(VI.11)}$$

y un punto $x_N \in N$ es un punto λ-anti-cent-dian si y solo si $f_{\mathrm{acd}}(\lambda, x_N) = \max_{e \in E} f(\lambda, x_e)$.

Moreno-Pérez y Rodríguez-Martín (1999) presentaron un procedimiento en $O(mn \log n)$ para obtener el punto anti-cent-dian cuando λ se fija a un valor particular. No obstante, podemos alcanzar un algoritmo en tiempo $O(mn)$.

VI.9.2 Análisis del problema y nueva cota superior

Sea $e = (v_s, v_t) \in E$ una arista. Cuando $\lambda = 1$, las solución a (VI.11) es $x_e = y_e$. Por otro lado, si $\lambda = 0$ entonces $x_e = z_e$. Por tanto, el análisis se centrará en el caso $0 < \lambda < 1$.

Como la función anti-cent-dian es una combinación convexa de las funciones $f_{\min}(x)$ y $f_{\mathrm{sum}}(x)$, las pendientes derecha e izquierda de $f_{\mathrm{acd}}(\lambda, x)$ en los nodos v_s y v_t deberían ser, respectivamente, como sigue

$$W'_s = \lambda + (1-\lambda)\frac{W_s}{W}, \qquad W'_t = \lambda + (1-\lambda)\frac{W_t}{W} \qquad \text{(VI.12)}$$

Como $W_s, W_t \le W$, entonces $W'_s, W'_t \le 1$. Si $W'_s \le 0$ o $W'_t \le 0$, el problema (VI.11) se puede resolver fácilmente usando el siguiente resultado.

Teorema VI.4. *Dado un valor de λ, $0 \le \lambda \le 1$, y dada una arista $e = (v_s, v_t) \in E$, podemos obtener una solución a* (VI.11) *en los siguientes casos:*

a) *Si $\lambda = W_s/(W_s - W) = W_t/(W_t - W)$, la solución es el intervalo $[v_s, v_t]$.*
b) *Si $\lambda = W_s/(W_s - W) \ne W_t/(W_t - W)$, la solución es el intervalo $[v_s, \min\{y_e, \min_{b_i \ne 0} b_i\}]$.*
c) *Si $\lambda = W_t/(W_t - W) \ne W_s/(W_s - W)$, la solución es el intervalo $[\max\{y_e, \max_{b_i \ne l_e} b_i\}, v_t]$.*
d) *Si $\lambda > W_s/(W_s - W)$ y $\lambda \ne W_t/(W_t - W)$ el punto óptimo es v_s.*
e) *Si $\lambda > W_t/(W_t - W)$ y $\lambda \ne W_s/(W_s - W)$ el punto óptimo es v_t.*

Vamos a mejorar la cota superior propuesta por Moreno-Pérez y Rodríguez-Martín (1999):

$$UB(\lambda,e)=\lambda UB_{\min}(e)+(1-\lambda)UB_{\text{sum}}(e) \tag{VI.13}$$

con $UB_{\text{sum}}(e)=(f_{\text{sum}}(v_s)+f_{\text{sum}}(v_t)+l_e)/2$ y $UB_{\min}(e)=(f_{\min}(v_s)+f_{\min}(v_t)+l_e)/2$. Dado que $UB_{\min}(x)=l_e/2$, reemplazamos $UB_{\text{sum}}(e)$ y $UB_{\min}(e)$ en (VI.13) para obtener

$$UB(\lambda,e)=\lambda\frac{l_e}{2}+(1-\lambda)\frac{f_{\text{sum}}(v_s)+f_{\text{sum}}(v_t)+l_e}{2}=\frac{(1-\lambda)(f_{\text{sum}}(v_s)+f_{\text{sum}}(v_t))+l_e}{2} \tag{VI.14}$$

Esta cota se calcula en $O(n)$, pero podemos mejorarla en el mismo tiempo computacional tal y como sigue. Asumimos que W_s' y W_t' son estrictamente positivos. El punto de intersección x tal que $f_{\text{acd}}(\lambda,v_s)+xW_s'=f_{\text{acd}}(\lambda,v_t)+W_t'(l_e-x)$, y su valor de ordenada $y(x)$ son:

$$x=\frac{f_{\text{acd}}(\lambda,v_t)-f_{\text{acd}}(\lambda,v_s)+W_t'l_e}{W_s'+W_t'},\qquad y(x)=\frac{f_{\text{acd}}(\lambda,v_t)W_s'+f_{\text{acd}}(\lambda,v_s)W_t'+W_s'W_t'l_e}{W_s'+W_t'}$$

Reemplazando $f_{\text{acd}}(\lambda,v_s)$ y $f_{\text{acd}}(\lambda,v_t)$ por, respectivamente, $(1-\lambda)f_{\text{sum}}(v_s)$ y $(1-\lambda)f_{\text{sum}}(v_t)$ produce

$$y(x)=\frac{(1-\lambda)(f_{\text{sum}}(v_t)W_s'+f_{\text{sum}}(v_s)W_t')+W_s'W_t'l_e}{W_s'+W_t'}$$

Sea $NUB(\lambda,e)=y(x)$ la nueva cota superior. Como $f_{\text{acd}}(\lambda,x)$ es una función cóncava, obviamente $f_{\text{acd}}(\lambda,x)\le NUB(\lambda,e)$, $\forall x\in e, 0\le\lambda\le 1$. Para poder demostrar que la nueva cota superior es tan buena como la (VI.14), necesitamos en primer lugar establecer el siguiente Lema.

Lema VI.3. $(1-\lambda)f_{\text{sum}}(v_t)\le(1-\lambda)f_{\text{sum}}(v_s)+W_s'l_e$.

Proposición VI.3. *Para cualquier arista* $e=(v_s,v_t)\in E$, $NUB(\lambda,e)\le UB(\lambda,e)$.

Denotamos la cota superior como una función G_{UB} de seis parámetros:

$$G_{UB}(\lambda,e,F_j,W_j,F_k,W_k)=\frac{(1-\lambda)(W_jF_k+W_kF_j)+W_jW_kl_e}{W_j+W_k}$$

Así, $NUB(\lambda,e)=G_{UB}(\lambda,e,f_{\text{sum}}(v_s),W_s',f_{\text{sum}}(v_t),W_t')$.

VI.9.3 Resolviendo el problema anti-cent-dian

Mostramos ahora cómo se puede resolver el problema anti-cent-dian para un valor particular de λ, $0<\lambda<1$, cuando $W_s'>0$ y $W_t'>0$.

Podemos formular la función $f_{\text{sum}}(x)$ como

$$f_{\text{sum}}(x)=H(x)+\frac{x}{W}\left(\sum_{R(x)}w_i-\sum_{L(x)}w_i\right)$$

Por otro lado, tenemos que $f_{\min}(x)=\min_{v_i\in V}d(x,v_i)$. Por tanto, para cualquier $x\in e$ tenemos que

$$f_{\min}(x)=\begin{cases}x & \text{if } x\le y_e\\ l_e-x & \text{if } x>y_e\end{cases} \tag{VI.15}$$

Finalmente, la función anti-cent-dian $f_{\text{acd}}(\lambda,x)$ sobre una arista e se define como

$$f_{\text{acd}}(\lambda,x)=(1-\lambda)\left(H(x)+\frac{x}{W}\left(\sum_{R(x)}w_i-\sum_{L(x)}w_i\right)\right)+\lambda\begin{cases}x & \text{if } x\le y_e\\ l_e-x & \text{if } x>y_e\end{cases} \tag{VI.16}$$

Podemos evaluar la pendiente de $f_{\text{acd}}(\lambda,x)$ en un punto particular x para comprobar si crece, decrece o permanece constante. Los puntos a evaluar son el conjunto de puntos $B_e\cup\{y_e\}$.

Sea $B_e'=B_e\cup\{y_e\}$ con $|B_e|=n$ y $b_{n+1}=y_e$ ($d_{n+1}=0$), $w_{n+1}=0$. Sean $l=1$ y $r=n+1$, respectivamente, los índices inferior y superior en B_e'. Sea d_q el valor mediana de todos los d_i ($l\le i\le r$). Sean b_q y w_q, respectivamente, el punto cuello de botella y el peso relacionado con d_q.

Ahora centramos el análisis en el punto mediana b_q. Sea $W_L=W_L(b_q)=\sum_{L(b_q)}w_i=\sum_{i=l}^{q-1}w_i$. Además, sea $W_R=\sum_{i=q+1}^{r}w_i=W-W_L-w_q$.

Ahora necesitamos definir nuevas variables para el problema anti-cent-dian:

$$W_L'=(1-\lambda)\frac{W_L}{W},\quad W_R'=(1-\lambda)\frac{W_R}{W},\quad w_q'=(1-\lambda)\frac{w_q}{W} \tag{VI.17}$$

Podemos expresar le pendiente izquierda en punto particular b_q como

$$(W_R'+w_q')-W_L'+\lambda\alpha_q \tag{VI.18}$$

siendo $\alpha_q=\begin{cases}1, & \text{if } b_q\le y_e\\ -1, & \text{if } b_q>y_e\end{cases}$.

mientras que la pendiente derecha es

$$W_R'-(W_L'+w_q')+\lambda\beta_q \tag{VI.19}$$

siendo $\beta_q=\begin{cases}1, & \text{if } b_q<y_e\\ -1, & \text{if } b_q\ge y_e\end{cases}$.

Ahora definimos las nuevas variables W_L^*, W_R^*, y w_q^* como sigue:

- Si $b_q<y_e$ ($d_q<0$) entonces sea $W_R^*=W_R'+\lambda$, $W_L^*=W_L'$ y $w_q^*=w_q'$.
- Si $b_q>y_e$ ($d_q>0$) entonces sea $W_L^*=W_L'+\lambda$, $W_R^*=W_R'$ y $w_q^*=w_q'$.
- En otro caso ($d_q=0$), sea $W_L^*=W_L'$, $W_R^*=W_R'$ y $w_q^*=w_q'+\lambda$.

Siguiendo el análisis dado más arriba, obtenemos el siguiente resultado.

Lema VI.4. *La pendiente izquierda de la función $f_{\text{acd}}(\lambda,x)$ en el punto b_q es $1-2W_L^*$, mientras que la pendiente derecha es $1-2(W_L^*+w_q^*)$.*

Usando el Lema anterior, el siguiente resultado caracteriza la solución óptima del problema anti-cent-dian.

Teorema VI.5. *Existe una solución a* (VI.11) *en los siguientes tres casos:*

a) *Si $W_L^*+w_q^*=W_R^*$, entonces la solución es* $[b_q,\min_{q<i\le r}b_i]$.

b) *Si $W_L^*=W_R^*+w_q^*$, entonces la solución es* $[\max_{l\le i<q}b_i,b_q]$.

c) *Si* $W_R^* - w_q^* < W_L^* < W_R^* + w_q^*$, *entonces la solución es el punto* b_q.

Estos tres casos son mutuamente excluyentes, aunque existen dos alternativas más.

d) $W_L^* + w_q^* < W_R^*$: la función $f_{\text{acd}}(\lambda, x)$ es creciente en el punto b_q. Todos los puntos b_i tal que $l \le i \le q$ pueden ser descartados. La búsqueda continúa con $l = q+1$.

e) $W_L^* > W_R^* + w_q^*$: implica que $f_{\text{acd}}(\lambda, x)$ es decreciente y por tanto todos los puntos b_i con $q \le i \le r$ pueden ser eliminados. La búsqueda continúa con $r = q-1$.

Introducimos una mejora al actualizar dinámicamente la nueva cota superior $NUB(\lambda, e)$ sobre cada punto b_q en cada iteración. De esta forma, el proceso de búsqueda puede terminarse tan pronto como el valor de $NUB(\lambda, e)$ es menor o igual al valor óptimo global.

VI.9.4 El algoritmo del anti-cent-dian para un valor particular de λ

El cálculo dinámico de la nueva cota usando el punto b_q se lleva a cabo por $G_{UB}(\lambda, e, F_j, W_j, F_k, W_k)$. Los valores F_j y F_k dependen del valor de $f_{\text{acd}}(\lambda, b_q)$, que es

$$f_{\text{acd}}(\lambda, b_q) = (1-\lambda) f_{\text{sum}}(b_q) + \lambda f_{\min}(b_q) =$$
$$= (1-\lambda)\left(\frac{1}{W}\sum_{L(b_q)} d(v_t, v_i) + \frac{1}{W}\sum_{R(b_q)} w_i d(v_s, v_i) + \frac{l_e}{W}\sum_{L(b_q)} w_i + \frac{b_q}{W}\left(\sum_{R(b_q)} w_i - \sum_{L(b_q)} w_i\right)\right) + \lambda f_{\min}(b_q)$$

Reemplazando $f_{\text{sum}}^L(b_q) = \sum_{L(b_q)} w_i d(v_t, v_i)/W$ y $f_{\text{sum}}^R(b_q) = \sum_{R(b_q)} w_i d(v_s, v_i)/W$ obtenemos

$$f_{\text{acd}}(\lambda, b_q) = (1-\lambda)(f_{\text{sum}}^L(b_q) + f_{\text{sum}}^R(b_q)) + l_e W_L' + b_q((W_R' + w_q') - W_L') + \lambda f_{\min}(b_q) =$$
$$= (1-\lambda)(f_{\text{sum}}^L(b_q) + f_{\text{sum}}^R(b_q)) + (W_R' + w_q')b_q + W_L'(l_e - b_q) + \lambda \begin{cases} b_q & \text{if } b_q \le y_e \\ l_e - b_q & \text{if } b_q > y_e \end{cases}$$

Si se calcula una nueva mediana en la siguiente iteración, por ejemplo d_p con punto cuello de botella b_p, el valor de $f_{\text{acd}}(\lambda, b_p)$ puede ser determinado a partir de $f_{\text{acd}}(\lambda, b_q)$ de una forma muy similar al problema maxian:

- Si $b_p < b_q$ entonces

$$f_{\text{sum}}^L(b_p) + f_{\text{sum}}^R(b_p) = f_{\text{sum}}^L(b_q) + f_{\text{sum}}^R(b_q) + \frac{1}{W}\sum_{i=p}^{r} w_i(d(v_s, v_i) - d(v_t, v_i)) =$$
$$= f_{\text{sum}}^L(b_q) + f_{\text{sum}}^R(b_q) - \frac{1}{W}\sum_{i=p}^{r} w_i d_i$$

- Si $b_p > b_q$ entonces

$$f_{\text{sum}}^L(b_p) + f_{\text{sum}}^R(b_p) = f_{\text{sum}}^L(b_q) + f_{\text{sum}}^R(b_q) + \frac{1}{W}\sum_{i=l}^{p-1} w_i(d(v_t, v_i) - d(v_s, v_i)) =$$
$$= f_{\text{sum}}^L(b_q) + f_{\text{sum}}^R(b_q) + \frac{1}{W}\sum_{i=l}^{p-1} w_i d_i$$

Asimismo, el cálculo de $W_L(b_p)$ se realiza de la misma forma al problema maxian. Finalmente, cada vez que se satisfacen los casos d) o e), los valores F_j, W_j y F_k, W_k deben actualizarse como corresponde:

- Si se cumple el caso d), actualizar $W_j = 1-2(W_L^* + w_q^*)$ y $F_j = f_{\text{acd}}(\lambda, b_q) - W_j b_q$. Debemos poner $W_L = W_L + w_q$ y $f_{\text{acd}}(\lambda, b_q) = f_{\text{acd}}(\lambda, b_q) + (1-\lambda) w_q d_q / W$.
- En otro caso, actualizar $W_k = 2W_L^* - 1$ y $F_k = f_{\text{acd}}(\lambda, b_q) - W_k(l_e - b_q)$, dejando W_L y $f_{\text{acd}}(\lambda, b_q)$ sin alterar.

Como en el problema maxian, cada iteración del bucle 'while' elimina $q = (l+r)/2$ puntos de B'_e. De este modo, la complejidad del algoritmo es la misma del problema maxian.

Teorema VI.6. *Asumiendo que la matriz de distancias está ya calculada, el nuevo algoritmo resuelve el problema del* λ *-anti-cent-dian en redes para un valor dado de* λ*,* $0 \le \lambda \le 1$*, en tiempo* $\mathrm{O}(mn)$.

VI.10 Conclusiones

El propósito principal de este capítulo es doble. En la primera parte, se analizó el problema de localización 1-maxisum (maxian) en redes. Se derivó una cota superior inicial $UB(e)$ del trabajo de Church y Garfinkel (1978), la cual fue mejorada con una nueva cota superior $NUB(e)$. Asimismo, esta cota puede ser actualizada dinámicamente sin incrementar el tiempo de cómputo total.

Hemos desarrollado un nuevo algoritmo en $\mathrm{O}(mn)$ que resuelve el problema. El procedimiento hace uso de la nueva cota superior, y por tanto, permite abandonar el proceso de búsqueda tan pronto como la cota superior es menor que el óptimo global. Este nuevo algoritmo ha sido comparado con el procedimiento de Church y Garfinkel (1978) incluyendo la cota inicial $UB(e)$, sobre redes de baja y alta densidad, así como en redes planares. En todos los casos, el nuevo algoritmo consigue un mejor comportamiento en los tiempos de cómputo.

Por otro lado, la segunda parte estudia el problema λ-anti-cent-dian. Se ha propuesto una nueva cota superior $NUB(\lambda, e)$, así como un nuevo algoritmo en $\mathrm{O}(mn)$ que mejora el método anterior en $\mathrm{O}(mn \log n)$ dado por Moreno-Pérez y Rodríguez-Martín (1999).

Capítulo VII (Resumen)

Problemas de localización de servicios no deseados en redes multicriterio

"El problema real de localizar un servicio no deseado es claramente un problema de decisión multiobjetivo"
E. ERKUT & S. NEUMAN

VII.1 Introducción

La mayor parte de la inmensa literatura en Análisis de Localización trata sobre el emplazamiento de servicios tales como centros comerciales, servicios de emergencia y centros educativos. Todas estos servicios son *deseables* (atractivos) para los habitantes cercanos quienes tratan de tenerlos lo más cerca posible.

Sin embargo, hay otros servicios tales como los vertederos, plantas químicas, reactores nucleares, instalaciones militares y plantas contaminantes (ruido/gas) que resultan ser *indeseables* (repulsivos) para la población circundante, que los evita e intenta permanecer lejos de ellos. En este sentido, Erkut y Neuman (1989) distinguen entre servicios *nocivos* (peligrosos) y *desagradables* (molestos), aunque ambos se pueden considerar simplemente como *indeseables*.

En este sentido, los modelos de localización no deseada analizados en capítulos previos son básicamente unicriterio, y estaban relacionados con los trabajos de Melachrinoudis y Zhang (1999), Berman y Drezner (2000), Minieka (1983), Church y Garfinkel (1978), y Tamir (1988, 1991). Sin embargo, Erkut y Neuman (1989) hicieron hincapié en la necesidad de enfoques multiobjetivo en el emplazamiento de servicios no deseados. Daskin (1995) y Zhang (1996) también apuntaron no sólo la necesidad de incluir múltiples criterios en los problemas de localización no deseada, pero también el hecho de que los investigadores han prestado poca atención a estos problemas y por tanto, se ha investigado muy poco en este campo tan prometedor.

Por consiguiente, en este capítulo presentamos un modelo de localización multicriterio de servicios no deseados en redes con varios pesos en los nodos y varias longitudes en las aristas, combinando los criterios maximin y maxisum mediante un parámetro λ. Este modelo puede ser descrito como el problema *λ-anti-cent-dian multicriteria* en redes.

VII.2 Notación y definiciones básicas

Sea $N=(V,E)$ una red no dirigida, simple y conexa, con el conjunto de nodos $V=\{v_1,v_2,\dots,v_n\}$, y siendo $E=\{(v_s,v_t):v_s,v_t\in V\}$ el conjunto de aristas. Sea p el número de pesos asociados con cada nodo, y q el número de longitudes (costes) fijadas en cada arista. Para cada vértice en V, definimos la siguiente función de pesos

$$\begin{array}{llll} w: & V & \longrightarrow & \mathbb{R}^p \\ & v_i\in V & \longrightarrow & w(v_i)=w_i=(w_i^1,\dots,w_i^p) \end{array}$$

De forma similar, sobre cada arista en E definimos la siguiente función de longitudes

$$\begin{array}{llll} l: & E & \longrightarrow & \mathbb{R}^q \\ & e=(v_s,v_t)\in E & \longrightarrow & l(e)=l_e=(l_e^1,\dots,l_e^q) \end{array}$$

Sea r un índice de longitud, con $1\le r\le q$, y sea $x\in e=(v_s,v_t)$ un punto dentro de e. Definimos $c_e^r(x,v_s)$ como la longitud del segmento de línea entre x y v_s con relación a la longitud r, con $0\le c_e^r(x,v_s)\le l_e^r$ y $c_e^r(x,v_t)=l_e^r-c_e^r(x,v_s)$. Para cualesquiera dos nodos $v_a,v_b\in V$, la distancia entre tales nodos, denotada por $d^r(v_a,v_b)$, se define como la longitud de cualquier camino mínimo en N que enlace v_a y v_b considerando la longitud r.

De la misma forma, dados cualquier punto $x\in N$ y un nodo $v_i\in V$, sea $d^r(x,v_i)=\min\{c_e^r(x,v_s)+d(v_s,v_i),c_e^r(x,v_t)+d(v_t,v_i)\}$ la distancia entre el punto x y el nodo v_i considerando la longitud r. El punto en la arista e donde $d^r(x,v_i)$ alcanza su equilibrio se denomina un *punto cuello de botella*, el cual se define como $b_i^r=(d^r(v_t,v_i)-d^r(v_s,v_i)+l_e^r)/2$. Dado un índice de longitud r, el conjunto de todos los puntos cuellos de botella en la arista e se denota por $B_e^r=\bigcup_{v_i\in V}b_i^r$, mientras que el conjunto de todos los puntos cuello de botella en la red N se denota por $B_N^r=\bigcup_{e\in E}B_e^r$.

Dado un índice de peso s y un índice de longitud r, sea Q_e^{sr} el conjunto de puntos $x\in e$ tal que, para dos nodos distintos $v_i,v_j\in V$, $w_i^s d^r(x,v_i)=w_j^s d^r(x,v_j)$ y además, $d^r(x,v_i)$ y $d^r(x,v_j)$ no decrecen simultáneamente cuando x es perturbado ligeramente en cualquier dirección. Sea $Q_N^{sr}=\bigcup_{e\in E}Q_e^{sr}$.

VII.3 El problema del uncenter multicriterio

Dado cualquier punto $x\in N$, y cualquier peso s $(1\le s\le p)$ y longitud r $(1\le r\le q)$, sea $f_{\min}^{sr}(x)=\min_{v_i\in V}w_i^s d^r(x,v_i)$ la mínima distancia pesada de x al conjunto de nodos. Dada una arista $e\in E$, un punto $y_e^{sr}\in Q_e^{sr}$ es un *punto uncenter local* en e si y sólo si $f_{\min}^{sr}(y_e^{sr})=\max_{x\in e}f_{\min}^{sr}(x)$, para cualquier par de valores (s,r), con $1\le s\le p$ y $1\le r\le q$. Asimismo, un punto $y_N^{sr}\in Q_N^{sr}$ es un *punto uncenter global* si y sólo si $f_{\min}^{sr}(y_N^{sr})=\max_{x\in N}f_{\min}^{sr}(x)=\max_{e\in E}f_{\min}^{sr}(y_e^{sr})$, para cualquier valor de los índices s y r.

Dado un punto $x\in N$, sea $F_{\min}(x)=(f_{\min}^{11}(x),f_{\min}^{12}(x),\dots,f_{\min}^{pq}(x))\in\mathbb{R}^{p\times q}$ los vectores de valores de la función uncenter $f_{\min}^{sr}(x)$ para todas las combinaciones de pesos $s=1,\dots,p$ y longitudes $r=1,\dots,q$. De ahora en adelante denotamos las funciones uncenter por $f_{\min}^{i}(x)$, con $i=1,\dots,k$.

Un conjunto de puntos $Y_N \subset N$ es un conjunto eficiente para el problema uncenter multicriterio si y sólo si $F_{\min}(Y_N) = \max_{x \in N} F_{\min}(x)$. Así, dados dos puntos $x, y \in N$, decimos que x domina a y, y se denota por $x \succ y$, si $f^i_{\min}(x) \geq f^i_{\min}(y)$, $\forall i = 1, \ldots, k$, con al menos una de las desigualdades estricta. Entonces, un punto $x \in N$ es un punto *eficiente* o *Pareto óptimo* para el problema del uncenter multicriterio si no existe otro punto $y \in N$ tal que $y \succ x$.

Dada una arista $e = (v_s, v_t) \in E$, sean y^1_e y y^2_e los puntos uncenter locales para función cada objetivo $f^i_{\min}(x)$, $i = 1, 2$.

Lema VII.1. *Si* $y^1_e \neq y^2_e$, *entonces el conjunto de puntos eficientes locales en la arista e es* $Y_e = [\min\{y^1_e, y^2_e\}, \max\{y^1_e, y^2_e\}]$.

Corolario VII.1. *Todos los puntos que pertenecen a los intervalos* $[v_s, \min\{y^1_e, y^2_e\})$ *y* $(\max\{y^1_e, y^2_e\}, v_t]$ *son puntos ineficientes.*

Corolario VII.2. *Si* $y^1_e = y^2_e$ *entonces el único punto eficiente local en la arista e el punto* $Y_e = y^1_e = y^2_e$.

Sean $y^1_N, y^2_N \in N$ los puntos uncenter globales para cada función objetivo.

Proposición VII.1. *La arista e no contiene puntos eficientes y, por tanto, puede ser descartada si algún punto* y^i_N, $1 \leq i \leq 2$, *satisface* $f^1_{\min}(y^1_e) \leq f^1_{\min}(y^i_N)$ *y* $f^2_{\min}(y^2_e) \leq f^2_{\min}(y^i_N)$, *con al menos una desigualdad estricta.*

Dada una arista $e = (v_s, v_t) \in E$, sea y^i_e el punto uncenter local de cada función objetivo $f^i_{\min}(x)$, $i = 1, \ldots, k$. Si todos los puntos y^i_e son iguales, entonces es obvio que el punto eficiente es $Y_e = y^1_e = \cdots = y^k_e$. En otro caso, las siguientes propiedades se verifican.

Lema VII.2. *El conjunto de puntos eficientes locales en la arista e es* $Y_e = [\min_{1 \leq i \leq k} y^i_e, \max_{1 \leq i \leq k} y^i_e]$.

Corolario VII.3. *Todos los puntos que pertenecen a los intervalos* $[v_s, \min_{1 \leq i \leq k} y^i_e)$ *y* $(\max_{1 \leq i \leq k} y^i_e, v_t]$ *son puntos ineficientes.*

Proposición VII.2. *Si algún punto uncenter global* y^i_N, $1 \leq i \leq k$, *satisface*

$$f^1_{\min}(y^1_e) \leq f^1_{\min}(y^i_N) \quad \wedge \quad f^2_{\min}(y^2_e) \leq f^2_{\min}(y^i_N) \quad \wedge \quad \ldots \quad \wedge \quad f^k_{\min}(y^k_e) \leq f^k_{\min}(y^i_N)$$

con al menos una desigualdad estricta, entonces la arista e no contiene puntos eficientes y por tanto, puede ser eliminada.

VII.4 El problema del maxian multicriterio

Dado cualquier punto $x \in N$, definimos la función $f^{sr}_{\text{sum}}(x) = \sum_{v_i \in V} w^s_i d^r(x, v_i)$ como la suma de distancias pesada del punto x al conjunto de nodos, con $1 \leq s \leq p$ y $1 \leq r \leq q$. Sobre cada arista $e \in E$, existe por lo menos un *punto maxian local* $z^{sr}_e \in B^r_e \cup \{v_s, v_t\}$ tal que $f^{sr}_{\text{sum}}(z^{sr}_e) = \max_{x \in e} f^{sr}_{\text{sum}}(x)$, con $1 \leq s \leq p$ y $1 \leq r \leq q$. Además, si $f^{sr}_{\text{sum}}(x)$ alcanza su máximo valor en dos puntos consecutivos $\tilde{z}^{sr}_e, \hat{z}^{sr}_e \in B^r_e \cup \{v_s, v_t\}$, entonces todos los puntos en $[\tilde{z}^{sr}_e, \hat{z}^{sr}_e]$ también maximizan $f^{sr}_{\text{sum}}(x)$.

Un punto $z_N^{sr} \in B_N^r \cup V$ es un *punto maxian global* si y sólo si se verifica que $f_{\text{sum}}^{sr}(z_N^{sr}) = \max_{x \in N} f_{\text{sum}}^{sr}(x) = \max_{e \in E} f_{\text{sum}}^{sr}(z_e^{sr})$, para $1 \le s \le p$ y $1 \le r \le q$. Asimismo, dos puntos consecutivos $\tilde{z}_N^{sr}, \hat{z}_N^{sr} \in B_N^r \cup V$ son los *puntos maxian globales* si y sólo si $f_{\text{sum}}^{sr}(z) = \max_{x \in N} f_{\text{sum}}^{sr}(x)$, $\forall z \in [\tilde{z}_N^{sr}, \hat{z}_N^{sr}]$.

Sea $F_{\text{sum}}(x) = (f_{\text{sum}}^{11}(x), f_{\text{sum}}^{12}(x), \ldots, f_{\text{sum}}^{pq}(x)) \in \mathbb{R}^{p \times q}$ el vector de valores de la función maxian $f_{\text{sum}}^{sr}(x)$ para todas las combinaciones de pesos $s = 1, \ldots, p$ y longitudes $r = 1, \ldots, q$. Para mayor claridad, sea $k = p \times q$, y denotamos las funciones maxian por $f_{\text{sum}}^{i}(x)$, con $i = 1, \ldots, k$. El conjunto de puntos $Z_N \subset N$ es el conjunto de puntos eficientes para el problema del maxian multicriterio si y sólo si $F_{\text{sum}}(Z_N) = \max_{x \in N} F_{\text{sum}}(x)$.

Sean $[\tilde{z}_e^1, \hat{z}_e^1]$ y $[\tilde{z}_e^2, \hat{z}_e^2]$, respectivamente, los *intervalos maxian locales* donde $f_{\text{sum}}^1(x)$ y $f_{\text{sum}}^2(x)$ alcanzan sus valores máximos, con $\tilde{z}_e^i, \hat{z}_e^i \in B_e^r \cup \{v_s, v_t\}$, $i = 1, 2$.

Lema VII.3. *El conjunto de puntos eficientes en la arista e es* $Z_e = [\min\{\tilde{z}_e, \hat{z}_e\}, \max\{\tilde{z}_e, \hat{z}_e\}]$, *donde* $\tilde{z}_e = \max\{\tilde{z}_e^1, \tilde{z}_e^2\}$ *y* $\hat{z}_e = \min\{\hat{z}_e^1, \hat{z}_e^2\}$.

Corolario VII.3. *Incluso si los puntos maxian locales se alcanzan en un solo punto, esto es* $\tilde{z}_e^1 = \hat{z}_e^1 = z_e^1$ *o* $\tilde{z}_e^2 = \hat{z}_e^2 = z_e^2$, *el conjunto de puntos eficientes en la arista e es* $Z_e = [\min\{\tilde{z}_e, \hat{z}_e\}, \max\{\tilde{z}_e, \hat{z}_e\}]$.

Corolario VII.5. *Si* $\tilde{z}_e^1 = \hat{z}_e^1 = \tilde{z}_e^2 = \hat{z}_e^2 = z_e$ *entonces el punto local eficiente en la arista e es el punto* $Z_e = z_e$.

Asumimos que $[\tilde{z}_N^1, \hat{z}_N^1]$ y $[\tilde{z}_N^2, \hat{z}_N^2]$, con $\tilde{z}_N^i, \hat{z}_N^i \in N$, $i = 1, 2$, son los intervalos maxian globales para cada función objetivo.

Proposición VII.3. *La arista e contiene sólo puntos no eficientes y por tanto puede ser eliminada si los puntos* $\tilde{z}_N^i, \hat{z}_N^i$, $1 \le i \le 2$, *satisfacen* $f_{\text{sum}}^1(z_e^1) \le f_{\text{sum}}^1(\tilde{z}_N^i)$ *y* $f_{\min}^2(z_e^2) \le f_{\min}^2(\tilde{z}_N^i)$, *o* $f_{\text{sum}}^1(z_e^1) \le f_{\text{sum}}^1(\hat{z}_N^i)$ *y* $f_{\min}^2(z_e^2) \le f_{\min}^2(\hat{z}_N^i)$, *con al menos una desigualdad estricta.*

Lema VII.3. *El conjunto de puntos eficientes en la arista e es* $Z_e = [\min\{\tilde{z}_e, \hat{z}_e\}, \max\{\tilde{z}_e, \hat{z}_e\}]$, *donde* $\tilde{z}_e = \max_{1 \le i \le k} \tilde{z}_e^i$ *y* $\hat{z}_e = \min_{1 \le i \le k} \hat{z}_e^i$.

Corolario VII.6. *En el caso de* $\tilde{z}_e^i = \hat{z}_e^i = z_e^i$ *para algunas funciones objetivo* $f_{\text{sum}}^i(x)$, *con* $1 \le i \le k$, *entonces el conjunto de puntos eficientes en la arista e es* $Z_e = [\min\{\tilde{z}_e, \hat{z}_e\}, \max\{\tilde{z}_e, \hat{z}_e\}]$.

Asumimos ahora que $[\tilde{z}_N^i, \hat{z}_N^i]$, $\tilde{z}_N^i, \hat{z}_N^i \in N$, $i = 1, \ldots, k$, son los intervalos maxian globales para cada función objetivo.

Proposición VII.4. *Si cualquier* $\tilde{z}_N^i, \hat{z}_N^i$, $1 \le i \le k$, *satisface*

$$f_{\text{sum}}^1(z_e^1) \le f_{\text{sum}}^1(\tilde{z}_N^i) \quad \wedge \quad f_{\text{sum}}^2(z_e^2) \le f_{\text{sum}}^2(\tilde{z}_N^i) \quad \wedge \quad \ldots \quad \wedge \quad f_{\text{sum}}^k(z_e^k) \le f_{\text{sum}}^k(\tilde{z}_N^i)$$

o

$$f_{\text{sum}}^1(z_e^1) \le f_{\text{sum}}^1(\hat{z}_N^i) \quad \wedge \quad f_{\text{sum}}^2(z_e^2) \le f_{\text{sum}}^2(\hat{z}_N^i) \quad \wedge \quad \ldots \quad \wedge \quad f_{\text{sum}}^k(z_e^k) \le f_{\text{sum}}^k(\hat{z}_N^i)$$

con al menos una desigualdad estricta, entonces la arista e no contiene puntos eficientes y, por lo tanto, puede ser eliminada.

VII.5 El problema del λ-anti-cent-dian multicriterio (PACDM)

Dado $\lambda \in [0,1]$ y $x \in N$, la función λ-anti-cent-dian se define como

$$f_{\text{acd}}^{sr}(\lambda,x) = \lambda f_{\min}^{sr}(x) + (1-\lambda) f_{\text{sum}}^{sr}(x)$$

siendo $f_{\min}^{sr}(x) = \min_{v_i \in V} w_i^s d^r(x,v_i)$ y $f_{\text{sum}}^{sr}(x) = \sum_{v_i \in V} w_i^s d^r(x,v_i)$, con $s=1,\ldots,p$ y $r=1,\ldots,q$. Juntando las propiedades de la función uncenter y la función maxian, podemos derivar nuevas propiedades para la función $f_{\text{acd}}^{sr}(\lambda,x)$.

Propiedad VII.1. *Dada cualquier arista $e=(v_s,v_t) \in E$ y un valor λ, $0 \le \lambda \le 1$, para cualquier punto $x \in e$ la función objetivo $f_{\text{acd}}^{sr}(\lambda,x)$, $1 \le s \le p$, $1 \le r \le q$, es una función continua, cóncava y lineal a trozos,*

a) *con un número finito de puntos de inflexión, todos perteneciendo a $B_e^r \cup Q_e^{sr}$,*
b) *con un número finito de valores máximos locales, alcanzándose todos en puntos que pertenecen al conjunto $A = \{v_s, v_t\} \cup B_e^r \cup Q_e^{sr}$,*
c) *con valor cero en los extremos de la arista para $\lambda = 1$, y*
d) $f_{\text{acd}}^{sr}(\lambda,v_s) = (1-\lambda) f_{\text{sum}}^{sr}(v_s)$ *y* $f_{\text{acd}}^{sr}(\lambda,v_t) = (1-\lambda) f_{\text{sum}}^{sr}(v_t)$.

Propiedad VII.2. *Dado un valor λ, $0 \le \lambda \le 1$, y $1 \le s \le p$, $1 \le r \le q$, existe al menos un punto, llamado el punto anti-cent-dian local $x_e^{sr} \in A = \{v_s,v_t\} \cup B_e^r \cup Q_e^{sr}$ en cada arista $e=(v_s,v_t) \in E$ tal que $f_{\text{acd}}^{sr}(\lambda,x_e^{sr}) = \max_{x \in e} f_{\text{acd}}^{sr}(\lambda,x)$. Si la función $f_{\text{acd}}^{sr}(\lambda,x)$ alcanza su máximo valor en dos puntos consecutivos $\tilde{x}_e^{sr}, \hat{x}_e^{sr} \in A$, entonces todos los puntos en $[\tilde{x}_e^{sr}, \hat{x}_e^{sr}]$ maximizan la función $f_{\text{acd}}(\lambda,x)$.*

Asimismo, un punto $x_N^{sr} \in N$ se denomina un *punto anti-cent-dian global* para un cierto valor de λ, $0 \le \lambda \le 1$, si y sólo si $f_{\text{acd}}^{sr}(\lambda,x_N^{sr}) = \max_{x \in N} f_{\text{acd}}^{sr}(\lambda,x) = \max_{e \in E} f_{\text{acd}}^{sr}(\lambda,x_e^{sr})$, con $1 \le s \le p$ y $1 \le r \le q$. Más aún, como $f_{\text{acd}}^{sr}(\lambda,x) = f_{\min}^{sr}(x)$ cuando $\lambda = 1$, el punto anti-cent-dian global (local) x_e^{sr} (x_N^{sr}) es igual al punto uncenter local (global) y_e^{sr} (y_N^{sr}). Por otro lado, para $\lambda = 0$ obtenemos $f_{\text{acd}}^{sr}(\lambda,x) = f_{\text{sum}}^{sr}(x)$, por lo que el valor x_e^{sr} (x_N^{sr}) es igual al punto maxian local (global) z_e^{sr} (z_N^{sr}). Si la función $f_{\text{acd}}^{sr}(0,x)$ alcanza su valor máximo dentro del intervalo local (global) $[\tilde{x}_e^{sr}, \hat{x}_e^{sr}]$ ($[\tilde{x}_N^{sr}, \hat{x}_N^{sr}]$), entonces este intervalo coincide con el intervalo maxian local (global) $[\tilde{z}_e^{sr}, \hat{z}_e^{sr}]$ ($[\tilde{z}_N^{sr}, \hat{z}_N^{sr}]$).

Lema VII.5. *Dada una arista $e \in E$ y un valor de λ, $0 \le \lambda \le 1$, los puntos anti-cent-dian locales están dentro del intervalo $[\min\{y_e^{sr}, \tilde{z}_e^{sr}\}, \max\{y_e^{sr}, \hat{z}_e^{sr}\}]$, con $1 \le s \le p$ y $1 \le r \le q$.*

Sea $F_{\text{acd}}(\lambda,x) = (f_{\text{acd}}^{11}(\lambda,x), f_{\text{acd}}^{12}(\lambda,x), \ldots, f_{\text{acd}}^{pq}(\lambda,x)) \in \mathbb{R}^{p \times q}$ el vector de valores de todas las combinaciones de pesos $s=1,\ldots,p$ y longitudes $r=1,\ldots,q$. Sea $k = p \times q$, y denotamos las funciones λ-anti-cent-dian por $f_{\text{acd}}^i(\lambda,x)$, con $i=1,\ldots,k$. Un conjunto $X_N \subset N$ en un conjunto eficiente para el problema λ-anti-cent-dian si y sólo si $F_{\text{acd}}(\lambda,X_N) = \max_{x \in N} F_{\text{acd}}(\lambda,x)$.

Lema VII.6. *El intervalo de puntos eficientes en la arista e es $X_e = [\min\{\tilde{x}_e, \hat{x}_e\}, \max\{\tilde{x}_e, \hat{x}_e\}]$, donde $\tilde{x}_e = \max_{1 \le i \le k} \tilde{x}_e^i$ y $\hat{x}_e = \min_{1 \le i \le k} \hat{x}_e^i$.*

Lema VII.7. *Dado λ, $0 \le \lambda \le 1$, para cualquier arista $e \in E$*

$$f_{\text{acd}}^i(\lambda,x_e^i) \le UB_e^i = \lambda f_{\min}^i(y_e^i) + (1-\lambda) f_{\text{sum}}^i(\tilde{z}_e^i), \quad i=1,\ldots,k \tag{VII.1}$$

Teorema VII.1. *Sean* $[\tilde{x}_N^i, \hat{x}_N^i]$, $1 \le i \le k$, *los intervalos anti-cent-dian globales para los k criterios. Cualquier arista* $e = (v_s, v_t) \in E$ *cumpliendo*

$$UB_e^1 = \lambda f_{\min}^1(y_e^1) + (1-\lambda) f_{\text{sum}}^1(\tilde{z}_e^1) \le f_{\text{acd}}^1(\lambda, \tilde{x}_N^i) \wedge \ldots \wedge UB_e^k = \lambda f_{\min}^k(y_e^k) + (1-\lambda) f_{\text{sum}}^k(\tilde{z}_e^k) \le f_{\text{acd}}^k(\lambda, \tilde{x}_N^i)$$

o

$$UB_e^1 = \lambda f_{\min}^1(y_e^1) + (1-\lambda) f_{\text{sum}}^1(\tilde{z}_e^1) \le f_{\text{acd}}^1(\lambda, \hat{x}_N^i) \wedge \ldots \wedge UB_e^k = \lambda f_{\min}^k(y_e^k) + (1-\lambda) f_{\text{sum}}^k(\tilde{z}_e^k) \le f_{\text{acd}}^k(\lambda, \hat{x}_N^i)$$

con al menos una de las desigualdades estricta, no contiene puntos eficientes, y por tanto, puede ser eliminada.

Lema VII.8. *Para cada arista* $e = (v_s, v_t)$, *una cota inferior de* $f_{\text{acd}}^i(\lambda, x_e^i)$, $i = 1, \ldots, k$ *es*

$$LB_e^i = \max\{\lambda f_{\min}^i(y_e^i) + (1-\lambda) f_{\text{sum}}^i(y_e^i), \lambda \max\{f_{\min}^i(\tilde{z}_e^i), f_{\min}^i(\hat{z}_e^i)\} + (1-\lambda) f_{\text{sum}}^i(\tilde{z}_e^i)\} \qquad \text{(VII.2)}$$

Para cada criterio $1 \le i \le k$, sean

$$x_{LB}^i = \arg\max_{e \in E}\{LB_e^i\} \qquad \text{(VII.3)}$$

los puntos en N donde se alcanzan las cotas inferiores globales $LB_N^i = \max_{e \in E} LB_e^i$. Obviamente, $LB_N^i \le f_{\text{acd}}^i(\lambda, \tilde{x}_N^i) = f_{\text{acd}}^i(\lambda, \hat{x}_N^i)$.

Teorema VII.2. *Cualquier arista* $e = (v_s, v_t) \in E$ *cumpliendo para algún punto* x_{LB}^i, $1 \le i \le k$

$$UB_e^1 \le f_{\text{acd}}^1(\lambda, x_{LB}^i) \quad \wedge \quad \ldots \quad \wedge \quad UB_e^k \le f_{\text{acd}}^k(\lambda, x_{LB}^i)$$

con al menos una de las desigualdades estricta, contiene sólo puntos no eficientes, y por tanto, puede ser eliminada.

VII.6 El algoritmo para resolver PACDM

El método propuesto para solucionar ACDPM tiene cinco datos de entrada, a saber, la red $N(V,G)$, la matriz de distancias *d*, el número de pesos por nodo *p*, el número de longitudes por arista *q*, y el parámetro *λ*.

En primer lugar, definimos el conjunto de puntos *P* y el conjunto de segmentos *S*. Luego, se aplica el Teorema VII.2 para eliminar las aristas que no contienen puntos eficientes. Esto se lleva a cabo en tiempo $O(k^2mn)$. Para cada arista restante *e*, y para cada peso *s* y longitud *r* computamos las funciones $f_{\min}^{sr}(x)$ y $f_{\text{sum}}^{sr}(x)$. La función $f_{\min}(x)$ corresponde a la envoltura inferior de todas las *n* funciones distancia, y se calcula en $O(n \log n)$ (Hershberger, 1989). Siendo $k = p \times q$, el tiempo para obtener todas las funciones $f_{\min}^{sr}(x)$ es $O(kn \log n)$. Por otro lado, todas las funciones $f_{\text{sum}}^{sr}(x)$ pueden calcularse en $O(kn \log n)$.

A partir de estas dos últimas funciones, se construye la función *λ*-anti-cent-dian en a lo sumo tiempo $O(kn)$. A continuación, se aplica el Lema VII.6 para obtener el intervalo de puntos eficientes locales X_e para la arista actual *e*. Dentro de este conjunto X_e, los valores de las funciones *λ*-anti-cent-dian en los puntos de inflexión son usados para generar el conjunto de puntos *P* y el conjunto de segmentos *S* en a lo sumo tiempo $O(kn)$. De este modo, la complejidad total del bucle para todas las aristas es $O(kmn \log n)$, con $|P| \in O(km)$ y $|S| \in O(kmn)$.

Finalmente, sólo queda comparar todos los puntos en el conjunto P y todos los segmentos en S para obtener el conjunto de puntos no dominados P_{ND} y el conjunto de segmentos no dominados S_{ND}. Por tanto, la complejidad total del algoritmo es $\mathrm{O}(k^3m^2n^2)$.

VII.7 Un ejemplo

Hemos usado una red planar con $n=7$ nodos, $m=15$ aristas, $p=2$ pesos por nodo y $q=2$ longitudes por arista. Así, tenemos $k=4$ criterios. Junto a cada nodo $v_i \in V$ colocamos dos pesos enteros (w_i^1, w_i^2) generados aleatoriamente en el intervalo $[1,5]$. Asimismo, cada arista $e=(v_s,v_t)\in E$ es etiquetada con dos longitudes enteras (l_e^1, l_e^2) aleatoriamente generadas en el intervalo $[1,25]$. Ponemos el parámetro λ a 0.5.

El algoritmo comienza eliminando todas las aristas que no contienen puntos eficientes. En este sentido, calculamos para cada criterio $i=1,\dots,k$, las cotas superiores UB_e^i para cada arista e así como las cotas inferiores globales LB_N^i. Luego se aplica el Teorema VII.2 sobre cada arista e. En el conjunto de aristas restantes procedemos a calcular, para cada combinación de pesos y longitudes, las funciones $f_{\min}^{sr}(x)$ y $f_{\mathrm{sum}}^{sr}(x)$. Seguidamente, dado el parámetro $\lambda=0.5$ calculamos las funciones λ-anti-cent-dian $f_{\mathrm{acd}}^{sr}(\lambda,x)$.

Posteriormente, aplicamos el Lema VII.6 para obtener los intervalos que contienen los puntos locales eficientes. Los puntos de inflexión de estas k funciones λ-anti-cent-dian dentro de los intervalos se agrupan en parejas para formar los intervalos $[x_i, x_{i+1}]$ que son añadidos al conjunto de segmentos S. Finalmente, sólo resta comparar por parejas todos los puntos en el conjunto P y todos los segmentos en el conjunto S.

VII.8 Resultados computacionales

Sin tener en cuenta el número de nodos n, los tiempos de cómputo aumentan a medida que p y q aumentan. En la mayoría de los casos el número de aristas eliminadas por el Teorema VII.2 es muy alto, alcanzando en algunos casos el 99% de eliminación. Esta cuestión se hace especialmente notable cuando $p=q=1$ (un único criterio). En este caso particular, las cotas parecen estar muy ajustadas, y así, la regla de eliminación se convierte en muy efectiva ya que más del 95% de las aristas son eliminadas, dejando aquellas que contienen los puntos óptimos finales.

Por otro lado, en redes más grandes y con $p=q=1$, el porcentaje de eliminación en todos los casos es del 99%. Sin embargo cuando $p=q=3$, el porcentaje de eliminación es mayor para $\lambda=0$ que para $\lambda=1$, y por tanto, los tiempos promedio en el último caso son mayores. En todo caso, el tiempo promedio de cómputo nunca excede de un minuto, ni incluso para las redes más grandes.

Los tiempos de cómputo se incrementan polinomialmente con n, p y q. Cuando $p=q=1$, el número de aristas procesadas es muy pequeño. En el caso de $p=q=3$, resolver el problema del uncenter multicriterio ($\lambda=1$) requiere mucho más tiempo que el problema maxian multicriterio ($\lambda=0$).

VII.9 Conclusiones y discusión

En la primera parte del capítulo se han analizado los problemas del uncenter y del maxian en redes multicriterio, a saber, redes con varios pesos en los nodos y varias longitudes en las aristas. Se han establecido nuevas propiedades junto con nuevas reglas para eliminar las aristas que contengan puntos no eficientes.

A través de un parámetro λ, se estudió la combinación convexa de estos dos últimos problemas como el problema del λ-anti-cent-dian multicriterio. Hemos propuesto una regla para eliminar aristas ineficientes, así como un algoritmo polinomial en tiempo $O(k^3m^2n^2)$ para resolver este problema. Además, para $\lambda = 0$ podemos resolver el problema maxian multicriterio, mientras que para $\lambda = 1$ podemos obtener la solución para el problema uncenter multicriterio. Más aún, cuando $p = q = 1$ este procedimiento también puede resolver los problemas unicriterio del uncenter, maxian y el anti-cent-dian. La experiencia computacional corrobora la complejidad polinomial del algoritmo así como la efectividad de la regla para eliminar las aristas ineficientes.

Conclusiones (español)

"Las cosas realmente toman sentido cuando se han terminado"
ANÓNIMO

En esta tesis se han analizado y desarrollado varios modelos de localización de servicios deseados y no deseados en redes con múltiples criterios. Asimismo, hemos propuesto también algunas mejoras en modelos de localización de servicios no deseados en redes con un solo criterio.

Por consiguiente, con respecto a la localización de servicios deseados sobre redes con n nodos y m aristas, hemos propuesto un algoritmo $O(mn \log n)$ para solucionar el problema del λ–cent-dian biobjetivo. Hemos demostrado que el conjunto de puntos eficientes para localizar el λ-cent-dian puede ser infinito, en comparación con el caso uniobjetivo, donde el λ-cent-dian está situado en el conjunto de nodos o en el conjunto de mínimos locales de la función centro.

También hemos estudiado la localización de un servicio en una red con múltiples objetivos tipo mediana. En este caso, el conjunto de puntos eficientes no se restringe a los nodos o a los caminos mínimos que enlazan los vértices mediana de cada objetivo, sino a cualquier lugar en la red. Siendo q el número de longitudes por arista, hemos propuesto un algoritmo en $O(m^2q^3)$ para solucionar este problema. Además, también hemos presentado un nuevo procedimiento en tiempo $O(q)$ que soluciona un problema de programación lineal de dos variables para determinar el conjunto de puntos eficientes.

Asimismo, hemos desarrollado un algoritmo polinomial en tiempo $O(m^2n^2k^3)$ para solucionar el problema λ-cent-dian multicriterio en redes con p pesos por nodo y q longitudes por arista, con $k = p \times q$. Este modelo generaliza el presentado en el Capítulo II usando el algoritmo multicriterio dispuesto en el Capítulo III. Además, debido a la combinación convexa mediante un parámetro λ, este modelo permite obtener la solución al problema del centro multicriterio y al problema de la mediana multicriterio.

Con respecto a los problemas de localización de servicios no deseados, primero tratamos el problema de localización del 1-centro no deseado en redes. Demostramos que las cotas superiores ya propuestas en trabajos anteriores pueden ser ajustadas. Por medio de una formulación más adecuada del problema, hemos desarrollado un nuevo algoritmo en $O(mn)$, el cual es más sencillo y computacionalmente más rápido que los ya divulgados en la literatura.

Asimismo, hemos analizado el problema de localizar una mediana no deseada en una red, obteniendo una nueva y mejor cota superior. Hemos presentado un nuevo algoritmo en tiempo

$O(mn)$ para solucionar este problema. La nueva cota superior se actualiza dinámicamente dentro del algoritmo, y de este modo, se acelera la búsqueda de los puntos óptimos. Por otra parte, siguiendo la resolución del problema maxian, también hemos propuesto un nuevo algoritmo en $O(mn)$ para solucionar el problema del λ-anti-cent-dian en redes, mejorando el método anterior en $O(mn \log n)$.

Finalmente, hemos estudiado los problemas del centro no deseado y de la mediana no deseada en redes multicriterio, estableciendo nuevas propiedades y reglas para eliminar aristas ineficientes. También hemos presentado el modelo λ–anti-cent-dian como combinación convexa de los dos últimos problemas mediante un parámetro λ. Hemos propuesto una regla eficaz para quitar aristas que contienen puntos ineficientes, así como un algoritmo polinomial en tiempo $O(m^2n^2k^3)$, siendo k el número de criterios. Además, este modelo puede solucionar el problema del centro no deseado multicriterio y el problema de la mediana no deseada multicriterio. Más aún, cuando la red tiene un solo peso por nodo y una sola longitud por arista, este algoritmo puede solucionar eficientemente los problemas unicriterio del centro no deseado, la mediana no deseada y el λ-anti-cent-dian. Además, este modelo se puede modificar ligeramente para generalizar otros modelos presentados en la literatura.

Acknowledgements

I am truly indebted to Dr. Joaquín Sicilia Rodríguez for not only being my advisor but also for his improvised lectures on the subjects of Graph Theory, Location Analysis and Combinatorial Optimization, which made easier the accomplishment of this dissertation and significantly improved my knowledge in Operations Research.

My deepest thanks to Dr. Stefan Nickel, head of the *Fraunhofer Institut für Techno– und Wirtschaftsmathematik* (ITWM) in Kaiserslautern (Germany), for accepting me as a guest researcher. Likewise, I want to thank all the staff at ITWM, specially Michael, Patricia, Teresa, Jörg and Holger, for their kind help during my stay in Kaiserslautern.

I appreciate the computing facilities provided by the *Centro de Comunicaciones y Tecnologías de la Información* at the *Universidad de La Laguna* (ULL), directed by Félix Herrera Priano, and the technical assistance of José C. González González.

Last but not least, I ought to thank all my dear colleagues at the *Departmento de Estadística, Investigación Operativa y Computación* (ULL), for their permanent support, especially José Miguel, Sergio, Antonio, Macu, Rosa and Tere.

Finally, I would like to make a posthumous acknowledgment to Professor Dr. Félix Herrera Cabello (Committee Member of my Minor Thesis), for infusing the curiosity and critical attitude in science and technology into all who had the honour of being his pupils.

Preface

Ever since the most ancient civilizations, human beings have always sought for the best place to live. Nice weather, pleasant environmental conditions, wealth of food and water, and safeness against external harms, are some of the most important issues for choosing the best spot where a new settlement should be established.

Nowadays, we face countless situations in which an *entity* or *object* is to be placed within a spatial context. Obviously, we always demand the best (optimal) site fulfilling our own requirements. This selection process implies some kind of decision-making over a set of different alternatives. In this sense, picking the right choice first involves the definition of quantifiable objectives with regards to the criteria considered. Henceforth, suitable methods can be applied to determine the optimal solutions.

Within the subject of Location Theory, network location models have usually dealt with single criterion problems, that is, concerning one weight per node and/or one length per edge. However, to properly model many real problems the decision maker requires placing more parameters on both the nodes (demand, importance, number of customers, etc) and the edges (length, time, travel cost, etc). Furthermore, many authors have deeply argued in the literature that a lot of multicriteria/multiobjective location problems have remained unresearched even though this topic has become quite relevant in the last two decades. In this thesis, we mainly focus on network location models concerning multiple criteria, in terms of considering several node weights and several edge lengths.

On the other hand, most of the papers regarding location problems address the siting of facilities that are considered *desirable* by the surrounding population such as emergency services (police/fire stations), educational centers, hospitals, etc. Nevertheless, due to the great concern on environmental issues that has arisen in the last decades, the location of *undesirable* facilities (garbage dump sites, chemical plants, nuclear reactors, etc) is playing an important role nowadays. Taking into account these concerns, we have analyzed some undesirable facility location models on single criterion networks as well as multicriteria networks.

In the remaining paragraphs we summarize the contents of this dissertation.

Chapter I allows the reader to get acquainted with the definition, notation and literature in Location Theory. In this respect, more than 150 references are reviewed, from surveys and books in general location problems, to more specialized papers on multicriteria location on networks. Besides, we examine several classification schemes for location problems in order to suitably describe the models developed in this thesis.

Chapter II analyzes the cent-dian problem on a weighted, connected and undirected network from a biobjective viewpoint, that is, considering two lengths (costs) per edge. The problem consists of locating one facility on the network which minimizes the convex combination of both the total distance and the maximum distance from any point to the rest of the network. Using computational geometry techniques, we propose a polynomial algorithm time which determines all efficient points of the network. Several computational results are supplied at the end of the chapter. A main part of this chapter, co-authored with R.M. Ramos, J. Sicilia and T. Ramos has been published in *Studies in Locational Analysis* (2000).

In Chapter III we consider the problem of locating a single facility on a network in the presence of $q \geq 2$ median-type objectives represented by q sets of edge weights (or lengths) corresponding to each of the objectives. When $q = 1$, then one gets the classical 1-median problem where only the vertices need to be considered for determining the optimal location. The chapter examines the case when $q \geq 2$ and provides a method to determine the non-dominated set of points for locating the facility. A paper regarding the multiobjective 1-median location problem and co-authored with R.M. Ramos, J. Sicilia and T. Ramos appeared in *Annals of Operations Research* (1999).

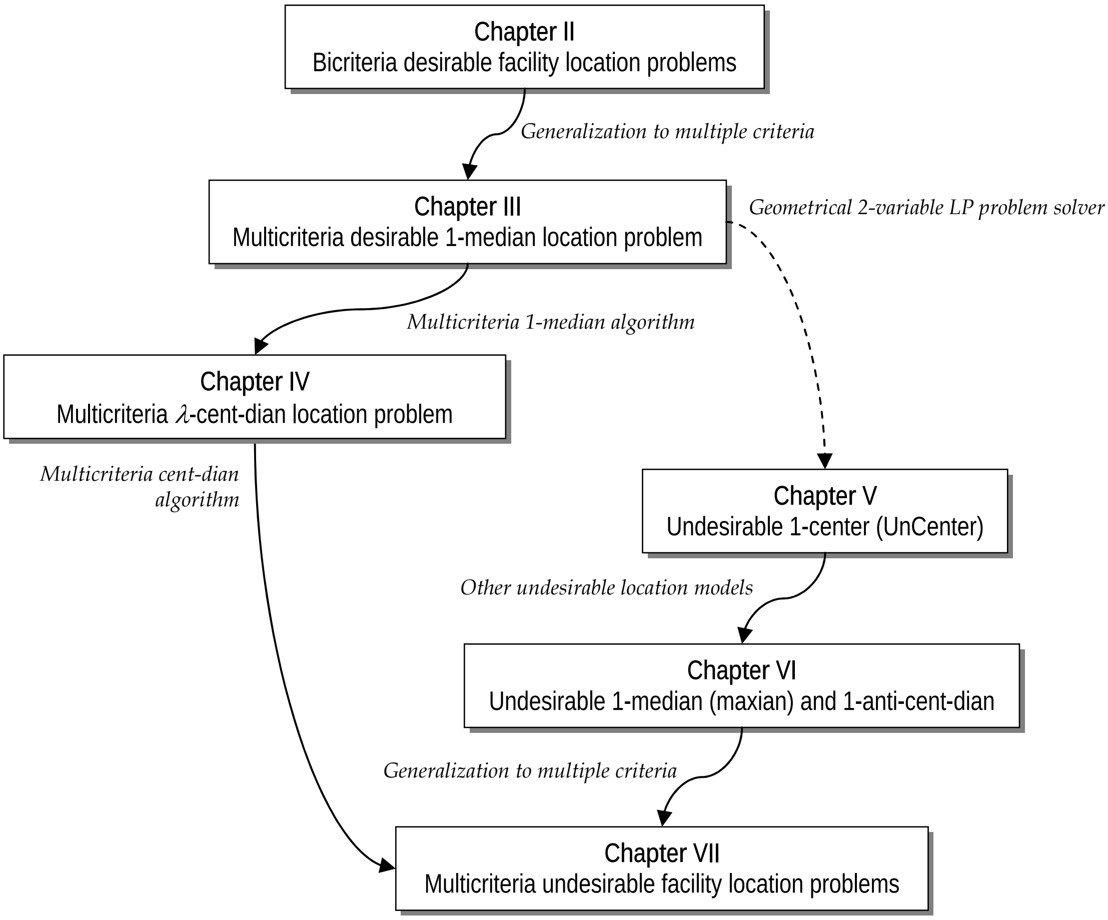
Considering networks with several weights on the nodes and several lengths on the edges, in Chapter IV we present a polynomial algorithm to solve the λ-cent-dian problem on multicriteria networks. Thus, we can easily obtain the solution to both the multicriteria center problem and the multicriteria median problem, which generalizes the model presented in the previous chapter.

Recent papers have developed efficient algorithms for the location of an undesirable (obnoxious) 1-center on general networks with n nodes and m edges. Even though the theoretical complexity of these algorithms is fine, the computational time required to get the solution can be diminished using a different model formulation and slightly improving the upper bounds. Thus, in Chapter V we present a new $O(mn)$ algorithm which is more straightforward and computationally faster than the previous ones. Computing time results comparing the former approaches with the proposed algorithm are supplied. A shorter version of this chapter, co-authored with J. Gutiérrez, S. Alonso and J. Sicilia, is published in *Journal of the Operational Research Society* (2002).

The problem of locating an undesirable facility on a network so as to maximize its total weighted distance to all nodes is addressed in Chapter VI. We propose a new upper bound to the problem. Likewise, we develop an algorithm in $O(mn)$ time which dynamically updates this new upper bound. Computational results on low and high dense networks, as well as planar networks, are shown. A paper co-authored with J. Gutiérrez and J. Sicilia regarding the new bound and the new algorithm for the maxian problem is accepted for publication in *Computers and Operations Research*. In this chapter we also analyze the anti-cent-dian problem which is a convex combination of the undesirable center problem and the undesirable median problem. We provide an efficient algorithm in $O(mn)$ time that improves a previous $O(mn \log n)$ method.

Chapter VII is devoted to the location of undesirable facilities on multicriteria networks. Firstly, we analyze the undesirable center and median models developing basic results that characterize the efficient solutions. Then, by means of a convex combination of these two latter functions, we address the λ-anti-cent-dian problem providing the algorithm that solves the problem along with an effective rule to remove inefficient edges.

The dissertation ends with some concluding remarks, as well as the bibliography reviewed. In the following figure, we illustrate the relationship among the different chapters.



Chapter I

Introduction to Location Theory

"The three most important things in real estate are:
location, location and location"
REAL ESTATE ADAGE

I.1 What is the meaning of "location"?

In a very wide sense, location problems deal with finding the right site where one or more new facilities (services) should be placed, in order to optimize (minimize or maximize) some specified criteria, which are usually related to the distance (performance measure) from the facilities to the demand points (customers).

Location problems arise fairly often in our daily modern lives. This was illustrated in a funny cartoon depicted in the preface of Mirchandani and Francis (1990), and which is also the proverb that leads this chapter. As the lady states, the three fundamental principles in real estate business are *location, location and location*. The home place offered to the couple is considered a very good location, since the travel distance to the surrounding facilities is negligible.

There are hundreds of references and internet web pages describing how to locate the best place to live. Most of the requirements made by the potential owners fulfill the following criteria: school proximity, short distance to the place of employment, quick access to public transport, near medical/emergency services and shopping centers reasonably close. The key criterion seems to be always directly related to the travel distance.

In addition to its indisputable role in real estate, location theory has also been a great concern in the establishment of new private businesses and in the development of public services. For instance, in the private sector franchisors consider the following criteria, among others, as the more outstanding in the set up of a new franchisee:

- Demographic information: density and type of surrounding population.
- Traffic count and accessibility: amount of traffic (cars and pedestrians) passing by the future franchise site.
- Competitors: who are they? Where are they located?

Salvaneschi (1996), former President of *Blockbuster Video*, Vice-President of *McDonald's* Corporation and Senior Vice-President of *Kentucky Fried Chicken* (three of the biggest franchise companies in the world), affirms that location is one of the most crucial matters in the development of a new franchise.

Likewise, when it comes to siting new businesses, dealers have tried to place them as near as possible to the potential customers. We summarize this basic idea in the following market law: *the closer the offer is to the source of demand, the more profitable the business shall become*. Further private business location problems arise as well in the location of production and assembly plants, warehouses, new offices and distribution centers.

On the other hand, the public sector also requires optimal approaches in the location of emergency services (ambulances and fire/police stations), public resources (water and electricity), or even undesirable facilities (landfills, waste treatment plants and nuclear reactors).

Daskin (1995) states in short that *"the success or failure of both private and public sector facilities depends on the locations chosen for those facilities"*. Moreover, in many circumstances locations turn out to be quite critical. For example, in the assistance to people suffering a heart attack, poorly sited ambulance stations will lead to an increased average response time, with the associated increase in mortality likelihood (Handler and Mirchandani, 1979; Daskin, 1995).

Location is also applied to the military field, involving the emplacement of resource facilities such as food centers, weapons and ammunition stores, and medical supplies. Besides, the location of either military installations or missile silos is considered an undesirable facility location problem.

The mathematical field that formulates location problems, builds up appropriate mathematical models and derives methods for solving them is called *Location Analysis*. Being a branch of the *Operational Research* framework, this subject provides decision-makers with qualitative tools for finding good solutions to realistic location decision problems. Besides, modern Location Analysis has drawn the interest of practitioners such as economists, geographers, regional planners and architect researchers, as well as researchers in diverse fields like *Industrial Engineering*, *Management Science* and *Computer Science*.

Regarding location theory taxonomy, location problems mostly fall in one of the following three types:

- *Continuous location*: locations are allowed to be anywhere in a continuous d dimensional space.
- *Discrete location*: a finite number of possible locations on the space are specified in advance. Sometimes it is also called location-allocation.
- *Network location*: special kind of location problems which are modeled on networks or trees.

Section I.4 will describe a more precise classification of location models. In this thesis we focus on network location problems. This type of problems can model real location problems on river networks, air transport networks (flight corridors), ocean transport networks (shipping lanes); highways, roads, avenues and street networks; and communication and computer networks. The literature on network location is full of inherent real applications. We briefly mention some of them:

- Locating switching centers in a communication network to minimize transmission costs or locating computer facilities or programs in a computer network to minimize annual storage and transmission costs (Handler and Mirchandani, 1979).
- A city is faced with the problem of designing a water treatment network. Untreated water emanates from a number of different sources in the city. A central water

treatment facility is to be located to minimize the total length of piping needed to conduct the untreated water to the treatment facility (Brandeau and Chiu, 1989).

- An emergency service unit is to be located in a rural area to minimize the maximal intervention time to population centers (Labbé, Peeters and Thisse, 1995).

As it has been illustrated in previous examples, decision-making on real problems involve, most of the time, more than one single criterion. Many researchers in several excellent reviews and books, for instance, ReVelle, Cohon and Shobrys (1981a,b), Ross and Soland (1980), Krarup and Pruzan (1990), Current, Min and Schilling (1990), Daskin (1995), have deeply emphasized the importance of dealing with several objectives in Location Analysis. Some other authors go even further (Erkut and Neuman, 1989; Daskin, 1995; Zhang, 1996), explicitly pointing out not only the necessity of including multiple criteria in undesirable facility location problems, but also the fact that scarce research has been done in this promising field.

The current thesis is primarily focused on bicriteria and multicriteria network location problems on both desirable and undesirable facilities. Nevertheless, we have also obtained new results on undesirable single-criterion network location problems.

Despite most of these location problems seeming to be close related to the contemporary world, they were originally proposed centuries ago. This is described in the next section where we present a brief historical background, as well as a comprehensive review of the literature on Network Location Analysis. After this, we introduce a general notation and basic concepts in Location Theory. These concepts are used to describe the classification of location problems in the last section.

I.2 Brief historical background and review of the literature

Location problems have existed almost simultaneously to the normal life of human beings. Thus, our ancestors had to decide the best location where they should inhabit to shelter from hazards, as well as considering the closeness to natural wealth sources such as rivers and fertile lands.

To best of our knowledge, the first reference on location theory dates back to the XVIIth century, when the mathematician P. Fermat proposed the following problem: *"Given three points in the plane, find a fourth such that the sum of its distances to the three given points is minimum"*.

In 1640 Torricelli observed that this problem had a geometrical solution based on three circumscribing circles. In 1834 Heinen proved that the Torricelli property was not general. Prior to this, in 1750 Simpson generalized the problem to obtaining the point that minimizes the weighted sum of distances from the three given points.

In 1857 Sylvester posed the following one sentence problem description: *"It is required to find the least circle which shall contain a given set of points in the plane"*. This is the equivalent of a location problem under the *minimax* criterion, or sometimes described as the *center* problem.

The origin of modern location theory is credited to A. Weber (1909), who incorporated the original problem by Fermat into Location Analysis in his influential treatise on the theory of industrial location *"Über den Standort der Industrien"* (Theory of the location of industries), translated later by Friedrich (1929). The problem concerned the optimal location of a factory serving a single market and with two different material source sites. The criterion considered for such location was the minimization of transport costs (travel distance). This was the beginning of the *minisum* location problems, usually called *median* problem or just *Weber* problem (Wesolowsky, 1993).

All the above references are with regard to location problems on the plane. However, some problems are modeled on networks. So, Jordan (1869) obtained a characterization of the median set of a tree. With regards to location problems on general networks, we must mention Hakimi (1964), who introduced both the median and the center on weighted networks, and thus, his seminal paper set the foundations for the development of forthcoming network location problems.

Literature on Location Analysis is extremely huge and fairly interlaced. One of the first and most extensive compilations is due to Domschke and Drexl (1985), who compiled a bibliography of over 1800 papers. In a more recent book, Drezner (1995) provided more than 1200 references. Trevor Hale (1998) keeps a web page with a list of over 3000 location science, facility location and related references. And this number keeps counting!

Next, we cite some reviews, surveys and books on location problems.

I.2.1 Surveys, reviews and books on location problems

Francis, McGinnis and White (1983) gave a selective review on location literature considering four classes of models: planar, warehousing, network and discrete models. About the same time, Hansen, Peeters and Thisse (1983) surveyed public facility location models. Hansen, Labbé and Thisse (1987a) put forward the economic interpretation of location problems.

Brandeau and Chiu (1989) presented a survey of over fifty representative problems in location research, covering standard problems as well as less traditional location problems which had emerged at that time. Eiselt (1992) reviewed facility location applications. Soon after, Chhajed, Francis and Lowe (1993) pointed out the contribution of Operations Research to the development of Location Analysis. Marsh and Schilling (1994) sought to review the literature on equity issues concerning facility location and introduced a framework and a common notation. Eiselt and Laporte (1995) examined location models with different objective functions, namely, pull, push and balance objectives. Labbé (1998) presented models, methods and applications in facility location. Hale (1999) summarized facility location in the context of taxonomy.

One of the latest overviews is due to Drezner (2002) who outlined some problems arising in Location Analysis, as well as the techniques used to get the optimal solution.

Besides, there have been several special issues in high standing research journals concerning location theory, such as Osleeb and Ratick (1986), Louveaux, Labbé and Thisse (1989), and Drezner (1992) in *Annals of Operations Research*, Current (1988) in *Environment and Planning B*, Current and Schilling in *Geographical Analysis* (1990) and *INFOR* (1991), Boffey and Karkazis (1991) in *RAIRO*, and Current and Ratick (1992) in *Papers in Reginal Science*.

Regarding excellent books on general location problems, Thisse and Zoller (1983) collected essays providing an overview of public facility location both from the perspective of economic theory and operations research. Arnott (1986) and Hansen *et al* (1987) dealt, respectively, with location theory and systems of cities in facility location. The main topic in Love, Morris and

Wesolowsky (1988) is how to locate objects in the plane or on a sphere such that a weighted sum of all distances to given objects is minimized. Hurter and Martinich (1989) connected location models with the theory of production.

A classical and state-of-the-art text on discrete location is due to Mirchandani and Francis (1990). Francis, McGinnis and White (1992) is a comprehensive introduction to quantitative methods for facility layout and location. Drezner (1995) presented a wide-ranging survey of location analysis. Puerto (1996) collected several papers concerning continuous and discrete location. Recently, Drezner and Hamacher (2002) covered theory, methodology and selected applications of Location Analysis.

The major goal of this thesis is to study, develop and, in some cases, improve several location algorithms on networks. Accordingly, in the subsequent sections we review, in chronological order, the most outstanding references on location of desirable/undesirable facilities on networks considering both one single criterion and several criteria.

In the next section, the literature on desirable facility location on networks is reviewed, regarding the *center* and the *median* criteria, as well as its combinations (*cent-dian* and *medi-center*). Then, references on undesirable facility network location are briefly commented. We end this section with the references to multicriteria location on networks.

I.2.2 Simple location of desirable facilities on networks

As we stated in the previous section, Jordan (1869) was the first to study a location problem on networks. However, Hakimi (1964) is considered the forerunner in Network Location Analysis. In this influential paper, the concepts of the center and the median vertex of a graph are generalized to the *absolute center* and the *absolute median* of a network. This led to the famous Hakimi's property: *the absolute median of a network will be always located at a node*. Thus, the absolute median coincides with the median.

Soon after, Hakimi (1965) again generalized the concept of a median in a weighted graph to a multimedian. Goldman (1969) pointed out that the Hakimi property was not general enough to apply to some particular supply/demand network problems considering warehouses. Hakimi and Maheshwari (1972) presented a generalization of the results of Hakimi and Goldman on optimum locations of centers (warehouses) in a network.

Goldman (1971) addressed the location of a central facility in a network so as to minimize the sum of its distances from the sources of flow to itself, whereas the argument in Goldman (1972) was to locate a facility on a network so as to minimize the largest of its distances from the vertices.

All these early problems were network location models. Nevertheless, some of them can also be considered on trees (acyclic networks). Goldman (1972) proposed and solved the problem of locating a facility in a tree so as to minimize the largest of its distances from the vertices. Handler (1973) provided a simple algorithm for finding the center and the absolute center of a weighted tree. Halfin (1974) obtained a modification of Goldman's algorithm.

Minieka (1977) extended previous results for calculating the centers and medians of a graph in such a way that every point on every edge as well as all vertices could be served. Hakimi, Schmeichel and Pierce (1978) presented some improvements and some generalizations of previous techniques for computing a 1-center of a network and a p-center of a tree. For a network with n vertices (nodes) and m edges, they also provided an $\mathrm{O}(mn^2 \log n)$ time algorithm for the vertex weighted network and an $\mathrm{O}(mn \log n)$ in the vertex unweighted case.

In the late seventies, Garey and Johnson (1979) established the foundations of computational complexity and $\mathcal{NP}$-completeness, which involve the recognition of $\mathcal{NP}$-hard problems in optimization. In this sense, Kariv and Hakimi (1979a,b) proved that the p-center and the p-median problems on networks were both $\mathcal{NP}$-hard, and also improved previous algorithms for finding the absolute 1-center on a weighted and unweighted network to, respectively, $\mathrm{O}(mn \log n)$ and $\mathrm{O}(mn + n^2 \log n)$ time. Minieka (1980) addressed the problem of optimally locating a facility on a network when one or more other facilities have already been located. Soon after, Minieka (1981) presented a polynomial algorithm in $\mathrm{O}(mn + n^2 \log n)$ time for finding a network absolute center. Cuninghame-Green (1984) provided an $\mathrm{O}(mn \log n)$ algorithm for finding a 1-center on unweighted networks.

In the following years, researchers tended toward new topics on network location. In this sense, Hansen, Thisse and Wendell (1986b) compared solutions concepts associated with three location problems on a network: single-facility distance minimization problem, a two facility spatial competition problem and a single-facility locational voting problem. Tamir (1987) showed that total balance and total unimodularity properties hold in matrices defined by center location problems. Chiu (1987) generalized the 1-median problem of a network with both discrete nodal and general continuous link demands, developing an exact and a heuristic procedure, as well as an efficient algorithm for tree networks. Batta and Palekar (1988) examined a modeling framework for facility location problems which allowed for a mixture of planar and network components. Hansen, Labbé and Nicolas (1991) studied the properties of the continuous center set on networks and gave an algorithm in $\mathrm{O}(m^2 \log m)$ time.

Sforza (1990) proposed new algorithms for finding the absolute center of a network with a computational effort of $\mathrm{O}(mn \log n)$ for unweighted networks and $\mathrm{O}(kmn \log n)$ in the weighted case, where k is a factor depending on the required precision and vertex weight distribution. Tamir (1992) studied the maximal direct covering tree problem and presented complexity bound improvements on another three location models: the planar 1-center rectilinear roundtrip location model, the 1-center rectilinear asymmetric distance location model, and the equity maximizing facility model. Burkard, Çela and Woeginger (1995) approached the problem of embedding a given set of communication centers into an undirected network so as to find the routing pattern which minimizes the maximum intermediate traffic over all centers.

Nickel and Puerto (1999) introduced a new type of single-facility location problem on networks which includes as special cases most of the classical criteria in the location literature. Lastly, Kalcsics, Nickel, Puerto and Tamir (2002) identified finite dominating sets for location models derived from the ordered median function, and developed polynomial time algorithms.

Most of all these previous references concern either the center or the median problem on networks. However, some authors noticed that the combination of both criteria could yield very interesting and realistic models. Thus, Halpern (1976) coined the term *cent-dian* for the point which minimizes the convex combination of the center and median objective functions, and presented a simple and efficient method to identify the cent-dian of a tree. Later on, Halpern (1978) presented a procedure to locate a facility on a network under this cent-dian criterion.

Halpern (1980) proved the duality of the constrained center and median problems. On tree networks, Halpern and Maimon (1983) studied the divergence among both the Lorentz curve and the variance of distances traveled by all customers to the new facility, and the traditional minisum (median) and minimax (center) criteria.

Independently to the work done by Halpern, Handler (1985) defined the term *medi-center* for the combination of the center and median criteria in a single formulation. He also presented efficient algorithms for locating a single facility on a tree. Hansen, Labbé and Thisse (1991) provided a complete characterization of the cent-dians in the case of a tree and an algorithm to determine this set in the case of a general network. They also introduce the concept of *generalized center* defined as the point that minimizes the difference between maximum and average distances. Berman and Yang (1991) considered two medi-center problems: the m-medi-center problem and the uncapacitated medi-center facility location problem.

Carrizosa, Conde, Fernández and Puerto (1994) provided a new axiomatic characterization for the cent-dian criterion implying an intuitive interpretation of the parameter used in the cent-dian. Ogryczak (1997a) showed that the classical approaches based on the λ-cent-dian and the generalized center solution concepts have some flaws when applied to a general network. To avoid these flaws, he proposed a new solution concept called the Chebyshev λ-cent-dian. Recently, Averbakh and Berman (1999) considered the problem of finding an optimal location of a path on a tree, using combinations of minisum and minimax criteria.

Before concluding this section we briefly mention some reviews and books on network location. The first survey is due to Tansel, Francis and Lowe (1983a,b) who listed almost one hundred references on network location models, taking special attention to those applied on trees. Moon and Chaudhry (1984) addressed network location problems with distance constraints. Hansen, Labbé, Peeters and Thisse (1987b) reviewed the main models, theorems and algorithms for the location of a single facility on a network. Hooker, Garfinkel and Chen (1991) unified and generalized previous results on the identification of the *finite dominating set* (FDS) in network location. In a quite extensive chapter, Labbé, Peeters and Thisse (1995) covered median (minisum) problems, center (minimax) problems, economic models in location and discrete location models, as well as all their extensions. Labbé and Louveaux (1997) contributed with an annotated bibliography on the uncapacitated facility location problem (UFLP), p-facility location problems, covering problems, path location problems, location-routing and hub location problems. Lately, Current, Daskin and Schilling (2002) reviewed not only basic facility location models but also location-routing models, facility location related to network design models, multiobjective models, dynamic and stochastic location models, and heuristic approaches to location models.

Excellent reference books on simple network location are the following. Handler and Mirchandani (1979) provided a cohesive treatment on center problems, median problems as well as multiobjective models, citing more than one hundred references up to that date. Daskin (1995) discussed the key classical problems in discrete and network location such as covering, center, median and fixed charge location problems, outlining for each of them the model properties, methodological tools to get the solution and several important applications. Miller, Friesz and Tobin (1996) emphasized the interdependence of the location, production and distribution decisions made by a firm operating over a network.

All the preceding literature primarily considers the facility to be located as desirable. In the following section we comment on the references regarding undesirable facility location problems on networks.

I.2.3 Undesirable facility location problems on networks

There are not many papers devoted to location of undesirable (sometimes called *obnoxious*) facilities on networks. This subject shyly emerged in the mid 1970s, and has gradually drawn the interest of researchers due to environmental issues. These types of problems are the opposite of the classical center (*minimax*) and median (*minisum*) problems, and hence, they are usually modeled using the *maximin* and the *maxisum* criteria. Other authors established alternative criteria which are not covered in this disseration. Thus, Slater (1975) defined the security center and security centroid of a graph using the criterion that a vertex u is "more central" than vertex v if there are more vertices closer to u than to v.

In the same way as Hakimi is considered the forerunner of Network Location Analysis, Church and Garfinkel (1978) are the precursors of the location of undesirable facilities on networks. They dealt with the problem of locating a point on a network so as to maximize the sum of its weighted distances (maxisum) to the nodes, and proposed an algorithm in $\mathrm{O}(mn\log n)$ time. The optimal point was called *maxian*. Minieka (1983) characterized the *anticenter* and *antimedian* location models. The former is formulated as a *maxmax* problem, whereas the latter is a directed approach to that of Church and Garfinkel (1978).

Ting (1984) treated the problem of locating a single facility in a tree network considering the maxisum criterion, providing a solution algorithm with computational effort $\mathrm{O}(n)$. Kuby (1987) pointed out that the optimal maximin objective value could be used as a lower bound on the distances between selected facilities. Moon (1989) addressed the problem of finding a point in a tree network whose distance to the closest pendant vertex (incident to a single edge) is maximal. He presented a polynomial time algorithm in $\mathrm{O}(n)$ time.

Tamir (1988) demonstrated that for some center and (obnoxious) location problems it is possible to take advantage of dynamic data structures to achieve better complexity bounds. Labbé (1990) dealt with the location of an obnoxious facility on a network using a voting procedure. She also defined the *anti-Condorcet* point as a point such that no other point is farther from a strict majority of users. Tamir (1991) discussed new complexity results for several models dealing with the location of obnoxious or undesirable facilities on graphs such as p-maximin and p-maxisum problems.

Regarding location and routing of hazardous wastes, Stowers and Palekar (1993) developed a combined model that quantifies the total exposure of the population during transportation as well as long term storage.

Kincaid and Berger (1994) studied the problem of selecting a subset of size p of the distance matrix column indices such that the smallest row sum in the resulting $n \times p$ submatrix is as large as possible. Drezner and Wesolowsky (1995) considered the problem of locating a point that should be as far as possible from arcs and nodes of a network. Berman, Drezner and Wesolowsky (1996) approached the location of a new facility on a network so that the total number (weight) of nodes within a prespecified distance is minimized.

Moreno-Pérez and Rodríguez-Martín (1999) studied the problem of locating an undesirable facility on a network maximizing a convex combination of the average and minimum distance to the population. Since this is the opposite of the cent-dian model, they called it the *anti-cent-dian*. The same problem including distance constraints was previously pointed out by Moon and Chaudhry (1984) as the *anticenter-maxian* model.

Although Tamir (1988, 2001) already commented in brief an $O(mn)$ method for the maximin problem, Melachrinoudis and Zhang (1999) solved the location of a point on a network under the maximin criterion with the same computational effort. Soon after, Berman and Drezner (2000) developed the same problem from a linear programming viewpoint in $O(mn)$ time as well. Salhi, Welch and Cuninghame-Green (2000) provided an alternative analytical approach to the Voronoi based method for the weighted 1-maximin location problem. Their enhanced method relied on two reduction tests and a suitable branch and bound scheme.

Burkard, Dollani, Lin and Rote (2001) derived algorithms with linear running time in the cases where the network is a path or a star, as well as improving previous results proposed by Tamir (1988, 1991). In a quite similar approach, Burkard and Dollani (2003) studied the pos/neg 1-center problem on networks, which asks to minimize a linear combination of the maximum weighted distance of the center to the positive and negative weighted vertices respectively. On networks, they provided an $O(mn\log n)$ algorithm, whereas on star graphs the problem can be solved in linear time. They also studied the extensions to the location of p facilities on trees.

López-de-los-Mozos and Mesa (2001) analyzed a new locational equity measure defined as the maximum absolute deviation. They investigated its properties and proposed an algorithm for locating a single facility on a network such that it minimizes this new criterion. Recently, Carrizosa and Conde (2002) addressed a p-facility location for semi-desirable facilities whose location was restricted to the edges of a planar network with rectilinear edges.

Concerning surveys and reviews on undesirable location, Moon and Chaudhry (1984) discussed and surveyed uncapacitated distance constrained network location problems such as maxian, defense, anti-center, dispersion, anticenter-maxian and dispersion-defense models. A widely cited review on this subject was due to Erkut and Neuman (1989), who brilliantly surveyed over sixty papers on maximization location models and presented a synthesis of the solution methods. In the same sense, Erkut and Verter (1995), and later Verter and Erkut (1995), overviewed and treated logistics models involving hazardous materials.

Despite not regarding network models, it is worth citing the overview on (semi-) undesirable facility location of Plastria (1996). A close related paper by Carrizosa and Plastria (1999) presented a critical overview of the mathematical models used in the field of semi-obnoxious facility location. Murray, Church, Gerrard and Tsui (1998) reviewed several approaches for addressing equity and community impact in the location of undesirable facilities. Finally, in an excellent report, Cappanera (1999) surveyed mathematical models for undesirable location problems in the plane and particularly on networks.

There are no books solely devoted to location of undesirable facilities thus far. Daskin (1995) discussed dispersion models, outlined a maxisum problem and commented on some multiobjective location problems. In Puerto (1996) there is a chapter concerning location of undesirable centers on the plane as well as on networks.

From now on we focus on multiobjective/multicriteria network location models, reviewing first the literature on the simplest multicriteria models and then the multicriteria undesirable facility models.

I.2.4 Multicriteria location of desirable facilities on networks

In spite of its wide applicability in real problems, multicriteria network location models have not been researched as much as single criterion problems. Although new research lines have developed in the last years, it seems that a lot of work still remains to be done.

Though not closely related to network location, it is worth citing an early paper by Warszawski (1973) who analyzed two multi-dimensional location problems involving location of supply sources for several commodities and a multistage distribution system in which the location of demand varies in time.

Lowe (1978) considered the problem of locating a single facility on a tree where there was more than one objective function to be minimized. Schilling (1980) approached the dynamic location of public facilities from a multicriteria viewpoint. Ross and Soland (1980) treated multicriteria issues on a model for selecting a subset of M sites at which public facilities should be established in order to serve clients located at N distinct points. Furthermore, they firmly argued that practical problems involving the location of public facilities ought to be modeled as multicriteria problems.

Tansel, Francis and Lowe (1980) studied a multiobjective multifacility location problem on a tree, where each objective concerned either the distance between a specified new facility and a specified existing facility, or the distance between a specified pair of new facilities. Nijkamp and Spronk (1981) extended the traditional location theory to a multidimensional programming framework by introducing multiple objectives. Hultz, Klingman, Ross and Soland (1981) described an interactive computer software to assist a decision maker in finding the most preferred efficient solution to a multicriteria location model. Bitran and Rivera (1982) developed an implicit enumeration algorithm to determine the set of efficient points in zero-one multiple criteria problems, which is specialized for the solution of a particular class of facility location problems. Tansel, Francis and Lowe (1982) considered a biobjective multifacility minimax location problem on a tree, which involved as objectives the maximum of the weighted distances between specified pairs of new and existing facilities, and the maximum of the weighted distances between specified pairs of new facilities.

Hansen, Thisse and Wendell (1986a) gave properties of efficient points on networks, yielding a linear algorithm for efficient points on a tree, an $O(m \log n)$ algorithm for the set of links common to all shortest paths between two points, and a polynomial algorithm for efficient points on a general network. Buhl (1988) emphasized the need for adequate objective functions of multiobjective approaches in location theory. Mirchandani (1990) presented a generalization of the p-median problem on probabilistic multidimensional networks. Puerto and Fernández (1994) dealt with the multicriteria minisum and minimax problems, introducing new solution concepts related to the equilibrium between the different aspects covered by the objectives.

Malczewski and Ogryczak (1995) formalized a discrete multicriteria location problem and developed a generalized network model. Besides, they overviewed various techniques for generating efficient solutions to multicriteria decision problems. Soon after, Malczewski and

Ogryczak (1996) focused on two approaches to locational decision-making, namely, optimizing decision rules and satisfying decision rules, discussing their advantages and disadvantages.

Krumke, Noltemeier, Ravi and Marathe (1996) studied the complexity of bicriteria compact location problems on undirected networks. Ogryczak (1997b) developed the concept of lexicographic minimax solution (lexicographic center) being a refinement of the standard minimax approach to location problems. He showed that this lexicographic minimax approach complied with both the Pareto-optimality principle and the principle of transfers, whereas the standard minimax approach may violate both such principles. Ramos, Sicilia and Ramos (1997) dealt with the problem of determining the absolute center of a network, taking into account two objective functions. These functions consist of minimizing the maximum of the distances from any point on the network to the vertices, using two independent lengths on each edge.

Hamacher, Labbé and Nickel (1999) discussed network location problems with several objectives, where every single objective is a classical median objective function. Instead of tackling location decisions with either the maximal distance (center) or the average distance (median), Ogryczak (1999) considered all the travel distances among the clients as a set of multiple uniform criteria to be minimized. This yields a multiple criteria model that takes into account the entire distribution of distances.

Despite the lack of literature on multicriteria network location problems, in the last decades many multicriteria practical applications have been developed. Some of the following may not be modeled on networks. We just cite them for its real applicability or likely use in a near future.

One of the first references is due to Cohon *et al* (1980), who built a multiple objective linear programming model for selecting the sites, types and sizes of power plants. Their objectives included the minimization of transmission costs, fuel transportation costs, water reservoir capacities and population impacts. Mladineo, Margeta, Brans and Mareschel (1987) presented a methodology for ranking the locations for the construction of small scale hydro plants with minimum costs for data gathering and the technical economic analysis.

Min (1987) minimized a cost objective and the sum of distances from each facility to the nearest competitor to model a retail service for fast-food restaurants. Again Min (1988) considered expanding and relocating public libraries in the Columbus metropolitan area. The criteria considered include coverage of population, proximity to each community, proximity to facilities being closed, and accessibility to transportation routes or parking lots.

Fortenberry, Mitra and Willis (1989) dealt with the optimal location of emergency vehicles from a multicriteria approach. Barda, Dupuis and Lencioni (1990) tackled the multicriteria location of thermal power plants. They presented a case study carried out by Electricité de France International in a North African country. Current and Storbeck (1994) introduced a multiobjective integer programming model for the franchise outlet network design problem, where the franchisor would like to maximize system-wide market coverage, while the franchisee wishes to maximize its individual market share.

Badri, Mortagy and Alsayed (1998) presented a multiple criteria approach to the fire-station location problem in Dubai, United Arab Emirates, involving conflicting objectives such as travel times and travel distances from stations to demand sites, cost-related objectives considered in previous studies and other technical, political and system required criteria.

Mahmoud, Fahmy and Labadie (2002) gave a methodology that integrated geographic information systems with multicriteria decision analysis for regionally locating and sizing desalinization facilities for domestic water supply. They applied this model to the northwestern coast of Egypt far from the freshwater sources in the Nile Valley and the Delta to optimally locate and size desalination facilities over that region.

Regarding surveys on this subject, ReVelle, Cohon and Shobrys (1981a,b) reviewed some early developments in multiobjective facility location. Current, Min and Schilling (1990) reviewed 45 papers in the area of location analysis. Lastly, it is worth citing the annotated bibliography on multiobjective combinatorial optimization provided by Ehrgott and Gandibleux (2000), which has a section devoted to location problems.

Only few books address multicriteria network location models. Handler and Mirchandani (1979) devoted a chapter to treat multiobjective location in the sense of mixed minisum-minimax models. Daskin (1995) briefly discussed multiobjective problems in location. Lately, Current, Daskin and Schilling (2002) have overviewed some issues and references regarding multiobjective location models.

To end this section, we comment in short three doctoral dissertations on multiobjective location. Oudjit (1981) studied median-type locations on deterministic and probabilistic networks with multiple dimensions. Carrizosa (1992) addressed some extensions of vector optimization problems to restricted close region location problems. Zhang (1996) developed solution procedures for two bicriteria optimization problems on networks, namely, the maximin-minisum and the maximin-maxisum models.

I.2.5 Multicriteria undesirable facility location on networks

Surprisingly, literature on multicriteria undesirable location starts in the late 80s. It seems that the concern on the location of undesirable facilities has grown only in the last years, along with the use of multiobjective/multicriteria tools to model and solve such problems.

Ratick and White (1988) proposed a multiobjective model for the location of undesirable facilities considering three objectives: minimizing the facility location costs, minimizing the opposition to the sitting plan, and maximizing equity. List and Mirchandani (1991) presented a combined routing/siting model that can be used not only for making routing decisions on waste shipments, but also for sitting decisions of waste treatment facilities. Risk, cost and risk equity were considered jointly in a multiobjective framework. A simplified form of their model was applied to the Capital District of the State of New York. Erkut and Neuman (1992) developed a multiobjective model for the location of one or more undesirable facilities to service a region which minimizes the total cost of the facilities located, the total opposition to such facilities, and power-generating stations.

By means of a multiobjective model, Rahman and Kuby (1995) examine the tradeoffs between minimizing costs (transshipment and fixed-charge problems) and public opposition (decreasing distance function from the facility) in the location of a solid waste transfer station. A case study was accomplished in the City of Phoenix, Arizona.

Giannikos (1998) presented a multiobjective model for locating disposal facilities and transporting hazardous waste along the links of a network considering four objectives, namely, minimization of total operating cost, minimization of total perceived risk, equitable distribution of risk among population centers and equitable distributions of the disutility caused by the operation of the treatment facilities.

Zhang and Melachrinoudis (2001) considered the problem of locating an obnoxious facility on a general network using two objectives, maximizing the minimum weighted distance from the point to the vertices (maximin) and maximizing the sum of weighted distances between the point and the vertices (maximsum). Skriver and Andersen (2001) modeled a semi-obnoxious facility location problem as a bicriterion problem in both the plane and the network case, applying these models to the location of a new international airport in the Jutland mainland, Denmark.

Finally, Hamacher, Labbé, Nickel and Skriver (2002) presented a polynomial time algorithm for the location of a semi-obnoxious facility on networks, and generalized the results to include maximin and minimax objectives.

Once more, the ensuing papers are commented for their real life application, though they might not be addressed on networks. Melachrinoudis, Min and Wu (1995) developed a dynamic (multiperiod) multiobjective mixed integer programming model for locating landfills. Their objectives are: minimization of total cost during the planning horizon, minimization of total risk posed on population centers, minimization of total risk posed on ecosystem and minimization of risk disequity over all individuals and time periods in the planning horizon.

Hokkanen and Salminen (1997) described an application of multicriteria decision aid to the location of a waste treatment facility in eastern Finland. The alternative locations for the new facility were considered based on 14 criteria by 28 decision makers.

To the best of our knowledge, there are no published books on multicriteria undesirable facility location problems on networks. However, Daskin (1995) devoted a complete section of a chapter to emphasize the need of more multicriteria models on undesirable facility location.

Lastly, before presenting some basic definitions and the notation, we comment in short two doctoral dissertations on multicriteria undesirable location. Saameño (1992) studied the problem of locating obnoxious facilities on a polygonal region with multiple objectives. Skriver (2001) investigated, among other models, the bicriterion semi-obnoxious location problem, the multicriteria semi-obnoxious network location problem with sum and center objectives and the bicriteria network location problem with criteria dependent lengths and minisum objectives.

I.3 Basic definitions and notation

In this section we introduce the concepts and basic definitions that are essential for the remaining chapters. We begin with the notation on classical network models, followed by the definitions related to networks with multiple criteria.

I.3.1 Standard networks

Mathematical networks can model innumerable real world problems such as aisle/road networks, river/air/ocean transport networks or communication/computer networks. All of these networks are, barring exceptions, simple (no loops or multiple edges), connected and undirected.

Thus, let $N=(V,E)$ be a network with such features, where $V=\{v_1,v_2,\ldots,v_n\}$ denotes the set of vertices or nodes, and $E=\{(v_s,v_t):v_s,v_t\in V\}$ the set of edges, with $n=|V|$ and $m=|E|$. The nodes represent demand, supply or junction points on which existing facilities or clients are already placed, whereas edges correspond to transportation lines, roadways, railways or communication channels.

Each node $v_i\in V$ is set with a positive weight w_i as follows:

$$\begin{array}{rccc} w: & V & \longrightarrow & \mathbb{R}_+ \\ & v_i\in V & \longrightarrow & w(v_i)=w_i>0 \end{array}$$

This weight w_i stands for demand rates, time/cost/loss per unit distance, number of clients, probability that a demand occurs at node v_i, or even the importance of a potential damage. Obviously, the weights are positive because a weight $w_i=0$ means null demand, time, etc, and hence it makes no sense.

On the other hand, each edge $e=(v_s,v_t)$ is labeled with a positive number l_e in terms of the following length function:

$$\begin{array}{rccc} l: & E & \longrightarrow & \mathbb{R}_+ \\ & e=(v_s,v_t)\in E & \longrightarrow & l(e)=l_e>0 \end{array}$$

Thus, a point *x* inside edge *e* ranges in the interval $[0,l_e]$. This length represents travel time/cost, reliability or any other travel attribute. The lengths are positive since any $l_e=0$ implies a null distance between v_s and v_t, and hence, it can be discarded. Figure I.1 shows a network with $n=5$ nodes and $m=7$ edges. Weights w_i are in bold, whereas lengths l_e are in italic.

Besides, each edge is assumed to be rectifiable, in the sense that there is a one-to-one correspondence between each edge and the interval $[0,1]$. Hence, given any edge $e=(v_s,v_t)\in E$ of length l_e and an inner point $x\in e$, then there is a unique number $t_e(x)\in[0,1]$ such that $t_e(x)l_e$ and $(1-t_e(x))l_e$ are the lengths along edge *e* between v_s and *x*, and *x* and v_t, respectively.

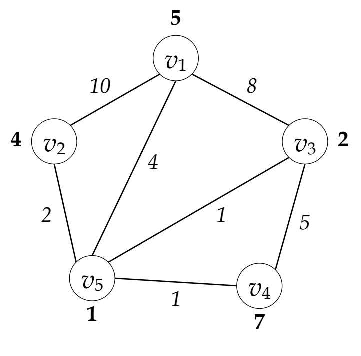


Figure I.1: Network with five nodes (weights in bold) and seven edges (lengths in italic).

A *path* is a sequence of adjacent edges, with each of the adjacent edges sharing a common node. Then, for each pair of nodes $v_a,v_b\in V$ we define the *distance* $d(v_a,v_b)$ between these two nodes as the length of any shortest path in *N* joining v_a and v_b. Moreover, given any two points $x, y \in N$, the distance $d(x, y)$ is the length of the shortest path between *x* and *y*. Given a certain edge $e = (v_s, v_t)$, sometimes it is possible that $d(v_s, v_t) < l_e$ since the edge may not provide the shortest path between the nodes v_s and v_t. This distance function $d(\cdot,\cdot)$ satisfies the following *metric properties* for any $x, y \in N$:

1. *Nonnegativity*: $d(x, y) \geq 0$, with $d(x, y) = 0$ if $x = y$.
2. *Symmetry*: $d(x, y) = d(y, x)$.
3. *Triangle inequality*: $d(x, y) \leq d(x, z) + d(z, y)$, for any $z \in N$.

At this point, the principal issue to be emphasized is that *network location models are usually based on the assumption that travel distances are lengths of shortest paths*. In this sense, given any edge $e = (v_s, v_t) \in E$, a node $v_i \in V$ and an inner point $x \in e$, we define the distance between point *x* and node v_i as:

$$d(x, v_i) = \min\{x + d(v_s, v_i), l_e - x + d(v_t, v_i)\}$$

The point on *e* where $d(x, v_i)$ attains its equilibrium, i.e. $x + d(v_s, v_i) = l_e - x + d(v_t, v_i)$, is called a *bottleneck point* b_i, with

$$b_i = \frac{d(v_t, v_i) + l_e - d(v_s, v_i)}{2}$$

A fundamental property of network distances is the following *piecewise linearity and concavity property*. This property states that the function in $x \in e = (v_s, v_t)$ defined by $d(x, v_i)$:

1. Is continuous on *e*.
2. As *x* varies from node v_s to v_t in edge *e*, either
   - increases linearly with slope w_i (see Figure I.2a), or
   - decreases linearly with slope $-w_i$ (see Figure I.2b), or
   - first increases linearly and then decreases linearly, with a breakpoint at b_i (see Figure I.2c).
3. Is *concave*, in the sense that a line segment joining any two points on the graph of the function lies on or below such graph.

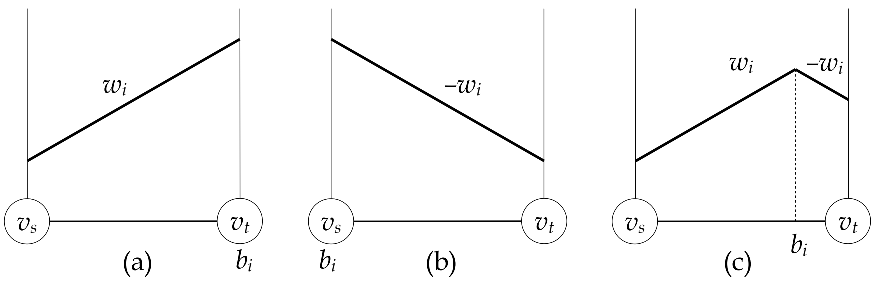


Figure I.2: The three possible plots of $d(x, v_i)$.

These are the basic concepts on standard networks. In the next section we introduce the basic notions on networks with multiple criteria, namely, considering several weights on each node as well as several lengths on each edge.

I.3.2 Networks with multiple parameters on nodes and edges

Most of the huge literature on network location problems deals with the optimization of one *single criterion*. This criterion is usually associated with the weighted distance from a certain point to the rest of nodes, for example, the minimization of the total weighted distance from a facility to the customers.

However, there are many applications in which several parameters need to be considered on each node and on each edge. Thus, several weights on each node may represent different criteria to be considered by the decision-maker(s), namely, demand rate, importance, number of potential clients, etc. On the other hand, several lengths (travel costs) on each edge might deal with distance, travel time, traffic congestion, toll rate, travel cost, etc.

In this sense, on each node $v_i \in V$, the previous weight function is now replaced by the following:

$$\begin{array}{llll} w: & V & \longrightarrow & \mathbb{R}^p \\ & v_i \in V & \longrightarrow & w(v_i) = w_i = (w_i^1, \dots, w_i^p) \end{array}$$

where p is the number of weights per nodes. For any vector of weights w_i, each w_i^r is a nonnegative number for $r = 1, \dots, p$, and we assume that not all are equal to zero.

Likewise, each edge is set with a vector of lengths (costs), as follows:

$$\begin{array}{llll} l: & E & \longrightarrow & \mathbb{R}^q \\ & e = (v_s, v_t) \in E & \longrightarrow & l(e) = l_e = (l_e^1, \dots, l_e^q) \end{array}$$

in which q is the number of lengths. Again, we assume that each component l_e^r is nonnegative for any vector l_e, and not all $l_e^r = 0$, for $r = 1, \dots, q$.

As an example of a network holding several parameters, Figure I.3 shows the same network as Figure I.1, but with two weights per node (in bold) and three lengths per edge (in italic).

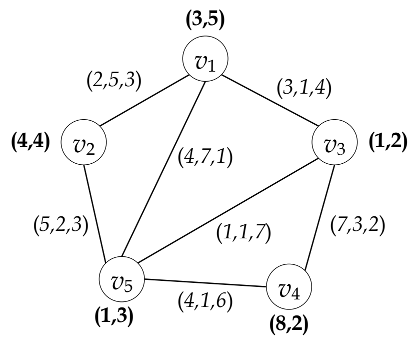


Figure I.3: Five-node and seven-edge network with several parameters.

Let r be a length index, with $1 \le r \le q$, and let $x \in e = (v_s, v_t)$ be a point inside edge e. Then, $c_e^r(x, v_s)$ is defined as the piece of line segment between x and v_s considering length r. Obviously, we have that $0 \le c_e^r(x, v_s) \le l_e^r$, with $c_e^r(x, v_t) = l_e^r - c_e^r(x, v_s)$.

For each pair of nodes $v_a, v_b \in V$ we can define the *distance* $d^r(v_a, v_b)$ between these two nodes as the length of any shortest path in N joining v_a and v_b considering length r. Likewise, given any two points $x, y \in N$, the distance $d^r(x, y)$ is the length of the shortest path between x

and y. These q distance functions also comply with the metric properties stated in the preceding section.

Given any node $v_i \in V$, we have that

$$d^r(x,v_i) = \min\{c_e^r(x,v_s) + d(v_s,v_i), c_e^r(x,v_t) + d(v_t,v_i)\}$$

denotes the distance between a point and a node considering length r, with $b_i^r = (d^r(v_t,v_i) + l_e^r - d^r(v_s,v_i))/2$ being the bottleneck point concerned with node v_i. These r network distance functions fulfill the piecewise linearity and concavity property as well.

Finally, we introduce some basic theory on multicriteria/multiobjective optimization. Usually, *multicriteria* models are those which perform a simultaneous optimization of several incommensurable objectives, for instance, minimizing the maximal travel distance and minimizing the total travel cost. On the other hand, a closely related concept is that of *vector optimization*, which determines the non-dominated solutions to a multicriteria problem.

In this sense, let $f = (f_1, f_2, \ldots, f_k)$ and $g = (g_1, g_2, \ldots, g_k)$ be two vectors belonging to $\mathbb{R}^k$. Vector f is said to *dominate* vector g, and it is denoted by $f \prec g$, if and only if:

$$f_i \le g_i, \forall i = 1,\ldots,k \quad \text{and} \quad \exists j \in \{1,\ldots,k\} : f_j < g_j$$

Then, given the subset of vectors $U \subseteq \mathbb{R}^k$, a vector $f \in U$ is called *non-dominated*, *efficient* or *Pareto optimal* (Pareto, 1896) with respect to subset U if there is no other vector $g \in U$ such that $g \prec f$. The set of all non-dominated vectors with respect to U is denoted by U_{ND}. For a further knowledge in multicriteria optimization, the reader is referred to Steuer (1986).

Having described the basic concepts and the notation used to model the location problems developed in this dissertation, we next present a general classification of network location models.

I.4 Problem classification

As we remarked in section I.2, there might be currently more than 3000 references on location. This huge literature ought to have been classified somehow. Accordingly, several authors proposed some schemes of classification in order to concisely state and unambiguously describe location models.

The first attempt was made by Handler and Mirchandani (1979), who suggested a classification of location problems on networks regarding the objective function, the number of facilities, the type of network, the points of demand, and the feasible facility sites. Brandeau and Chiu (1989) presented a taxonomy based on the following categories: the objective function, the decision variables and the system parameters. Within each category, a menu of choices was designed to specify the most common location problems. Francis, McGinnis and White (1992) considered six major elements in classifying facility location problems: new facility characteristics, existing facility locations, new and existing facility interactions, solution space characteristics, distance measure, and the objective.

Eiselt and Laporte (1995) discussed the most important components of location models such as space, number of facilities to be located, number of existing facilities, objective and customers. Daskin (1995) developed a similar taxonomy on location problems to the ones given by Brandeau and Chiu (1989) and Krarup and Pruzan (1990). Lately, Hale (1999) provided an integrated taxonomy of facility location problems based on twelve parameters, namely, objective function, distance metric, feasible subspace, number of facilities, demand portrayal, market competition, facility setup costs, facility capacity, region symmetry, facility utilization, facility type, and planning horizon.

All these latter classifications were meant for describing a wide range of models. However, some researchers have suggested other schemes for some particular location models. Thus, Moon and Chaudhry (1984) provided a 3-position scheme for distance-constrained location problems on networks. Eiselt, Laporte and Thisse (1993) used a 5-position scheme to classify competitive location models. Carrizosa, Conde, Muñoz and Puerto (1995) presented a 6-position scheme for classifying planar models. Regarding undesirable facility location, Erkut and Neuman (1989) classified such problems with respect to nine criteria, namely, number of facilities to be located, solution space, feasible region, distance measure, distance constraints, weights, distance terms included, interactions considered, and objective.

Recently, Hamacher and Nickel (1998) proposed a 5-position scheme that can be used not only for classes of specific location models, but for covering all location models. It has been in use since 1992, and has proven to be very useful in research issues, software development and in university subjects and lectures. Consequently, we decided to follow this taxonomy to define and describe the location models developed in this thesis.

The classification scheme has five positions written as

Pos1 / Pos2 / Pos3 / Pos4 / Pos5

Table I.1 shows a brief meaning of these five positions. No special assumptions are indicated by a •. For a more detailed explanation, the reader is referred to Hamacher and Nickel (1998).

Position	*Meaning*	*Usage (example)*	
1	Number of facilities	$1, 2, \ldots, n$	
2	Type of problem	$\mathcal{P}$	Planar location problem
		$\mathcal{D}$	Discrete location problem
		$\mathcal{G}$	Network location problem
		$\mathcal{T}$	Tree network location problem
3	Special assumptions and restrictions	$w_m = 1$	All weights are equal
		$\mathcal{R}$	Forbidden region
4	Type of distance function	l_1	Manhattan metric
		$d(\mathcal{V},\mathcal{V})$	Node to node distance
		$d(\mathcal{V},\mathcal{G})$	Node to point distance
5	Type of objective	$\sum$	Median problem
		max	Center problem
		CD	Cent-dian
		$\max_{\text{obnox}}$	Anti-center problem
		$Q-\sum_{\text{par}}$	Q criteria median, Pareto locations

Table I.1: Summarized classification scheme for location problems.

To end this chapter, Table I.2 provides the references of several classical network location problems and their associated classification. Some of these algorithms form the base for the subsequent new or improved algorithms presented in this dissertation.

Problem	*Author(s)*	*Classification*
Absolute median	Hakimi (1964)	$1/\mathcal{G}/\bullet/d(\mathcal{V},\mathcal{V})/\sum$
Absolute center (weighted)	Kariv & Hakimi (1979a)	$1/\mathcal{G}/\bullet/d(\mathcal{V},\mathcal{G})/\max$
Absolute center (unweighted)	Kariv & Hakimi (1979b) Minieka (1983)	$1/\mathcal{G}/w_m=1/d(\mathcal{V},\mathcal{G})/\max$
λ-cent-dian	Halpern (1978)	$1/\mathcal{G}/\bullet/d(\mathcal{V},\mathcal{G})/\mathrm{CD}$
Undesirable median (maxian)	Church & Garfinkel (1978) Tamir (1991)	$1/\mathcal{G}/\bullet/d(\mathcal{V},\mathcal{G})/\sum_{\text{obnox}}$
Undesirable center (maximin)	Tamir (1988) Melachrinoudis & Zhang (1999) Berman & Drezner (2000)	$1/\mathcal{G}/\bullet/d(\mathcal{V},\mathcal{G})/\max_{\text{obnox}}$
Multiobjective tree network locations	Lowe (1978)	$1/\mathcal{T}/\bullet/d(\mathcal{V},\mathcal{G})/Q-(f:\text{convex})_{\text{par}}$

Table I.2: Network location problems with their associated classification scheme.

Chapter II

Bicriteria location of a desirable facility on networks

"In many real world problems, the objective function is a mixture of the two different, possibly adverse objectives, center and median"
J. HALPERN

II.1 Introduction

The center and median location problems, which consider one cost per edge, were introduced by Hakimi (1964). Since then, several efficient algorithms have been proposed for their solution, and many applications of these concepts have also been found, for example, Handler (1974), Kariv and Hakimi (1979a,b), and Minieka (1981). Briefly, location problems consist of finding the optimal points on the network where services should be situated in order to minimize a specific function, which is related to the location of the demand points.

The aim of the center problem is to locate a point on the network so that the distance to the farthest node is a minimum. In the real world, this type of function is frequently associated with the location of emergency services such as ambulance, fire and police stations.

On the other hand, the median problem is concerned with the location of a point on the network so that the total distance (the sum of all the distances) from this point to all the nodes is minimized. Real problems related to the median arise in the location of service points which are dedicated to the distribution of persons and goods (products delivery, school transport, mail service, etc).

However, sometimes these two concepts are combined. For example, the location of a supermarket should consider both the center function, so that it is not very far for the customers, and the median function, so that food delivery is much faster.

The convex combination of these two functions (center and median) is called the *cent-dian function*, and the point which minimizes such a function is called the *cent-dian* of a network. This convex combination was initially proposed in the mid 70's by Halpern (1976), who coined the term *cent-dian*, and independently by Handler (1976, 1985) who proposed the term *medi-center*.

However, in many situations, determining the cent-dian of a network must be carried out considering several criteria. So, using the example of the supermarket introduced above, we could define two parameters per edge: its length, and the transport cost involved (vehicle maintenance, petrol, toll, etc).

Following the works done in Ramos, Sicilia and Ramos (1992, 1997), we approach in this chapter the study of the cent-dian problem with two associated objective functions, and we present a method to find out the set of all possible cent-dian efficient points. The algorithms which we propose for obtaining these points use computational geometry techniques.

Computational Geometry is a branch of Computer Science that studies methods and algorithms for solving geometric problems. The geometric problems that we need to solve in order to obtain the efficient points arise on a bidimensional space, that is, on the plane. These problems consist of determining intersections among pair of segments, sorting segments and calculating the lower envelope of line segments in order to identify the non-dominated ones. For more details, the reader is referred to Bentley and Ottman (1979), Preparata and Shamos (1985), Manber (1989), and Hershberger (1989).

The remainder of the chapter is structured as follows. In section II.2 we introduce the notation used throughout the chapter. Section II.3 is devoted to a brief discussion about possible properties of the biobjective cent-dian problem. Before starting the search procedure, section II.4 presents a rule for removing edges which will not contain efficient points. Section II.5 gives algorithms for the center and median functions. Section II.6 presents the algorithm to calculate the biobjective cent-dian. Finally, we present some computational results together with the conclusions.

II.2 Notation and formulation of the model

Let $N=(V,E)$ be a finite, simple, undirected and connected network, with $V=\{v_1,v_2,\ldots,v_n\}$ as the set of nodes (vertices) and $E=\{(v_s,v_t):v_s,v_t\in V\}$ as the set of edges, $m=|E|$. A positive weight w_i is associated with each node $v_i\in V$, and on each edge $e=(v_s,v_t)\in E$ we place two independent parameters or costs (lengths) l_e^r, with $r=1,2$, which may represent the length of edge *e*, the travel time between v_s and v_t, the cost of shipping one unit of a certain commodity along edge *e*, etc.

Recall from section I.3.2 that for any pair of points $x,y\in N$, the distance $d^r(x,y)$, with $r=1,2$, was defined as the length of the shortest path between *x* and *y* when the *r*-th cost was considered. Likewise, for $r=1,2$, the distance between a point *x* inside edge *e* and any node $v_i\in V$ is defined as $d^r(x,v_i)=\min\{c_e^r(x,v_s)+d(v_s,v_i),c_e^r(x,v_t)+d(v_t,v_i)\}$, where $c_e^r(x,v_s)$ and $c_e^r(x,v_t)$ are the lengths of the pieces of edge between point *x* and each of its end nodes v_s and v_t, respectively.

As mentioned earlier, the center problem consists of minimizing the maximum distance from any point (center) of the network to the set of nodes. Formally, for $r=1,2$, the *center function* can be formulated as

$$f_{\max}^r(x)=\max_{v_i\in V} w_i d^r(x,v_i),\quad \forall x\in N$$

and a point $x_c\in N$ is an (absolute) *center* for the *r*-th cost if $f_{\max}^r(x_c)=\min_{x\in N} f_{\max}^r(x)$.

On the other hand, the *median function* is defined as the total minimum distance from one point (median) of the network to the set of nodes. The formulation of this function is:

$$f_{\text{sum}}^r(x)=\sum_{v_i\in V} w_i d^r(x,v_i),\quad \forall x\in N$$

and a point $x_{\mathrm{m}} \in N$ is a *median* for the *r*-th cost when $f^r_{\mathrm{sum}}(x_{\mathrm{m}}) = \min_{x \in N} f^r_{\mathrm{sum}}(x)$.

The cent-dian function is made up from the convex combination of the two previous functions:

$$f^r_{\mathrm{cd}}(\lambda, x) = \lambda f^r_{\max}(x) + (1-\lambda) f^r_{\mathrm{sum}}(x) = \lambda \max_{v_i \in V}\{w_i d^r(x, v_i)\} + (1-\lambda)\sum_{v_i \in V} w_i d^r(x, v_i)$$
$$\forall x \in N, \quad 0 \le \lambda \le 1, \quad r = 1,2$$

Given a cost index *r*, the λ-cent-dian is the point on the network which minimizes the convex combination of the two goals. The value of λ reflects the importance attributed to the weighted maximum distance compared to the weighted total distance.

However, one may still observe a large discrepancy in the values of the functions $f^r_{\max}$ and f^r_{sum}, due to the fact that the values of the second function are always larger than the first one. This seems to contradict any intuitive idea of distributional equity between criteria, thus justifying another convex combination to build the cent-dian function. Hence, several authors use the *unweighted center function* and the *weighted median function* divided by the sum of weights. See for instance Halpern (1978), Hansen, Labbé and Thisse (1991), Labbé, Peeters and Thisse (1995).

In accordance with these authors, we remove the weights from the center function as follows

$$f^r_{\max}(x) = \max_{v_i \in V} d^r(x, v_i), \quad \forall x \in N$$

Then, we have the next objective function:

$$F^r_{\mathrm{cd}}(\lambda, x) = \lambda \max_{v_i \in V} d^r(x, v_i) + \frac{(1-\lambda)}{W}\sum_{v_i \in V} w_i d^r(x, v_i) = \lambda f^r_{\max}(x) + (1-\lambda) f^r_{\mathrm{sum}}(x)/W$$
$$\forall x \in N, \quad 0 \le \lambda \le 1, \quad r = 1,2$$

where $W = \sum_{v_i \in V} w_i$ represents the sum of weights. Thus, the problem to be solved can be now formulated as follows: to find a point *x* on *N* such that

$$\min_{x \in N}(F^1_{\mathrm{cd}}(\lambda, x), F^2_{\mathrm{cd}}(\lambda, x)), \quad 0 \le \lambda \le 1$$

According to the classification scheme given in section I.4, this problem is denoted by $1/\mathcal{G}/\bullet/d(\mathcal{V},\mathcal{G})/2-\mathrm{CD}_{\mathrm{par}}$.

To solve this problem, an order on $\mathbb{R}^2$ has to be defined. We consider the Pareto order, that is, given two vectors $f, g \in \mathbb{R}^2$, the component-wise order is defined by

$$f = (f_1, f_2) \le (g_1, g_2) = g \Leftrightarrow f_i \le g_i,\ i = 1,2$$

If at least one of the latter inequalities is strict, the expression $f \prec g$ is used then, and *f* is said to *dominate g* (see section I.3.2).

Let $U = \{(F^1_{\mathrm{cd}}(\lambda, x), F^2_{\mathrm{cd}}(\lambda, x)), x \in N, \lambda \in [0,1]\}$ be the set of all possible vectors associated with all the *x* points on network *N*. Recall from section I.3.2 that a vector $f \in U$ is called *non-dominated* or *efficient* if there is no vector $g \in U$ such that $g \prec f$. The set of all non-dominated vectors is denoted by U_{ND}.

The set of all locations *x* on *N* such that $(F^1_{\mathrm{cd}}(\lambda, x), F^2_{\mathrm{cd}}(\lambda, x)) \in U_{ND}$ is denoted by *L*, and a point $x \in L$ is called a *non-dominated* or *efficient location point*.

The rest of the chapter is devoted to find these efficient location points in the biobjective cent-dian problem. But before showing the method developed, we prove in the next section that some of the properties stated for the uniobjective cent-dian do not hold in the biobjective case.

II.3 Properties of the cent-dian

Given a cost (length) index *r* and an edge $e=(v_s,v_t)\in E$, let P_e^r be the set of points containing the nodes $v_s,v_t \in V$ and the local minima of $f_{\max}^r(x)$, for any point *x* on *N*. The following properties of the cent-dian were stated and proved in Halpern (1978):

Property II.1. *Given r,λ and one inner point x on edge e, the function*

$$F_{\text{cd}}^r(\lambda,x)=\lambda f_{\max}^r(x)+(1-\lambda)f_{\text{sum}}^r(x)/W$$

is continuous, piecewise linear and with a finite number of local minimal values of $F_{\text{cd}}^r(\lambda,x)$ *on the edge e, all attained at points which are members of* P_e^r.

Property II.2. *Given r, the function* $F_{\text{cd}}^r(\lambda,x)=\min\{F_{\text{cd}}^r(\lambda,x):x\in N\}$ *is a continuous, piecewise linear and concave function for λ,* $0\le\lambda\le 1$.

Property II.3. *Given the r-th cost, if* $x_{\text{cd}}(\lambda)\in N$ *is a cent-dian point for a given λ, then the function* $f_{\max}^r(x_{\text{cd}}(\lambda))$ *is a non-increasing function of λ and the function* $f_{\text{sum}}^r(x_{\text{cd}}(\lambda))/W$ *is a non-decreasing function of λ.*

Property II.4. *Given r and λ, the cent-dian of a network is on the shortest path connecting the center and the median of the network.*

The first property states that the set of *λ*-cent-dian locations is in the set $P_N^r=\{P_e^r : e\in E\}$, that is, we only need evaluate the objective function $F_{\text{cd}}^r(\lambda,x)$ at the nodes of the network and at the local minima of $f_{\max}^r(x)$ along all the edges. The set P_N^r is always finite. This result has been used by several authors to obtain $\text{O}(mn\log n)$ algorithms which determine the *λ*-cent-dian on a network for a given value of *λ*.

However, the set of efficient locations for the biobjective cent-dian problem can be infinite and non-numerable. This possibility is shown using the network of Figure II.1. There are four nodes and five edges. A weight is associated with each node, and on each edge there are two independent parameters or costs. For $\lambda=0.4$, the set of efficient points where the *λ*-cent-dian can be located is shown in Table II.1. That set is infinite and non-numerable.

Now, taking into account the last property of the *λ*-cent-dian for the uniobjective case, we may wonder if it is possible to find a similar result for the biobjective case: Is the biobjective cent-dian of a network on any shortest path connecting any uniobjective center with any uniobjective median of the network?

Unfortunately, the answer is negative as it may be seen in the example on Figure II.1. In Table II.2, the absolute centers $(x_{\text{c}}^1,x_{\text{c}}^2)$ and medians $(x_{\text{m}}^1,x_{\text{m}}^2)$ are shown for the two objectives.

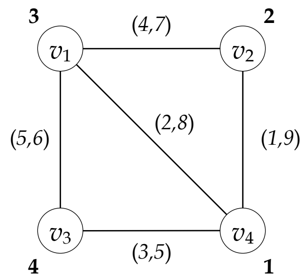


Figure II.1: Weighted network with four nodes, five edges and two independent costs on each edge.

Edge	*Set of efficient points*
(v_1, v_3)	$[0.436681, 1]$
(v_1, v_4)	$[0, 0.191048]$, $[0.4375, 0.494318]$
(v_2, v_4)	$[0.944444, 1]$
(v_3, v_4)	$[2.5, 3]$

Table II.1: Set of efficient points for the λ-cent-dian, with $\lambda = 0.4$. The intervals are determined according to the first cost.

	Objective 1	*Objective 2*
Center	$x_c^1 = 2.5$ on edge (v_3, v_4)	$x_c^2 = 0.125$ on edge (v_1, v_4)
Median	$x_m^1 = v_4$	$x_m^2 = v_1$
Cent-dian	$x_{cd}^1 = 2.5$ on edge (v_3, v_4)	$x_{cd}^1 = v_1$

Table II.2: Centers, medians and λ-cent-dians for the two objectives. The points are presented with respect to the first cost.

It is easy to see that the possible shortest paths connecting any center and any median of the network pass only through edges (v_1, v_4) and (v_3, v_4). So, the set of efficient points should be on those edges. However, it may be seen in Table II.1 that there are also efficient points (infinite in number) on edges (v_1, v_3) and (v_2, v_4).

Another question could be whether the set of efficient points is on any shortest path connecting the cent-dians for the two objectives. The answer is also negative and the same network of Figure II.1 helps to prove this. The λ-cent-dians x_{cd}^1 and x_{cd}^2 are in Table II.2.

The shortest path connecting the λ-cent-dians passes through edges (v_1, v_4) and (v_3, v_4). Unfortunately, there are still efficient points outside (see Table II.1).

Therefore, it is not possible to generalize some of the properties obtained for the uniobjective λ-cent-dian. To characterize the set of efficient locations we should analyze all the edges of the network. The next section shows a simple rule to remove non-efficient edges, that is, edges containing non-efficient points only.

II.4 Removal of edges

The algorithms we present in the following sections perform the calculations over the edges of the network to obtain the efficient points. For this reason, it is very important that the number of edges to examine should not be very large. Here we show a simple rule to remove the edges which will not contain efficient points.

Given a value of λ, $0 \le \lambda \le 1$, and for all edges $e = (v_s, v_t) \in E$, check whether one of these conditions is verified for some nodes $v_k, v_l \in V$:

a)
$$\lambda \min\{d^1(v_s, v_k), d^1(v_t, v_k)\} + (1-\lambda)\sum_{v_i \in V} w_i \min\{d^1(v_s, v_k), d^1(v_t, v_k)\} / W \ge F_{\text{cd}}^1(\lambda, v_{\text{cd}}^1)$$
and
$$\lambda \min\{d^2(v_s, v_l), d^2(v_t, v_l)\} + (1-\lambda)\sum_{v_i \in V} w_i \min\{d^2(v_s, v_l), d^2(v_t, v_l)\} / W \ge F_{\text{cd}}^2(\lambda, v_{\text{cd}}^1)$$

or

b)
$$\lambda \min\{d^1(v_s, v_k), d^1(v_t, v_k)\} + (1-\lambda)\sum_{v_i \in V} w_i \min\{d^1(v_s, v_k), d^1(v_t, v_k)\} / W \ge F_{\text{cd}}^1(\lambda, v_{\text{cd}}^2)$$
and
$$\lambda \min\{d^2(v_s, v_l), d^2(v_t, v_l)\} + (1-\lambda)\sum_{v_i \in V} w_i \min\{d^2(v_s, v_l), d^2(v_t, v_l)\} / W \ge F_{\text{cd}}^2(\lambda, v_{\text{cd}}^2)$$

In the above formulae v_{cd}^1 and v_{cd}^2 are the cent-dian vertices of the network for each single objective. These cent-dian vertices can be calculated using the algorithm given in Halpern (1978). The values $F_{\text{cd}}^1(\lambda, \cdot)$ and $F_{\text{cd}}^2(\lambda, \cdot)$ are the values of the cent-dian function over these nodes. If a) or b) are verified, then edge *e* is removed. Otherwise, the edge is examined.

II.5 Calculating the center and median functions

As was stated in the above sections, the cent-dian function $F_{\text{cd}}^r(\lambda, x)$ for the *r*-th objective is made up as a convex combination of the center and median functions. Thus, to build up each $F_{\text{cd}}^r(\lambda, x)$ we must first develop the algorithms that will allow us to calculate these two functions.

The center function can be calculated using the next Algorithm II.1. The time complexity of this algorithm is $\mathrm{O}(mn + n^2 \log n)$, provided that the distance matrix is given.

On the other hand, to calculate the median function the following Algorithm II.2 may be used. Its time complexity is $\mathrm{O}(mn \log n)$, assuming again that the distance matrix is known.

II.6 Determining the biobjective cent-dian

We now propose an exact algorithm in $\mathrm{O}(mn \log n)$ which determines the biobjective cent-dian points (see Algorithm II.3).

In order to obtain the non-dominated vectors corresponding to the efficient points, we can use Hershberger's algorithm (1989) to calculate the lower envelope of line segments on the objective space in $\mathrm{O}(S \log S)$ time, where *S* is the number of segments.

```
function Center(Network N(V,E), DistanceMatrix d)
{   // Do for all edges and r costs…
    for all edges e := (v_s, v_t) ∈ E do
        for r := 1 to 2 do
            {   Calculate the local minima using Kariv & Hakimi (1979a) or Minieka (1981)
                p := number of local minima points on edge e
                // Calculate the local maxima using the following procedure.
                for i := 1 to p−1 do
                    {   Let x_i and x_{i+1} be two consecutive local minima inside edge e,
                        and let f^r_max(x_i) and f^r_max(x_{i+1}) be their radii.
                        if f^r_max(x_i) = f^r_max(x_{i+1}) then
                            {   x := (x_i + x_{i+1}) / 2
                                f^r_max(x) := f^r_max(x_i) + x − x_i
                            }
                        else if x_{i+1} − x_i > | f^r_max(x_{i+1}) − f^r_max(x_i) | then
                            {   if f^r_max(x_i) > f^r_max(x_{i+1}) then   { α := x_i , β := 1 }
                                else                                    { α := x_{i+1} , β := −1 }
                                x := α + β(x_{i+1} − x_i − | f^r_max(x_{i+1}) − f^r_max(x_i) |) / 2
                                f^r_max(x) := f^r_max(α) + | x − α |
                            }
                    }
                Drawing the lines which connect the radii of the local minima points and
                    the radii of the local maxima points we get the center function for edge e
            }
    return f^r_max(x), ∀x ∈ N, r = 1,2
}
```

Algorithm II.1: The center function.

```
function Median(Network N(V,E), DistanceMatrix d)
{   // Do for all edges and r costs…
    for all edges e := (v_s, v_t) ∈ E do
        for r := 1 to 2 do
            {   Let f^r_sum(v_s) := Σ_{v_i∈V} w_i d^r(v_s, v_i) and f^r_sum(v_t) := Σ_{v_i∈V} w_i d^r(v_t, v_i)
                for all nodes v_i ∈ V do   // Compute all the n bottleneck points (see section I.3.1).
                    b^r_i := (d^r(v_t, v_i) + l^r_e − d^r(v_s, v_i)) / 2
                Sort points b^r_i in increasing order. Let b^r_1, b^r_2, …, b^r_p be their p possible values.
                for i := 1 to p do
                    f^r_sum(b^r_i) := Σ_{v_k∈A} w_j (b^r_i + d^r(v_s, v_k)) + Σ_{v_k∈B} w_j (l^r_e − b^r_i + d^r(v_t, v_k))
                        where   A = {v_k ∈ V : b^r_i ≤ b^r_k},   B = {v_k ∈ V : b^r_i > b^r_k}
                Draw lines linking points (b^r_i, f^r_sum(b^r_i)), with (v_s, f^r_sum(v_s)) and (v_t, f^r_sum(v_t))
            }
    return f^r_sum(x), ∀x ∈ N, r = 1,2
}
```

Algorithm II.2: The median function.

function *BiobjectiveCentDian*(Network $N(V,E)$, DistanceMatrix *d*, Parameter λ)
{ Apply the reduction rule presented in section II.4 to remove the non-efficient edges
  for all remaining edges $e := (v_s, v_t) \in E$ **do**
    { **for** r := 1 to 2 **do**
      { Calculate the center and median functions according to Algorithm II.1 and Algorithm II.2 respectively
        Build up the cent-dian function $F^r_{\text{cd}}(\lambda, x), \forall x \in e$
      }
      Keep the polygonal line that joins the pair of values $(F^1_{\text{cd}}, F^2_{\text{cd}})$ related to the points of function $F^r_{\text{cd}}(\lambda, x)$ where the slope changes
    }
  Using the graphical representation of these polygonal lines, calculate the set U_{ND} of all non-dominated vectors which will correspond to the set *L* of efficient points
  return *L*
}

Algorithm II.3: The biobjective cent-dian function.

Given an edge, there are $\text{O}(n)$ line segments linking pairs of values $(F^1_{\text{cd}}, F^2_{\text{cd}})$, so there will be at most $S = mn$ line segments. Since the complexity of the final step is greater or equal than the complexity of the previous steps, the total time complexity of the algorithm is $\text{O}(mn \log n)$.

Before presenting the computational results in the next section, we will use the network shown in Figure II.1 to illustrate an example of how Algorithm II.3 works. First, we apply the reduction rule described in section II.4. Unfortunately, no edge is removed from the network by this procedure, so all five edges remain.

In the following step, we calculate for every *r*-th objective the center and median functions, using Algorithm II.1 and Algorithm II.2. For example, for edge (v_3, v_4), Figure II.2 shows the center function and Figure II.3 the median function, both considering the two objectives.

Next, given the parameter λ, we must build up the cent-dian function using the center and median functions of the previous step. In Figure II.4, the cent-dian function for each of the two objectives is depicted.

Then, we have to draw the polygonal lines which join the pair of values $(F^1_{\text{cd}}, F^2_{\text{cd}})$ where the cent-dian function $F^r_{\text{cd}}(\lambda, x)$ changes the value of its slope. Figure II.5 shows the polygonal lines obtained from the cent-dian functions of edge (v_3, v_4).

Finally, we have to find the non-dominated vectors using all the polygonal lines obtained for each edge on the objective space. For that, we can use Hershberger's algorithm (1989), which determines the lower envelope of these polygonal lines.

These non-dominated values will correspond to the efficient location points on the network. Table II.1 shows the set of efficient λ-cent-dian location points for $\lambda = 0.4$.

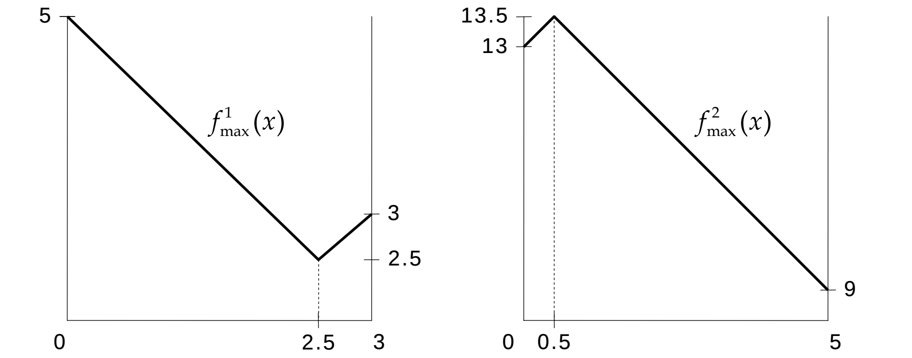


Figure II.2: Center function of edge (v_3, v_4) for the first (left) and second (right) objectives.

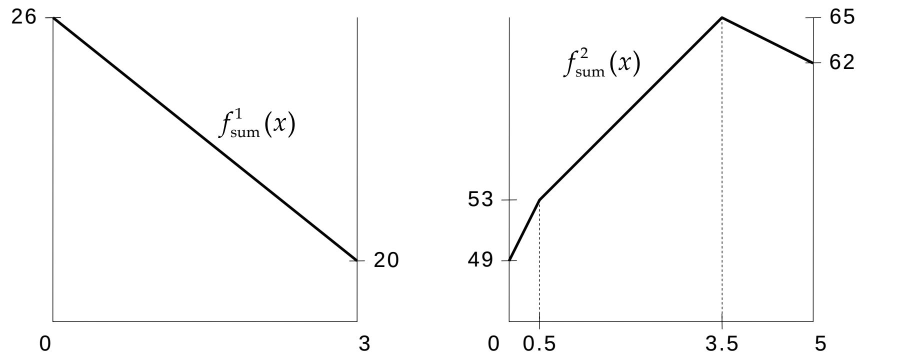


Figure II.3: Median function of edge (v_3, v_4) for the first (left) and second (right) objectives.

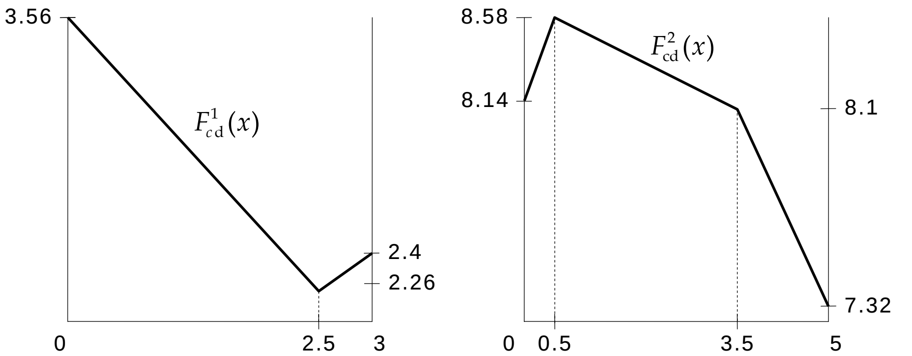


Figure II.4: Cent-dian function ($\lambda = 0.4$) of edge (v_3, v_4)
for the first (left) and second (right) objectives.

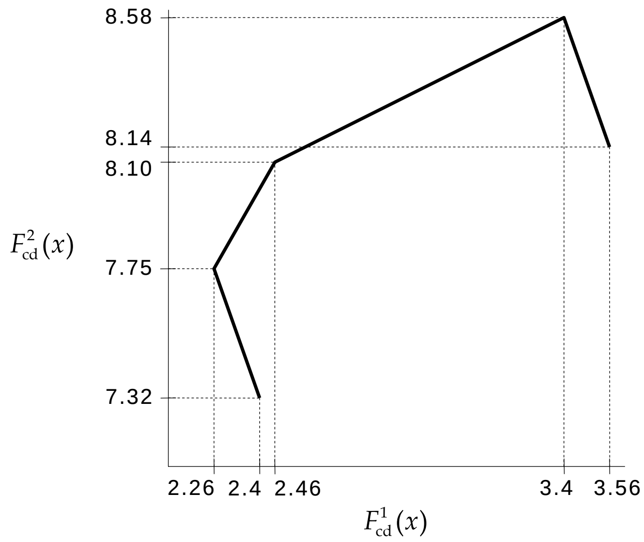


Figure II.5: Polygonal lines of edge (v_3, v_4).

II.7 Computational results

The algorithm described above has been programmed on a Sun Ultra Sparc 5 (240 Mhz) workstation with 128 Mb RAM, using the GNU C++ compiler (g++ 2.8.1) and the *Library of Efficient Datatypes and Algorithms* (LEDA 3.7.1).

The method followed for testing the goodness of this algorithm has been the generation of random planar graphs with a number of nodes n between 10 and 100, and a number of edges $m = 3n - 6$. The value of λ ranges from 0 to 1, with an increment of 0.1.

For every pair (n, λ), ten instances have been solved. Table II.3 shows the average running times (in seconds), whereas Table II.4 shows the average number of edges remaining after the elimination of non-efficient edges. We remark that the average running times for the instances are not always increasing when n is increasing. This is due to the number of edges remaining after the removal rule described in section II.4 is applied.

On the other hand, the minimum average running times are reached when $\lambda = 1$. Also, it can be seen that in all the cases the average running times are less than one minute and a half.

II.8 Conclusions

In this chapter we have proposed an $O(mn \log n)$ algorithm to solve the biobjective cent-dian problem. This procedure also allows us to solve two interesting particular cases: for $\lambda = 0$ the efficient points for the biobjective median problem are obtained, and for $\lambda = 1$ the efficient points for the biobjective center problem are determined.

We should remark that the set of efficient points to locate the λ-cent-dian could be infinite, as opposed to the uniobjective case where the λ-cent-dian is located on the set of nodes or on the set of local minima of the center function.

n	$\lambda = 0$	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
10	0.53	0.55	0.57	0.54	0.48	0.52	0.48	0.50	0.45	0.40	0.19
20	2.40	2.04	2.05	1.86	2.23	1.71	1.95	2.01	2.25	1.79	0.63
30	5.64	6.59	6.01	5.03	6.25	5.53	6.12	5.48	6.67	5.85	1.23
40	6.98	7.72	7.82	7.40	7.18	8.16	9.74	7.62	7.76	6.70	1.70
50	16.95	19.60	18.23	14.02	13.75	17.39	14.82	15.98	15.02	20.75	3.62
60	23.07	25.13	21.06	18.97	20.67	22.74	19.57	22.87	18.09	15.62	4.16
70	20.29	22.87	26.88	25.20	23.35	22.96	22.70	19.85	25.52	23.74	6.01
80	23.87	27.71	27.57	27.88	23.14	28.28	29.93	28.79	27.86	30.74	6.75
90	58.10	64.31	48.10	53.78	48.92	55.24	60.48	54.71	61.51	66.58	10.40
100	67.53	69.08	64.89	71.03	67.26	64.90	73.82	56.81	60.15	73.34	13.62

Table II.3: Average running times (in seconds) of ten instances randomly generated for every pair of (n, λ).

n	$\lambda = 0$	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
10	21	19.6	20.2	19.5	18.3	19.3	18	18.7	16.6	14.5	12.7
20	41.5	32.9	33.1	29.8	36.2	29.4	29.8	31.2	34.3	31.1	24.6
30	45.4	45.1	42	38.5	47.1	42.9	43.9	42.6	47.3	43.1	26
40	48.8	53.3	51.1	47.1	45.7	53.1	64.1	54.1	54.6	41.7	30.4
50	56.5	63.3	57.4	49.3	43.6	59.7	48	50.8	52.5	73.8	38.8
60	70.6	70.9	63	57.4	66.1	73.3	63	65.7	59.6	52.3	39
70	57.8	64.3	68.6	69.2	62.2	68.6	63.4	54.3	69.9	71.5	52
80	67.2	72.4	70.2	70.8	60.2	66.6	74.7	69.1	68.4	78.1	39.9
90	79.5	89.8	63.4	68.8	67.3	70.6	81.5	74.2	79.7	94.1	39.9
100	79	93.9	81.3	82.7	81.7	84.1	92.9	70.8	75.6	92.2	47.8

Table II.4: Average number of remaining edges for every pair of (n, λ) shown in Table II.3.

Chapter III

Multicriteria location of a 1-median facility on networks

"Most location problems are inherently multiobjective in nature"
M. Daskin

III.1 Introduction

In this chapter the emphasis is placed upon network location of one desirable facility and our aim will be the 1-median problem. We consider the demand points as corresponding to the network vertices, and we try to locate the point on the network such that it minimizes the sum of the distances to all the vertices of the network. We will study this problem considering multiple objectives, that is, the network takes multiple lengths on the edges, which implies considering multiple distance functions.

The simple 1-median problem was resolved by Hakimi since 1964, when he proved that the optimal location should be on the vertices of the network. Therefore, it is only necessary to compare distances between vertices in order to solve the problem in polynomial time.

Although researchers have paid much attention to the 1-median and, in general, to the p-median problem (locating p facilities, see Hakimi, 1965; Singer, 1968; Jarvinen, Sinervo and Rajala, 1972; Narula, Ogbu and Samuelsson, 1977; Kariv and Hakimi, 1979b), surprisingly certain generalizations of these problems, which take into account various real-life considerations, have not been studied thoroughly.

Nevertheless, a few researchers have studied some generalizations. Handler and Mirchandani (1979) gave a list of various natural generalizations that might occur, which include the consideration of probabilistic demands and costs, multi-attribute nonlinear transportation costs, multiple commodities and multiple objectives. Obviously, it is impossible to study all of these generalizations here. Instead, only the 1-median problem with multiple objectives will be analyzed.

In this sense, suppose that the demand of certain goods is concentrated in different towns represented by vertices on a road network. We assume that it is possible to consider several lengths on each edge of the network. These lengths may represent the time needed to cross the edge, the travel cost, the environmental impact, etc. Thus, the multiple criteria are expressed as the minimization of the total travel time, the sum of the travel cost, the sum of the environmental impact, etc. We wish to locate a desirable facility on the network such that the multiple criteria are optimized (multiobjective 1-median problem).

Oudjit (1981) studied the multiobjective 1-median problem on trees. He proved that the group of all multidimensional 1-median points of the considered tree is in the sub-tree formed by the union of all the minimum paths between all the pairs of 1-medians. Unfortunately, this condition is not true for any general network. In this work we will present a procedure for calculating efficient location points for the multiobjective 1-median problem on any network. According to the classification scheme given in section I.4, this problem is classified as $1/\mathcal{G}/w_m = 1/d(\mathcal{V},\mathcal{G})/Q-\sum_{\text{par}}$. A closely related paper to this chapter is presented in Hamacher, Labbé and Nickel (1999), who addressed the multicriteria median problem taking into account several weights on the nodes.

We will consider a connected and non-directed network $N(V,E)$ without loops or multiple edges, where $V=\{v_1,v_2,\dots,v_n\}$ is the set of vertices (nodes) and $E=\{(v_s,v_t):v_s,v_t\in V\}$ is the set of edges of the network. This condition does not imply loss of generality, because the located points could never be on the loop edges. The reason is that the vertex related to any loop edge would always be a better location point.

We will use the following notation:

$n=|V|$: number of vertices.
$m=|E|$: number of edges.
q: number of criteria or objectives.
l_e^r: length of edge e under criterion $r=1,2,\dots,q$.
$d^r(v_i,v_j)$: shortest path distance from v_i to v_j under criterion r.
$(l_e^1,l_e^2,\dots,l_e^q)$: vector of lengths for different criteria on edge e.

For any point x on edge $e=(v_s,v_t)\in E$ and for any criterion r, we will consider $c_e^r(x,v_s)$ as the length of the segment between x and v_s (see section I.3.2). Besides, let $d^r(x,v_i)=\min\{c_e^r(x,v_s)+d(v_s,v_i),c_e^r(x,v_t)+d(v_t,v_i)\}$ be the shortest distance from point x to any vertex v_i (see section I.3.2).

For any criterion r and each point x on N, we define

$$f^r(x)=\sum_{v_i\in V} d^r(x,v_i)$$

If x_m is a point on N so that $f^r(x_m)=\min_{x\in N} f^r(x)$, then x_m is a 1-median for the objective r.

Given the points $x,y\in N$, we say that x dominates y if, and only if, $f^r(x)\le f^r(y)$ for all r, and $f^r(x)<f^r(y)$ for at least one r. The set of efficient points is the set of all points of the network that are not dominated.

We remark that the objective functions are concave on each edge, then for any $x\in N$ and any criterion r, there exists a vertex $v_i\in V$ such that $f^r(v_i)\le f^r(x)$. In this way, the 1-median problem for simple networks is converted to the 1-median vertex problem. However, in this chapter we will see that not all multiobjective 1-medians are situated on the vertices. So, our problem is more interesting, since the possible location points could be situated all over the network.

In the following section we comment on how Hakimi's theorem, which restricts the search to the vertices of the graph, cannot be generalized to the multiobjective case. In section III.3 we present an algorithm to determine the breakpoints of the distance functions, which will be needed later on. Following this, we give an exact algorithm in polynomial time to find non-dominated (or efficient) points that are 1-medians on multiobjective networks. This algorithm requires as input data the breakpoints of the objective functions. In section III.5 a numeric example will be solved in order to help clarify the steps of the algorithm. We end this chapter with the conclusions.

III.2 Some examples and observations

We have previously commented that Hakimi proved that the 1-median problem becomes the 1-median vertex problem, using the concavity property of the objective function. We might ask if the efficient location points for the multiobjective case will always be at the vertices of the graph. The answer is negative, as we show using the network drawn in Figure III.1.

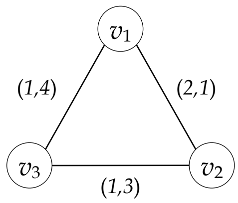


Figure III.1: A network with two lengths per edge where Hakimi's theorem does not hold.

The 1-median with respect to the first objective is vertex v_3, with objective values $(2,7)$. The 1-median with respect to the second objective is vertex v_2, with objective values $(3,4)$. However, all points on the edge (v_3, v_2) are also efficient points. For example, the middle point of this edge, with objective values $(2.5, 5.5)$, is not dominated by the above points. Therefore, Hakimi's theorem is not true for several objectives.

We may also think that all points on the shortest paths linking median vertices should be efficient points on a multiobjective network. Maybe there are efficient points on these paths but some non-efficient or dominated points may also be found on them.

For example, in the network shown in Figure III.2 the 1-medians for both objectives are the vertices v_3 and v_4, both with objective values $(10, 20)$. The shortest distance matrices for the two objectives are shown in Table III.1. However, all inner points of the edge (v_3, v_4) are dominated by the vertices v_3 and v_4. For instance, observe that the middle point has objective values $(10.5, 21)$, and therefore, none of them are efficient points for the multiobjective 1-median problem. Besides, in this case, all inner points on any edge of the network are dominated by both vertices. So, vertices v_3 and v_4 are the biobjective 1-medians.

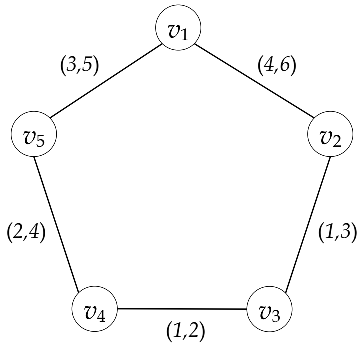


Figure III.2: A network where not all points sited on the shortest paths linking median vertices are efficient.

	Objective 1						*Objective 2*					
	v_1	v_2	v_3	v_4	v_5	Sum	v_1	v_2	v_3	v_4	v_5	Sum
v_1	0	4	5	5	3	17	0	6	9	9	5	29
v_2	4	0	1	2	4	11	6	0	3	5	9	23
v_3	5	1	0	1	3	10	9	3	0	2	6	20
v_4	5	2	1	0	2	10	9	5	2	0	4	20
v_5	3	4	3	2	0	12	5	9	6	4	0	24

Table III.1: Distance matrices of the network shown in Figure III.2.

Now, we may ask ourselves if all efficient points are only on the shortest paths linking 1-median vertices or whether some of these efficient points could be also found outside. As it can be seen in Figure III.3, vertex v_2, with objective values $(8,5)$, is the 1-median with respect to the first objective, whereas vertex v_3 with objective values $(11,3)$ is the other 1-median with respect to the second objective.

The shortest path linking vertices v_2 and v_3 for both objectives is the edge (v_2, v_3). However, vertex v_1 with objective values $(9,4)$ is also an efficient point and it is not on that shortest path.

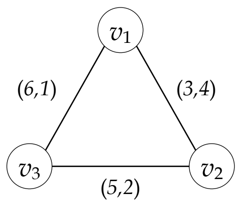


Figure III.3: A network with two lengths per edge in which there are efficient points outside the shortest paths linking 1-median vertices.

The following question is whether efficient points should be only on those edges that contain any 1-median vertex. The answer is negative. The network drawn in Figure III.4 is a good example for answering this question. We have calculated the shortest distance matrices for the two objectives. These matrices are shown in Table III.2.

The 1-median for the first objective is vertex v_5 with objective values $(12,14)$, while the 1-median for the second objective is vertex v_4 with objective values $(16,9)$. However, vertex v_1

with objective values (13,12) and vertex v_3 with objective values (14,10) are also efficient points for the biobjective 1-median problem. Therefore, efficient points for the multiobjective 1-median problem could be located on any edge of the network.

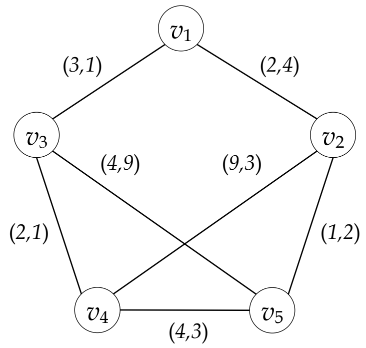


Figure III.4: A network where there are efficient location points outside the set of edges incident to any 1-median vertex.

	Objective 1						*Objective 2*					
	v_1	v_2	v_3	v_4	v_5	Sum	v_1	v_2	v_3	v_4	v_5	Sum
v_1	0	2	3	5	3	13	0	4	1	2	5	12
v_2	2	0	5	5	1	13	4	0	4	3	2	13
v_3	3	5	0	2	4	14	1	4	0	1	4	10
v_4	5	5	2	0	4	16	2	3	1	0	3	9
v_5	3	1	4	4	0	12	5	2	4	3	0	14

Table III.2: Distance matrices of the network shown in Figure III.4.

III.3 Efficient points for the multiobjective 1-median problem

In order to simplify the search for efficient points, we now propose a simple rule to eliminate edges of the network. Since the objective functions are concave on each edge, an edge $e = (v_s, v_t) \in E$ can be removed if the following condition is satisfied:

$$f^r(v_s) \geq f^r(v_m) \quad \text{and} \quad f^r(v_t) \geq f^r(v_m), \quad \forall r = 1,2,\ldots,q$$

where v_m is any median vertex for some criterion *r*.

Otherwise, it is possible to check if there exist efficient points on this edge. Therefore, the edges which join 1-median vertices are never removed.

Next we explain the search procedure for efficient points on a multiobjective network. This procedure will be applied to the remaining edges. This method is based on two algorithms. The first algorithm determines the distance functions for each objective. These functions are concave polygonals and they will be completely characterized when we know the breakpoints $(b_i^r, f^r(b_i^r))$ for each objective *r* of the polygonal lines where the slope changes its value. This method is closely related to the one presented in Algorithm II.2, but now we consider several objectives and the nodes are unweighted.

The second algorithm uses the breakpoints of the objective functions and splits the edges into segments according to those points corresponding to maximum values of the objectives. Then non-dominated points for each segment are determined, and the value vectors of the points obtained are compared in order to remove the dominated ones.

function *Median*(Network $N(V,E)$, DistanceMatrix *d*, Parameter *q*)
{ *// After removing all useless edges, do for all the remaining edges and q costs.*
  for all remaining edges $e := (v_s, v_t) \in E$ **do**
    for each objective $r := 1$ to q **do**
      { Let $f^r(v_s) := \sum_{v_i \in V} d^r(v_s, v_i)$ and $f^r(v_t) := \sum_{v_i \in V} d^r(v_t, v_i)$
        for all nodes $v_i \in V$ **do** *// Compute all the n bottleneck points (see section I.3.1).*
          $b_i^r := (d^r(v_t, v_i) + l_e^r - d^r(v_s, v_i)) / 2$
        Sort points b_i^r in increasing order. Let $b_1^r, b_2^r, \ldots, b_p^r$ be their *p* possible values.
        for $i := 1$ to p **do**
          $f^r(b_i^r) := \sum_{v_k \in A} (b_i^r + d^r(v_s, v_k)) + \sum_{v_k \in B} (l_e^r - b_i^r + d^r(v_t, v_k))$
            where $A = \{v_k \in V : b_i^r \le b_k^r\}$, $B = \{v_k \in V : b_i^r > b_k^r\}$
        Draw lines linking points $(b_i^r, f^r(b_i^r))$, with $(v_s, f^r(v_s))$ and $(v_t, f^r(v_t))$
      }
  return $f^r(x), \forall x \in N, r = 1, \ldots, q$
}

Algorithm III.1: The unweighted median function.

Given the distance matrix *d*, the complexity of the Algorithm III.1 is $O(mqn \log n) + O(qmn + qn^2 \log n)$, where *m* is the number of the edges, *q* is the number of objectives and *n* the number of vertices. The computation of the shortest distance matrices for the *q* objectives requires $O(qmn + qn^2 \log n)$ time using Fredman and Tarjan (1987), while sorting the points b_i^r is performed in at most $O(n \log n)$ time.

Next, we propose the algorithm that determines the efficient location points. This algorithm needs as input the results obtained from the algorithm given above. The objective functions for each edge are known, because they were obtained in Algorithm III.1. Then, both the set of points *P* and the set of segments *S* are defined. Finally, these sets are compared to get the non-dominated or efficient points on the network by calling Algorithm III.3 and Algorithm III.4. These functions perform a straight comparison, respectively, among all points and between points and segments, storing the non-dominated ones.

Regarding Algorithm III.7, the comparison is performed between the segments stored in set *S*. Such comparison is not as easy as the preceding algorithms. Therefore, it will be thoroughly explained in a subsequent section.

The total complexity of Algorithm III.2 is $O(m^2q^3)$. This complexity is calculated as follows. On each edge, there are at most $q+1$ segments $[x_i, x_{i+1}]$. The number of segments and points to compare will be $O(mq)$. Since we need at most $\binom{mq}{2}$ comparisons, and each comparison in Algorithm III.3, Algorithm III.4 and Algorithm III.7 requires at most $O(q)$ time, then the overall complexity is $O(m^2q^3)$.

```
function MultiobjectiveMedian(Network N(V,E), DistanceMatrix d, Parameter q)
{   // Let P be the set of candidate points to be non-dominated.
    P := ∅
    // Let S be the set of possible non-dominated segments.
    S := ∅
    for all remaining edges e := (v_s, v_t) ∈ E do
        {   for each objective r := 1 to q do
                Determine the maximal value and the point related to this maximum
            Sort the maximum points in increasing order with respect to the first objective
                (not including the repeated points or the extreme points v_s and v_t)
            Let x_1, x_2, ..., x_p be the selected points
            for i := 1 to p do
                Calculate f(x_i) := (f^1(x_i), f^2(x_i), ..., f^q(x_i))
            Split edge e into segments given by the partition [v_s = x_0, x_1, ..., x_p, x_{p+1} = v_t]
            for i := 0 to p do
                {   Let [x_i, x_{i+1}] be a segment of edge e
                    if f^r(x_i) ≤ f^r(x_{i+1}), ∀r = 1,...,q then          P := P ∪ {x_i}
                    else if f^r(x_i) ≥ f^r(x_{i+1}), ∀r = 1,...,q then     P := P ∪ {x_{i+1}}
                    else    P := P ∪ {x_i} ∪ {x_{i+1}} and S := S ∪ {[x_i, x_{i+1}]}
                }
        }
    Compare the points in P using Algorithm III.3 and store in set P_ND the
        non-dominated points obtained
    Compare segments in S using Algorithm III.7 and store in set S_ND the
        non-dominated segments obtained
    Compare the points of P_ND with segments in S_ND using Algorithm III.4, storing
        what is non-dominated
    return P_ND and S_ND
}
```

Algorithm III.2: The multiobjective 1-median function.

```
function PointComparison(PointSet P)
{   // Let {x_1, x_2, ..., x_p} be the points belonging to P, and P_ND the set of non-dominated points.
    P_ND := ∅
    for i := 1 to p do
        {   Let x_i ∈ P be a point
            if ∄x_j ∈ P_ND : x_j ≺ x_i then
                {   P_ND := P_ND ∪ {x_i}
                    if ∃x_k ∈ P_ND : x_i ≺ x_k then
                        P_ND := P_ND / {x_k}
                }
        }
    return P_ND
}
```

Algorithm III.3: The point comparison function.

function *PointAgainstSegmentComparison*(PointSet *P*, SegmentSet *S*)
{ $P_{\mathrm{ND}} := P$, $S_{\mathrm{ND}} := S$
  for all points $z \in P_{\mathrm{ND}}$ **do**
    for all segments $X := [x_0, x_1] \in S_{\mathrm{ND}}$ **do**
      { **if** $z \prec X$ **then**
        { Let $[x_{\min}, x_{\max}] \in X$ be the interval dominated by point z
          $X := X / [x_{\min}, x_{\max}]$
        }
        if $X \prec z$ **then** $P_{\mathrm{ND}} := P_{\mathrm{ND}} / \{z\}$
      }
  return P_{ND} and S_{ND}
}

Algorithm III.4: Comparing points against segments.

III.4 Segment vs. segment comparison

First, each segment $[x_i, x_{i+1}]$ of set S is divided into segments $[x_i, b_j] \cup [b_j, b_k] \cup \cdots \cup [b_p, x_{i+1}]$ with one single objective function line over them, where $b_j, b_k, \ldots, b_p$ are the breakpoints of the q objective functions with respect to the first objective. For example, the segment shown in Figure III.5 is split into $[x_i, x_{i+1}] = [x_i, b_1] \cup [b_1, b_2] \cup \cdots \cup [b_4, x_{i+1}]$.

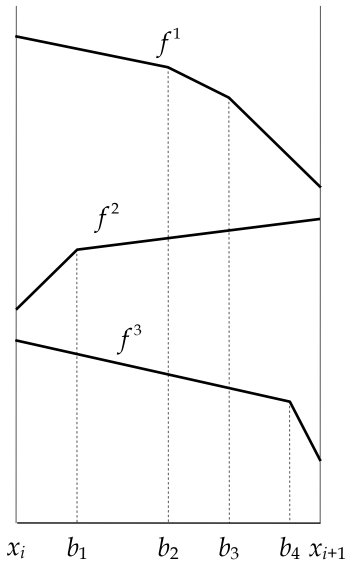


Figure III.5: Segment with three objective functions and four inner breakpoints.

Given any two segments $X = [x_0, x_1] \in S$ and $Y = [y_0, y_1] \in S$, and two inner points $x \in X$ and $y \in Y$, the q objective functions are of the form

$$f_X^r(x) = f_X^r(x_0) + m_X^r(x - x_0), \quad f_Y^r(y) = f_Y^r(y_0) + m_Y^r(y - y_0), \quad \forall r = 1, \ldots, q$$

If X dominates Y ($X \prec Y$), then the following inequalities must be fulfilled:

$$\begin{aligned}
f_X^1(x) \le f_Y^1(y) &\Rightarrow f_X^1(x_0)+m_X^1(x-x_0) \le f_Y^1(y_0)+m_Y^1(y-y_0)\\
f_X^2(x) \le f_Y^2(y) &\Rightarrow f_X^2(x_0)+m_X^2(x-x_0) \le f_Y^2(y_0)+m_Y^2(y-y_0)\\
&\vdots\\
f_X^j(x) < f_Y^j(y) &\Rightarrow f_X^j(x_0)+m_X^j(x-x_0) < f_Y^j(y_0)+m_Y^j(y-y_0)\\
&\vdots\\
f_X^q(x) \le f_Y^q(y) &\Rightarrow f_X^q(x_0)+m_X^q(x-x_0) \le f_Y^q(y_0)+m_Y^q(y-y_0)
\end{aligned}$$

Therefore, for any inequality i we get

$$f_X^i(x_0)+m_X^i(x-x_0) \le f_Y^i(y_0)+m_Y^i(y-y_0) \Rightarrow y \ge \frac{f_X^i(x_0)-f_Y^i(y_0)-m_X^i x_0+m_Y^i y_0}{m_Y^i}+\frac{m_X^i}{m_Y^i}x \quad \text{(III.1)}$$

For example, given the two segments $X = Y = [0,1]$ drawn in Figure III.6, with $f_X^i(x) = 3-2x$, and $f_Y^i(y) = 4-3y$, we can confront the values of both intervals to get the pair of points $(x \in X, y \in Y)$ at which either $X \prec Y$ or $Y \prec X$.

If $X \prec Y$ then $3-2x < 4-3y$, and hence, $y < (1/3)+(2/3)x$. The pair of values (x,y) for which X dominates Y are enclosed in the shaded region to the right of the bold line in Figure III.7. This line is the projection of the intersection between the planes $f_X^i(x)$ and $f_Y^i(y)$, being its equation precisely $y = (1/3)+(2/3)x$. The shaded region to the left of the bold line represents the values of (x,y) where $Y \prec X$.

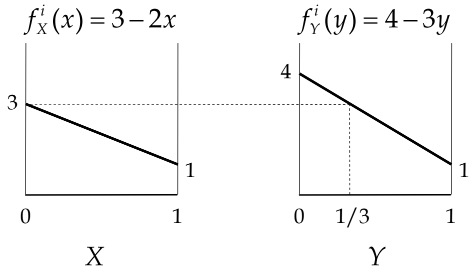


Figure III.6: Two different segments X and Y with their objective function values.

Let $p^i = (f_X^i(x_0)-f_Y^i(y_0)-m_X^i x_0+m_Y^i y_0)/m_Y^i$ and $q^i = m_X^i / m_Y^i$. Then, (III.1) is rewritten as $y \ge p^i + q^i x$. According to the values p^i and q^i, the next type of inequalities arise:

- If $m_X^i = 0 \Rightarrow q^i = 0 \Rightarrow \begin{cases} y \le p^i : \text{ type } \boxed{e} \\ y \ge p^i : \text{ type } \boxed{f} \end{cases}$
- If $q^i > 0 \Rightarrow \begin{cases} y \le p^i + q^i x : \text{ type } \boxed{a} \\ y \ge p^i + q^i x : \text{ type } \boxed{b} \end{cases}$
- If $q^i < 0 \Rightarrow \begin{cases} y \le p^i + q^i x : \text{ type } \boxed{c} \\ y \ge p^i + q^i x : \text{ type } \boxed{d} \end{cases}$
- In the particular case in which $m_Y^i = 0 \Rightarrow q^i = \infty$, and hence, the inequality holds with respect to x, that is $\begin{cases} x \le u^i : \text{ type } \boxed{g} \\ x \ge u^i : \text{ type } \boxed{h} \end{cases}$, with $u^i = (f_Y^i(y_0)-f_X^i(x_0)+m_X^i x_0)/m_X^i$.

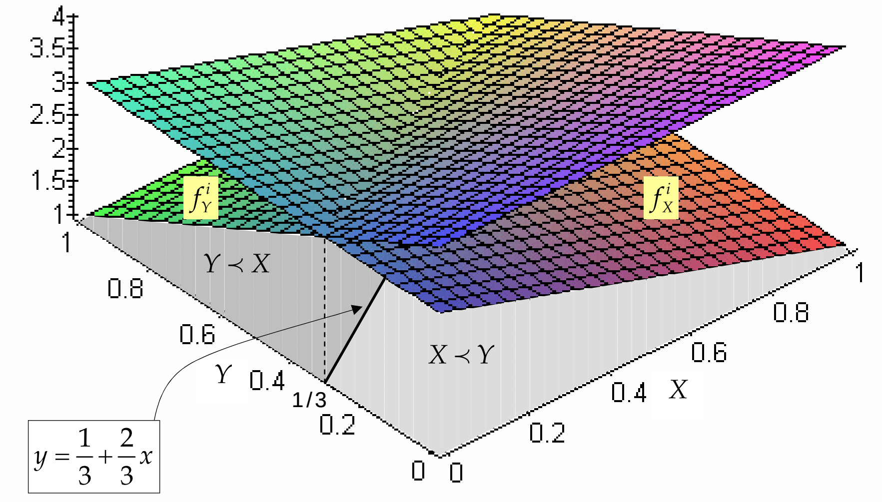


Figure III.7: 3D projection of f_X^i against f_Y^i and the regions where X and Y dominate each other.

Figure III.8 shows the eight types of inequalities and the region R where $X \prec Y$. The types a, c, e, and g are less-or-equal ($\leq$) inequalities, whereas types b, d, h and f are greater-or-equal ($\geq$) inequalities. This is drawn with a dotted line across region R.

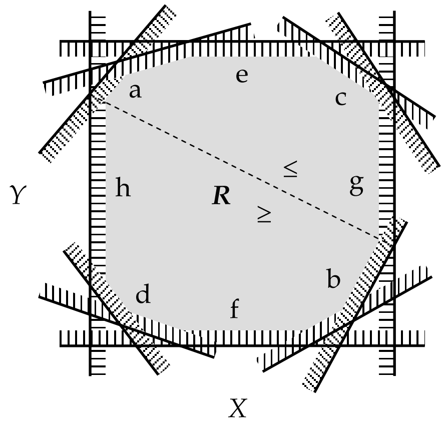


Figure III.8: The eight different types of inequalities with the domination region R.

Let T be the set of all inequalities, being $T_a, T_b, \ldots, T_h$ the sets of different inequalities with $T = T_a \cup T_b \cup \cdots \cup T_h$. Each inequality is denoted by the letter of the type it belongs to, namely $a \in T_a$, etc. Obviously, if there are any two inequalities $a \in T_a$ and $b \in T_b$ such that $a(x) < b(x)$, $\forall x \in [x_0, x_1]$, then region R is empty, and hence $X \nprec Y$. The following Lemma III.1 states this result for all inequalities in T, as shown in Figure III.9.

Lemma III.1. *If there are inequalities* $a \in T_a, b \in T_b, \ldots, h \in T_h$, *such that* $a(x) < b(x)$, $c(x) < d(x)$, $e(x) < f(x)$ *or* $g(y) < h(y)$, *for all points* $x \in X$ *and* $y \in Y$, *then* $X \nprec Y$.

Proof. Any of these latter conditions make region R to be empty, and hence $X \nprec Y$. ■

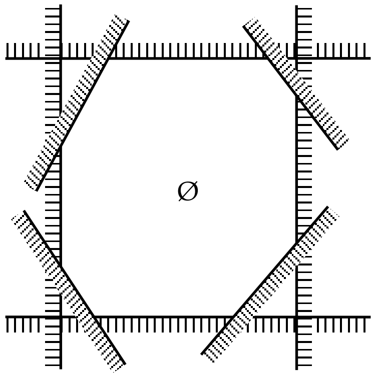


Figure III.9: An example of an empty domination region.

It is plain to see that there is a close connection between the convex region *R* defined by the set of inequalities *T* and a two-variable linear programming problem. This fact could lead us to solve the segment comparison using linear programming algorithms such as the simplex. However, as we show in the next pages, this problem can be readily solved by computational geometry techniques.

Once we have classified the inequalities, we proceed to find the points in segment *X* that dominate points on segment *Y*. For instance, consider the next region *R* in Figure III.10, in which some points $x \in X$ dominate some points $y \in Y$. Indeed, all points $x \in [x_{\min}, x_{\max}]$ dominate all points $y \in [y_{\min}, y_{\max}]$, that is, $[x_{\min}, x_{\max}] \prec [y_{\min}, y_{\max}]$. Our goal is to find these two values in segment *Y*.

In the subsequent analysis we first compute $y_{\max}$, and by means of a classic result, we then get $y_{\min}$. When some of the inequality sets in *T* are empty, the value of $y_{\max}$ is easily calculated, as stated in the next result.

Lemma III.2. *If* $T_a = \varnothing$ *and* $T_c = \varnothing$ *then* $y_{\max} = y_1$. *When* $T_a = \varnothing$ *we get* $y_{\max} = \min_{c \in T_c} c(x_0)$, *with* $x_{\max} = x_0$. *Likewise, if* $T_c = \varnothing$, $y_{\max} = \min_{a \in T_a} a(x_0)$, *with* $x_{\max} = x_1$.

Proof. The proof is straightforward. ■

Otherwise, $T_a \neq \varnothing$ and $T_c \neq \varnothing$, and therefore, the value $y_{\max}$ is attained at the intersection point between two inequalities of T_a and T_c. In this sense, given two inequalities $a \in T_a$ and $c \in T_c$ we define $x = I(a,c) \in X$ as the intersection point between them. Let $Q = \{I(a,c) : \forall a \in T_a, \forall c \in T_c\}$ be all the intersection points among all inequalities in T_a and T_c. Let $F(x) = \{a(x) : \forall a \in T_a\}$ be the set of inequalities with positive slope.

We assume that there is at least one intersection between an inequality of T_a and another of T_c. If not, it means that all inequalities in T_a are below T_c, or vice versa, and hence the value $y_{\max}$ can be obtained using Lemma III.2.

Besides, we also assume that the intersection point takes place below y_1. Otherwise, the next result states the value of $(x_{\max}, y_{\max})$.

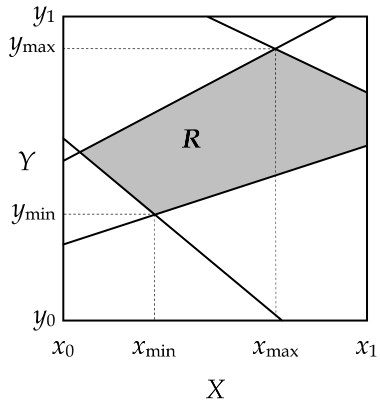


Figure III.10: Inside region *R*, $[x_{\min}, x_{\max}] \prec [y_{\min}, y_{\max}]$.

Lemma III.3. *If all the intersections between the inequalities in* T_a *and* T_c *take place above* y_1*, then* $y_{\max} = y_1$*, being* $[x^0_{\max}, x^1_{\max}]$ *the interval where this maximum value is achieved, with* $x^0_{\max} = \max\{x \in X : t(x) = y_1, t \in T_a\}$ *and* $x^1_{\max} = \min\{x \in X : t(x) = y_1, t \in T_c\}$.

Proof. The proof is straightforward. ■

Taking into account these latter assumptions, there must be a point $z \in Q$ such that $F(z) = \min_{x \in Q} F(x)$. Therefore, $x_{\max} = z$ and $y_{\max} = F(z)$. The next Lemma proves this result.

Lemma III.4. $y_{\max} = F(z)$ *and hence* $x_{\max} = z$.

Proof. Being *R* a convex region, the maximal value $y_{\max}$ is attained at the intersection (extreme point) of two inequalities with opposite sign slope. Let $a_m \in T_a$ and $c_m \in T_c$ be those two inequalities. Any other inequality $a \in T_a$ or $c \in T_c$ will be over a_m and c_m, since $F(z)$ is the minimal value of all the intersection points *Q*. To be precise, $a(z) \geq F(z)$, $\forall a \in T_a$, and $c(z) \geq F(z)$, $\forall c \in T_c$. If there is any $a^* \in T_a$, with $a^* \neq a_m$, such that $a^*(z) < F(z)$, then $x^* = I(a^*, c_m)$ and hence, $F(x^*) < F(z)$, which contradicts that $F(z)$ is the minimal value. The same analysis can be applied to an inequality $c^* \in T_c$, and thus the result follows. ■

From this proof we can immediately derive the following consequence.

Corollary III.1. *Provided that all intersection points fall inside* $X \times Y$*, let* $a \in T_a$ *and* $c \in T_c$*, with* $x = I(a,c)$ *and* $y = F(x)$*. If* $a_m(x) < y$ *then* $F(I(a_m, c)) < y$*, and if* $c_m(x) < y$ *then* $F(I(a, c_m)) < y$ (see Figure III.11).

This latter result will be subsequently used in the algorithm to speed up the search process of $(x_{\max}, y_{\max})$. Finally, once we have computed the value $y_{\max}$, if $y_{\max} < y_0$ then region *R* is empty, and consequently $X \nprec Y$.

To obtain the minimal value $y_{\min}$, we can apply Lemma III.2, Lemma III.3 and Lemma III.4 on inequalities T_d and T_b, and the classical optimization result that establishes $\min(y) = -\max(-y)$. Let d_m and b_m the inequalities whose intersection yields $x_{\min} = I(d_m, b_m)$, with $y_{\min} = d_m(x_{\min})$.

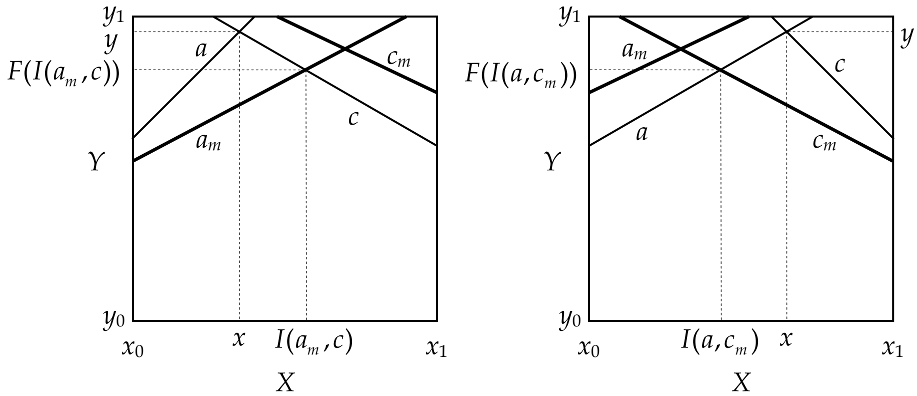


Figure III.11: Illustration of Corollary III.1.

As soon as we have obtained the values $y_{\min}$ and $y_{\max}$, segment X does not dominate segment Y if $y_{\min} > y_{\max}$. Otherwise, we can now check if these maximal and minimal values can be reached by the intersection of inequalities T_a and T_b or T_c and T_d, respectively.

If $x_{\min} < x_{\max}$, the following two situations depicted in Figure III.12 may arise.

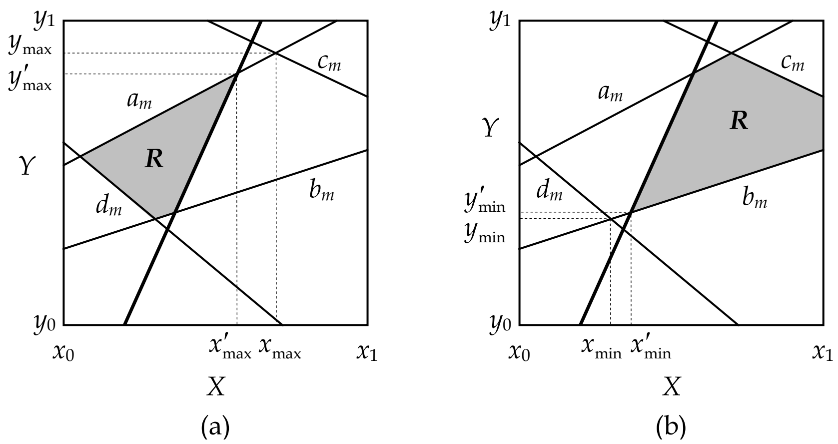


Figure III.12: Two examples where new maximum and minimum values need to be computed.

Before calculating the new values $y'_{\min}$ and $y'_{\max}$, the next result states a condition for which segment X will not dominate segment Y.

Lemma III.5. *If* $x_{\min} < x_{\max}$ *and* $a_m(x_{\min}) < y_{\min}$ *and* $b_m(x_{\max}) > y_{\max}$, *then* $X \not\prec Y$.

Proof. The proof follows since a_m is completely below b_m, and hence region R is empty. ■

We now search for the new maximum point $y'_{\max}$ among the crossing points of inequalities T_a and T_b. Firstly, we delete from T_b all useless inequalities:

$$T'_b = T_b / \{b \in T_b : b(x_{\max}) \le y_{\max}\}$$

We define a new set T'_b because the inequalities in T_b are used later to obtain $y'_{\min}$. If $T'_b = \varnothing$ then there is no $b \in T_b$ such that $b(x_{\max}) > y_{\max}$, and thus, the maximum point $(x_{\max}, y_{\max})$ remains unchanged.

Otherwise, we proceed to get the new maximal point $(x'_{\max}, y'_{\max})$ by means of a similar result to that of Lemma III.4. Let $Q' = \{I(a,b): \forall a \in T_a, \forall b \in T'_b, \text{slope}(a) < \text{slope}(b)\}$ be the set of intersection points between inequalities T_a and T'_b, where slope(a) and slope(b) denote the slopes of the line segment of each inequality. This requirement is important since we want the T'_b inequalities to cross the T_a inequalities as high as possible. Therefore, since $T'_b \neq \varnothing$, there must be at least one point $z' \in Q'$ such that $F(z') = \min_{x \in Q'} F(x)$, and accordingly we state the subsequent result.

Lemma III.6. $y'_{\max} = F(z')$ *and hence* $x'_{\max} = z'$.

Proof. Given that $T'_b \neq \varnothing$, then $\exists b^* \in T'_b$ with $x^* = I(a_m, b^*)$ such that $F(x^*) < y_{\max}$ (see Figure III.12a). Any other inequality $b \in T'_b$ fulfilling $b(x^*) > F(x^*)$, or any inequality $a \in T_a$ with $a(x^*) < F(x^*)$ might improve the value of $F(x^*)$. Hence, $\exists a'_m \in T_a$ and $\exists b'_m \in T'_b$ with $\text{slope}(a'_m) < \text{slope}(b'_m)$ and $z' = I(a'_m, b'_m)$ such that $y'_{\max} = F(z')$ with $x'_{\max} = z'$. ■

As in Corollary III.1, a result that can improve the search of $(x'_{\max}, y'_{\max})$ is derived from the preceding proof.

Corollary III.2. *Assuming that all intersection points fall inside* $X \times Y$, *let* $a \in T_a$ *and* $b \in T'_b$, *with* $x = I(a,b)$ *and* $y = F(x)$. *If* $a'_m(x) < y$ *then* $F(I(a'_m, b)) < y$, *and if* $b'_m(x) > y$ *then* $F(I(a, b'_m)) < y$.

We now try to tighten $y_{\min}$ searching for a new value $y'_{\min}$ in the intersection points between inequalities T_a and T_b (see Figure III.12b). Initially, we get rid of all useless inequalities in T_a.

$$T_a = T_a / \{a \in T_a : a(x_{\min}) \geq y_{\min}\}$$

In this case, there is no need to create a new set T'_a, since T_a will not be used later. If $T_a = \varnothing$, there is no inequality in T_a that can improve $y_{\min}$. Otherwise, the new minimum value $y'_{\min}$ can be obtained in a similar way to Lemma III.6 along with the fact that $\min(y) = -\max(-y)$.

Finally, when $x_{\min} > x_{\max}$ we obtain analogous situations to those in Figure III.12a and b, barring they now take place on the right-hand side.

Lemma III.7. *If* $x_{\min} > x_{\max}$ *and* $c_m(x_{\min}) < y_{\min}$ *and* $d_m(x_{\max}) > y_{\max}$, *then* $X \not\prec Y$.

Proof. The proof follows since c_m is completely below d_m, and hence region *R* is empty. ■

If this result is not held, we continue to get the new maximum value $y'_{\max}$. As we did above, first we remove from T_d all inequalities below $y_{\max}$:

$$T'_d = T_d / \{d \in T_d : d(x_{\max}) \leq y_{\max}\}$$

Then, if $T'_d \neq \varnothing$ we search for $y'_{\max}$ in the intersection points between inequalities T_c and T'_d. After this is accomplished, we eliminate all useless inequalities from T_c:

$$T_c = T_c / \{c \in T_c : c(x_{\min}) \geq y_{\min}\}$$

and search for $y'_{\min}$ in the intersection points of T_d and T_c.

The following procedure Algorithm III.5 gathers all the results previously described.

function *Dominate*(InequalitySet *T*, Interval $X = [x_0, x_1]$, Interval $Y = [y_0, y_1]$)
{ Classify all inequalities in $T = T_a \cup T_b \cup \cdots \cup T_h$
  // Bound X and Y.
  $X := [\max\{x_0, \max_{t \in T_h} t\}, \min\{x_1, \max_{t \in T_g} t\}]$
  $Y := [\max\{y_0, \max_{t \in T_f} t\}, \min\{y_1, \max_{t \in T_e} t\}]$
  if $x_0 > x_1$ **or** $y_0 > y_1$ **then return** $X \nprec Y$
  $y_{\max} := y_1$
  if $T_a \neq \varnothing$ **and** $T_c \neq \varnothing$ **then**
    { **if** there is an intersection point between T_a and T_b below y_1 **then**
      { **if** Lemma III.2 holds **then** Store solution in $(x_{\max}, y_{\max})$
        else $(x_{\max}, y_{\max}) :=$ *ComputeMaximum*(T_a, T_b, y_0)
      }
      else Apply Lemma III.3 to get $[x^0_{\max}, x^1_{\max}]$
    }
  $y_{\min} := y_0$
  if $T_d \neq \varnothing$ **and** $T_b \neq \varnothing$ **then**
    { $T'_d := \{-d(x) : \forall d \in T_d\}$, $T'_b := \{-b(x) : \forall b \in T_b\}$
      if there is an intersection point between T'_d and T'_b below $-y_0$ **then**
      { **if** Lemma III.2 holds for T'_d and T'_b **then** Store solution in $(x_{\min}, y_{\min})$
        else $(x_{\min}, y_{\min}) :=$ *ComputeMaximum*$(T'_d, T'_b, -y_1)$
        $y_{\min} := -y_{\min}$
      }
      else Apply Lemma III.3 on T'_b and T'_d to get $[x^0_{\min}, x^1_{\min}]$
    }
  if $y_{\min} > y_{\max}$ **then return** $X \nprec Y$
  if $x_{\min} < x_{\max}$ **then**
    { *// Check Lemma III.5.*
      if $a_m(x_{\min}) < y_{\min}$ **and** $b_m(x_{\max}) > y_{\max}$ **then return** $X \nprec Y$
      else
      { $T'_b := T_b / \{b \in T_b : b(x_{\max}) \leq y_{\max}\}$
        if $T'_b \neq \varnothing$ **then** $(x_{\max}, y_{\max}) :=$ *ComputeMaximum*(T_a, T'_b)
        $T_a := T_a / \{a \in T_a : a(x_{\min}) \geq y_{\min}\}$
        if $T_a \neq \varnothing$ **then**
          { $T'_b := \{-b(x) : \forall b \in T_b\}$, $T'_a := \{-a(x) : \forall a \in T_a\}$
            $(x_{\min}, -y_{\min}) :=$ *ComputeMaximum*(T'_b, T'_a)
          }
      }
    }
…

```
…
    else if x_min > x_max then
        { // Check Lemma III.7.
            if c_m(x_min) < y_min and d_m(x_max) > y_max then return X ⊀ Y
            else
                { T'_d := T_d / {d ∈ T_d : d(x_max) ≤ y_max}
                    if T'_d ≠ ∅ then (x_max, y_max) := ComputeMaximum(T_c, T'_d)
                    T_c := T_c / {c ∈ T_c : c(x_min) ≥ y_min}
                    if T_c ≠ ∅ then
                        { T'_d := {−d(x) : ∀b ∈ T_d}, T'_c := {−c(x) : ∀c ∈ T_c}
                            (x_min, −y_min) := ComputeMaximum(T'_d, T'_c)
                        }
                }
        }
    return [y_min, y_max]
}
```

Algorithm III.5: The algorithm to compare segments *X* and *Y*, and to check whether $X \prec Y$.

Next, we explain the procedure *ComputeMaximum* (Algorithm III.6). All previous results establish both the minimal and maximal points inside *R* where *X* dominates *Y*. Such results are based on the comparison of values over the intersection points among the inequalities. In the case of Lemma III.4, the inequalities taken into account are T_a and T_c. The set of intersection points forms the set *Q*. If there is one single inequality for each objective $r = 1,\ldots,q$, there are at most $\mathrm{O}(q)$ inequalities in *R*. Therefore, $|Q| \leq q^2$. However, below we show that both $(x_{\min}, y_{\min})$ and $(x_{\max}, y_{\max})$ can be computed in $\mathrm{O}(q)$ time.

We begin analyzing the computation of $(x_{\max}, y_{\max})$. Let $M = \{(a,c) : a \in T_a, c \in T_c\}$ be a set of pairs (matchings) of inequalities T_a and T_c such that $|M| = \max\{|T_a|, |T_c|\}$, with $|M| \leq |Q|$. For example, let $T_a = \{a_1, a_2, a_3, a_4\}$ and $T_c = \{c_1, c_2\}$. Then, $M = \{(a_1,c_1),(a_2,c_2),(a_3,c_1),(a_4,c_2)\}$, with $|M| = 4 = |T_a|$.

Each pair of inequalities $(a,c) \in M$ yields a point $x = I(a,c)$ and a value $y = F(x)$. Let $x_m \in X$ be a point such that $F(x_m) = \min_{(a,c)\in M} F(I(a,c))$. This point x_m might be the optimal, with $x_{\max} = x_m$, $y_{\max} = y_m = F(x_m)$, and a_m and c_m being the inequalities which cross at this maximum. Accordingly, all inequalities in T_a and T_c are then deleted. Otherwise, there may still be some inequalities below a_m and c_m.

Let $a^* \in T_a : a^*(x_m) = \min_{a\in T_a} a(x_m)$ and $c^* \in T_c : c^*(x_m) = \min_{c\in T_c} c(x_m)$ be the lowest inequalities underneath $F(x_m)$. Let $x_m = I(a^*, c^*)$ and $y_m = F(x_m)$. This value is the new optimal point. Furthermore, we can now remove from *M*, in the worst case, one inequality *a* or *c* from each pair $(a,c) \in M$. Indeed, each pair may have one single inequality under y_m, that is, either $a(x_m) < y_m$ or $c(x_m) < y_m$. Both inequalities cannot be below since it contradicts the fact that (x_m, y_m) is the minimal point. Therefore, at least $|M|/2 = \max\{|T_a|, |T_c|\}/2$ inequalities are deleted. This analysis proves the following result.

Lemma III.8. *In each search of the optimal point* (x_m, y_m) *we can remove at least* $|M|/2$ *inequalities from M.*

```
function ComputeMaximum(InequalitySet T_L, InequalitySet T_R, Value y_limit)
{   Choose any l_m ∈ T_L and r_m ∈ T_R that intersect inside X × Y
    x_m := I(l_m, r_m), y_m := F(x_m)
    while T_L ≠ ∅ and T_R ≠ ∅ do
      {  Let M := {(l, r) : l ∈ T_L, r ∈ T_R} be a matching between inequalities in T_L and T_R
            such that |M| := max{|T_L|, |T_R|}
         for all the pairs (l, r) ∈ M do
            {  x := I(l, r), y := F(x)
               Check if X ⊀ Y using Lemma III.1
               // Try to improve value 'y' by Corollary III.1 or Corollary III.2.
               if F(I(l_m, r)) < y then l := l_m
               if F(I(l, r_m)) < y then r := r_m
               if y has been improved then Recompute x := I(l, r) and y := F(x)
               // Check if intersection is below lower limit.
               if y < y_limit then return X ⊀ Y
               // Store the minimum value found so far.
               if y < y_m then
                  {  y_m := y, x_m := x
                     l_m := l, r_m := r
                  }
            }
         // Look for the lower inequalities under y_m.
         for all l ∈ T_L do
            if l is below l_m then l_m := l
         for all r ∈ T_R do
            if r is below r_m then r_m := r
         Check if X ⊀ Y using Lemma III.1
         x_m := I(l_m, r_m), y_m := F(x_m)
         for all l ∈ T_L do
            if l is over y_m then T_L := T_L / {l}
         for all r ∈ T_R do
            if r is over y_m then T_R := T_R / {r}
      }
    return (x_m, y_m) and l_m, r_m
}
```

Algorithm III.6: The algorithm to compute the maximum dominated value inside *R*.

Finally, the following theorem states the theoretical complexity of the *ComputeMaximum* algorithm.

Theorem III.1. *The ComputeMaximum algorithm runs in* $O(q)$ *time.*

Proof. Each inequality set has at most q elements. According to Lemma III.8, each iteration of the 'while' loop removes at least $|M|/2$ inequalities from *M*. Thus, the number of inequalities processed within this loop is

$$q+\frac{q}{2}+\frac{q}{4}+\cdots+\frac{q}{2^k}=q\left(\frac{2^k+2^{k-1}+\cdots+1}{2^k}\right)=\frac{q}{2^k}\sum_{i=0}^{k}2^i=\frac{q}{2^k}(2^{k+1}-1)$$

The loop keeps on until two inequalities remain only. Then $(q/2^k)=2 \Rightarrow q=2^{k+1}$, and hence $(q/2^k)/(2^{k+1}-1)=2(q-1)<2q\in O(q)$. ■

Megiddo (1982) and Dyer (1984) proposed $O(q)$ algorithms for calculating, respectively, the minimal and maximal values of a two-variable linear programming problem. However, the time complexity of their methods is bounded by $4q$, whereas the new approach is bounded by $2q$.

We end this section presenting the segment comparison algorithm. In the next section, we show an example that helps to clarify the ideas of the algorithms.

```
function SegmentComparison(SegmentSet S)
{    S_ND := S
   for all segments X := [x_0, x_1] ∈ S_ND do
      for all segments Y := [y_0, y_1] ∈ S_ND successors in S_ND to X do
         {  for r := 1 to q do
               {  Create inequality y(x)
                  T := T ∪ y(x)
               }
            Dominate(T, X, Y)
            if X ≺ Y then   Y := Y / [y_min, y_max]
            Change inequalities y(x) to x(y) defining the complementary region R̄
            Dominate(T, Y, X)
            if Y ≺ X then   X := X / [x_min, x_max]
         }
   return S_ND
}
```

Algorithm III.7: The segment comparison function.

III.5 An example to illustrate the algorithms

In this section we present an example applying the algorithms proposed in previous sections. We consider the network given in Figure III.13, which consists of 9 vertices and 16 edges. We have assigned 4 lengths on each edge, and so we have 4 objective functions on the network. The distance matrices between vertices for each objective are shown in Table III.3 and Table III.4.

The following 1-medians are obtained for the different objectives:

- Vertices v_2 and v_3, for the first objective, with value 29 for both of them.
- Vertex v_2 for the second objective, with value 35.
- Vertex v_3 for the third objective, with value 29.
- Vertex v_9 for the fourth objective, with value 46.

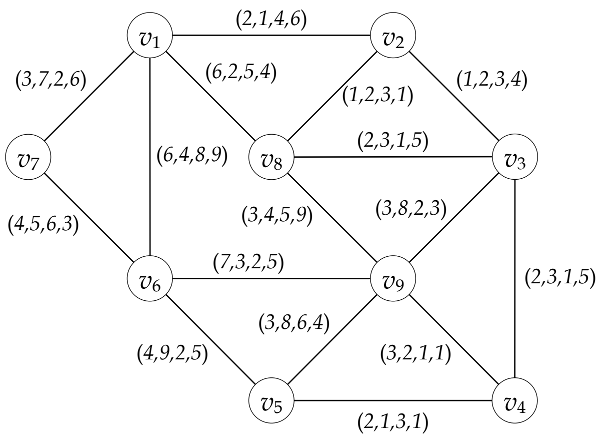


Figure III.13: A network with 9 vertices, 16 edges and 4 lengths per edge.

	Objective 1										*Objective 2*									
	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8	v_9	Sum	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8	v_9	Sum
v_1	0	2	3	5	7	6	3	3	6	35	0	1	3	6	7	4	7	2	6	36
v_2	2	0	1	3	5	8	5	1	4	29	1	0	2	5	6	5	8	2	6	35
v_3	3	1	0	2	4	8	6	2	3	29	3	2	0	3	4	7	10	3	5	37
v_4	5	3	2	0	2	6	8	4	3	33	6	5	3	0	1	5	10	6	2	38
v_5	7	5	4	2	0	4	8	6	3	39	7	6	4	1	0	6	11	7	3	45
v_6	6	8	8	6	4	0	4	9	7	52	4	5	7	5	6	0	5	6	3	41
v_7	3	5	6	8	8	4	0	6	9	49	7	8	10	10	11	5	0	9	8	68
v_8	3	1	2	4	6	9	6	0	3	34	2	2	3	6	7	6	9	0	4	39
v_9	6	4	3	3	3	7	9	3	0	38	6	6	5	2	3	3	8	4	0	37

Table III.3: Distance matrices for the first and second objectives
of the network shown in Figure III.13.

	Objective 3										*Objective 4*									
	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8	v_9	Sum	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8	v_9	Sum
v_1	0	4	6	7	10	8	2	5	8	50	0	5	9	13	14	9	6	4	12	72
v_2	4	0	3	4	7	7	6	3	5	39	5	0	4	8	9	12	11	1	7	57
v_3	6	3	0	1	4	4	8	1	2	29	9	4	0	4	5	8	11	5	3	49
v_4	7	4	1	0	3	3	9	2	1	30	13	8	4	0	1	6	9	9	1	51
v_5	10	7	4	3	0	2	8	5	4	43	14	9	5	1	0	5	8	10	2	54
v_6	8	7	4	3	2	0	6	5	2	37	9	12	8	6	5	0	3	13	5	61
v_7	2	6	8	9	8	6	0	7	8	54	6	11	11	9	8	3	0	10	8	66
v_8	5	3	1	2	5	5	7	0	3	31	4	1	5	9	10	13	10	0	8	60
v_9	8	5	2	1	4	2	8	3	0	33	12	7	3	1	2	5	8	8	0	46

Table III.4: Distance matrices for the third and fourth objectives
of the network shown in Figure III.13.

Therefore, we have obtained the following vectors which are 1-median vertices:

- Vertex v_2 with value vector $(29,35,39,57)$.
- Vertex v_3 with value vector $(29,37,29,49)$.

- Vertex v_9 with value vector $(38,37,33,46)$.

The rest of the vertices are dominated by the 1-median vertices. For example, we have:

- Vertex v_1, with value $(35,36,50,72)$, is dominated by vertex v_2.
- Vertex v_4, with value $(33,38,30,51)$, is dominated by vertex v_3.
- Vertex v_5, with value $(39,45,43,54)$, is dominated by vertices v_3 and v_9.
- Vertex v_6, with value $(52,41,37,61)$, is dominated by vertices v_3 and v_9.
- Vertex v_7, with value $(49,68,54,66)$, is dominated by vertices v_2, v_3 and v_9.
- Vertex v_8, with value $(34,39,31,60)$, is dominated by vertex v_3.

We have applied the rule for removing non-efficient edges, obtaining the results shown in Table III.5.

Edge	*Removal process*
(v_1,v_2)	Dominated by vertex v_2
(v_1,v_6)	**Not removed**
(v_1,v_7)	Dominated by vertex v_2
(v_1,v_8)	**Not removed**
(v_2,v_3)	**Not removed**
(v_2,v_8)	**Not removed**
(v_3,v_4)	Dominated by vertex v_3
(v_3,v_8)	Dominated by vertex v_3
(v_3,v_9)	**Not removed**
(v_4,v_5)	Dominated by vertex v_3
(v_4,v_9)	**Not removed**
(v_5,v_6)	Dominated by vertex v_3 and v_9
(v_5,v_9)	Dominated by vertex v_9
(v_6,v_7)	Dominated by vertex v_3 and v_9
(v_6,v_9)	Dominated by vertex v_9
(v_8,v_9)	**Not removed**

Table III.5: Removal process results of the network shown in Figure III.13.

Therefore, the partial subgraph that will be necessary to analyze is shown in Figure III.14.

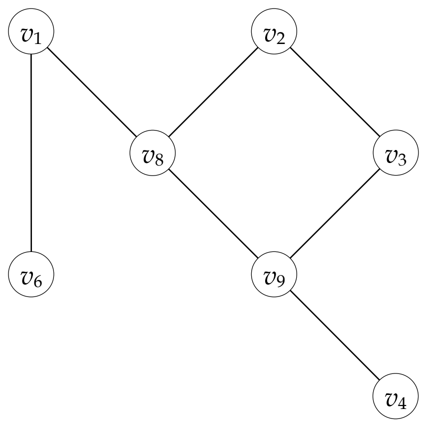


Figure III.14: Network to be analyzed after applying the edge removal process.

It may be seen that seven edges have not been removed, and on these edges there must be efficient location points. Now we can apply the procedure. The approach starts by calculating the vertex distance matrices for the four objectives, which are shown in Table III.3 and Table III.4. Following Algorithm III.1, the breakpoints are calculated, and the respective polygonal lines of the objective functions on each edge can be drawn. For example, the four objective functions

$$f^r(x) = \sum_{v_i \in V} d^r(x, v_i), \quad \text{with } r = 1, \ldots, 4 \quad \text{and } x \in (v_3, v_9)$$

are shown together in Figure III.15.

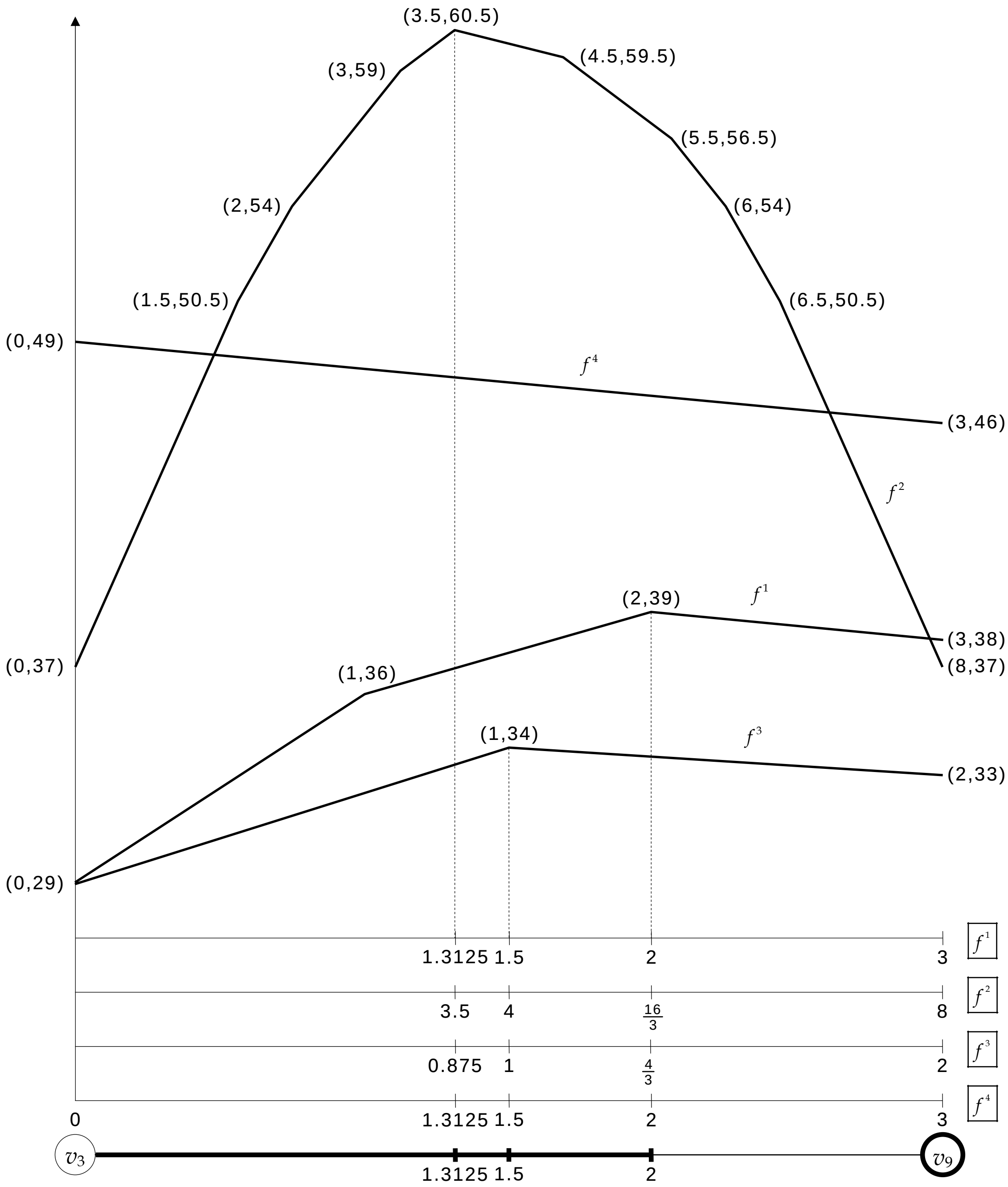


Figure III.15: Objective function obtained on edge (v_3, v_9).

The procedure continues with Algorithm III.2, which uses the maximum points of the objective functions to split the edges into segments. For each segment, we will leave a single point in its place if all the objective functions are decreasing or increasing, since that point is better than any point inside that segment. Otherwise, we leave the whole segment. So, for the edge (v_3, v_9) we have three segments and one point labeled at the bottom of Figure III.15. Using the maximum points of each function, the edge is split into four intervals: $[0,1.3125]$, $[1.3125,1.5]$, $[1.5,2]$ and $[2,3]$, with respect to f^1. Only three intervals ($[0,1.3125]$, $[1.3125,1.5]$ and $[1.5,2]$) are included in the set of segments *S*, and one point (vertex v_9) is included in the set of points *P*. At the bottom of Figure III.15, bold lines represent the intervals and the vertex which remain.

The segments and points are calculated for all the edges not removed, as shown in Table III.6. Next, we compare points and segments among them, and points against segments, to determine efficient points.

Edge	*Segments*	*Points*
(v_1, v_6)	$[2,2.25]$, $[2.25,3.5]$	Vertices v_1 and v_6
(v_1, v_8)	$[0,2.5]$, $[2.5,3]$	Vertex v_8
(v_2, v_3)	$[0,0.5]$, $[0.5,0.75]$	Vertex v_3
(v_2, v_8)	$[\frac{16}{3},0.75]$, $[0.75,1]$	Vertex v_2
(v_3, v_9)	$[0,1.3125]$, $[1.3125,1.5]$, $[1.5,2]$	Vertex v_9
(v_4, v_9)	$[0,1.5]$, $[1.5,2]$, $[2,3]$	–
(v_8, v_9)	$[1.125,\frac{7}{6}]$, $[\frac{7}{6},1.8]$, $[1.8,2]$	Vertices v_8 and v_9

Table III.6: For each edge not removed, we show all the segments and points obtained.

The final solution obtained is shown in Table III.7. The values *x*, *y* and *z* represent the following points:

- *x* is the point on the edge (v_2, v_3) located at distance 0.5 from vertex v_2, with respect to the first objective.
- *y* is the point on the edge (v_3, v_9) located at distance 5/3 from vertex v_3, with respect to the first objective.
- *z* is the point on the edge (v_4, v_9) located at distance 1.24576 from vertex v_4, with respect to the first objective.

Finally, the efficient points obtained are marked on the network in Figure III.16.

Edge	*Efficient points*
(v_2, v_3)	$[2,x]$
(v_3, v_9)	$[3,y]$
(v_4, v_9)	$[z,9]$

Table III.7: Efficient points are only located on edges (v_2, v_3), (v_3, v_9) and (v_4, v_9). With respect to the first objective, *x* is located at 0.5 from v_2, *y* at $\frac{5}{3}$ from v_3 and *z* at 1.24576 from v_4.

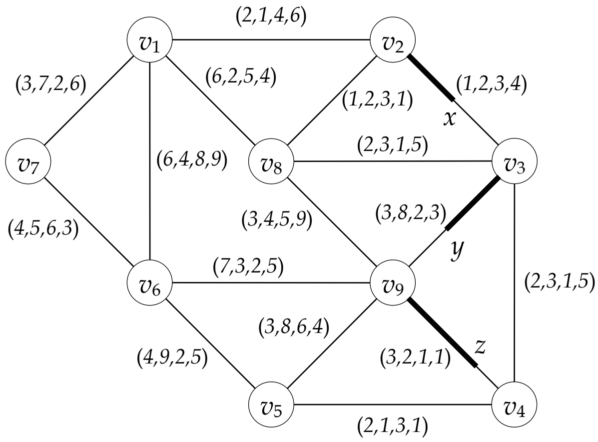


Figure III.16: Efficient location points are shown on the network with bold lines.

III.6 Conclusions

In this chapter, the problem of locating a facility on a network with multiple median-type objectives consisting of minimizing the sum of the distances or lengths from the location point to the vertices of the network has been studied. Although this problem, known as 1-median, is easy for the single objective case (Hakimi, 1964), its extension to the multiobjective case is not.

We have proved in this chapter that the efficient points need not be only at the vertices of the network, nor on the shortest paths linking median vertices corresponding to each median-type objective. Therefore, the search of efficient points is not restricted to vertices or to a specific part of the network, but rather it should be extended to all the edges of the network. To simplify this search, we proposed a simple rule to remove edges of the network which will never contain efficient points. In order to determine efficient location points we have presented a method which consists of two algorithms. The first calculates for each edge the breakpoints where the slope changes. The objective functions are obtained by using these breakpoints. The second splits each edge in several segments considering the maximum points of the objective functions. These segments are then compared to obtain the efficient location points.

Chapter IV

Extending the multiobjective network location framework to the cent-dian problem

"Rationally speaking, there are no criteria for the selection of criteria"
J. KRARUP & P.M. PRUZAN

IV.1 Introduction

In the last chapter we analyzed a network location problem with several median-type objectives, and we proposed a polynomial algorithm to solve it. It would be reasonable now to approach the multiobjective center location problem on networks. However, we consider more remarkable to analyze the λ-cent-dian problem on networks with not only several lengths on the edges, but also several weights on the nodes. Thus, as shown in Chapter II, for $\lambda = 0$ we can solve the median problem, whereas for $\lambda = 1$, we get the solution to the center problem.

As stated in Chapter I, the center location problem was proposed and solved by Hakimi (1964). This problem concerns *equity* issues, and it is used to locate emergency services such as fire, police, ambulance services, rescue depots, etc.

On the other hand, if we wish to minimize the total (aggregate or average weighted) distance, then we pose the median problem (Hakimi, 1964). The median addresses *spatial efficiency* and it is suitable for locating facilities that involve the distribution of persons or goods, i.e. schools, shopping centers, mail service, etc.

However, since the median is based on averaging, it can discriminate remote and low-population density areas against centrally situated and high-population density areas, which implies no equity (Hansen, Labbé and Thisse, 1991; Ogryczak, 1997).

On the other hand, the location of a facility at the center may cause a large increase in the total distance, which means no spatial efficiency (Hansen, Labbé and Thisse, 1991; Ogryczak, 1997). Halpern (1976) introduced the λ-cent-dian as a compromise between the center and the median, by means of a convex combination. This model allows exploiting jointly the main advantages of each previous problem.

We have already remarked that most of the huge literature on network location analysis deals with one criterion only on each node (weight) and/or one criterion on each edge (length). Nevertheless, there are many applications in which several criteria need to be considered. For example, several weights may represent demand, social and politic importance, number of potential complementary services, etc. Likewise, several costs (lengths) might stand for distance, time, traffic congestion, toll, etc.

Following the work done in Chapters II and III, we analyze the λ-cent-dian problem on a network, considering several weights on the nodes and several lengths on the edges. According to the location analysis classification scheme presented in Chapter I, this problem is defined as $1/\mathcal{G}/\bullet/d(\mathcal{V},\mathcal{G})/Q-\mathrm{CD}_{\mathrm{par}}$.

The remains of this chapter are structured as follows. Next, we introduce the notation and the basic definitions. Then, making use of the algorithms that solve the multiobjective median problem, we propose a polynomial algorithm to solve the multiobjective λ-cent-dian problem. The chapter ends with a brief example, the computational results and the conclusions.

IV.2 Definitions and model formulation

Let $N=(V,E)$ be a simple (no loops or multiple edges), connected and undirected network, with $V=\{v_1,v_2,\ldots,v_n\}$ being the set of nodes, and $E=\{(v_s,v_t):v_s,v_t\in V\}$ being the set of edges. Let p be the number of weights placed on each node, and q the number of lengths (costs) on each edge. Thus, for each node in V, we define the following weight function

$$\begin{array}{llll} w: & V & \longrightarrow & \mathbb{R}^p \\ & v_i\in V & \longrightarrow & w(v_i)=w_i=(w_i^1,\ldots,w_i^p) \end{array}$$

Likewise, over each edge in E we define the next length function

$$\begin{array}{llll} l: & E & \longrightarrow & \mathbb{R}^q \\ & e=(v_s,v_t)\in E & \longrightarrow & l(e)=l_e=(l_e^1,\ldots,l_e^q) \end{array}$$

Let r be a length index, with $1\le r\le q$, and $x\in e=(v_s,v_t)$ an inner point. We define $c_e^r(x,v_s)$ as the length of the line segment between x and v_s regarding length r, with $0\le c_e^r(x,v_s)\le l_e^r$ and $c_e^r(x,v_t)=l_e^r-c_e^r(x,v_s)$.

For any pair of nodes v_a and v_b, the distance between such nodes, denoted by $d^r(v_a,v_b)$, is defined as the length of any shortest path in N joining v_a and v_b concerning length r. In the same way, given any point $x\in N$ and any node $v_i\in V$, let

$$d^r(x,v_i)=\min\{c_e^r(x,v_s)+d(v_s,v_i),c_e^r(x,v_t)+d(v_t,v_i)\}$$

be the distance between point x and node v_i considering length r.

As we did in Chapter II, we now define the unweighted center function (Hansen, Labbé and Thisse, 1991) as

$$f_{\max}^r(x)=\max_{v_i\in V} d^r(x,v_i),\quad \forall x\in N,\ r=1,\ldots,q$$

and a point $x_c\in N$ is an (absolute) *center* for length r if $f_{\max}^r(x_c)=\min_{x\in N} f_{\max}^r(x)$.

On the other hand, the median function (Hansen, Labbé and Thisse, 1991) is defined as

$$f_{\mathrm{sum}}^{sr}(x)=\frac{1}{W^s}\sum_{v_i\in V} w_i^s d^r(x,v_i),\quad \forall x\in N,\ s=1,\ldots,p,\ r=1,\ldots,q$$

where $W^s = \sum_{v_i \in V} w_i^s$ represents the sum of weights for a certain weight index *s*. A point $x_m \in N$ is a *median* for a given weight index *s* and a certain length index *r* when $f_{\text{sum}}^{sr}(x_m) = \min_{x \in N} f_{\text{sum}}^{sr}(x)$.

Finally, the λ-cent-dian function arises from the convex combination of these two latter functions, that is

$$F_{\text{cd}}^{sr}(\lambda, x) = \lambda \max_{v_i \in V} d^r(x, v_i) + \frac{(1-\lambda)}{W} \sum_{v_i \in V} w_i^s d^r(x, v_i) = \lambda f_{\max}^r(x) + (1-\lambda) f_{\text{sum}}^{sr}(x)$$
$$\forall x \in N, \quad 0 \le \lambda \le 1, \quad s = 1, \ldots, p \quad r = 1, \ldots, q$$

The properties of the λ-cent-dian function were stated and commented on in Chapter II.

Let $F(\lambda, x) = (F_{\text{cd}}^{11}(\lambda, x), F_{\text{cd}}^{12}(\lambda, x), \ldots, F_{\text{cd}}^{pq}(\lambda, x)) \in \mathbb{R}^{p \times q}$. For a given value of λ, $0 \le \lambda \le 1$, the problem consists of finding the set $x_{\text{cd}} \in N$ such that

$$F(\lambda, x_{\text{cd}}) = \min_{x \in N} F(\lambda, x)$$

Let $k = p \times q$, and let $g = (g^1, g^2, \ldots, g^k)$ and $h = (h^1, h^2, \ldots, h^k)$ be two vectors in $\mathbb{R}^k$. Vector *g* is said to dominate vector *h*, denoted as $g \prec h$, iff $g^i \le h^i, \forall i$ and $g^i < h^i$ for at least one *i*. Let $U = \{(F^1(\lambda, x), F^2(\lambda, x), \ldots, F^k(\lambda, x)) : \forall x \in N\}$ be the set of all possible vector values on *N*. A vector $F \in U$ is *non-dominated* or *efficient* if $\nexists G \in U$ such that $G \prec F$. The set of all non-dominated vectors is denoted by U_{ND}.

Hence, let $L = \{x \in N : (F^1(\lambda, x), \ldots, F^k(\lambda, x)) \in U_{\text{ND}}\}$. A point $x \in L$ is called *non-dominated* or *efficient*. Our goal is to find out the set U_{ND} and thus, the set of efficient location points *L* on *N*. The next section presents the algorithm that determines the set *L*.

IV.3 The algorithm

Taking into account the approach to the multiobjective median problem, we now present the algorithm which solves the multicriteria λ-cent-dian problem.

As stated in Chapter II, for a given edge $e \in E$ and for all inner points $x \in e$, the λ-cent-dian function $F_{\text{cd}}^{sr}(\lambda, x)$, with $1 \le s \le p$ and $1 \le r \le q$, is neither convex nor concave. Due to this, we must split the $p \times q$ λ-cent-dian functions according to their breakpoints. Subsequently, the algorithm proceeds in a very similar manner to the multiobjective median procedure sketched in Algorithm III.2. The *MulticriteriaCentDian* function is outlined in Algorithm IV.1.

An important difference between this algorithm and the multiobjective median method lies in the splitting into segments and points of the $k = p \times q$ λ-cent-dian functions. This process is performed in at most $O(kn)$ steps, since there might be at most $O(n)$ breakpoints on each of the *k* functions. Therefore, the number of segments and points generated for all the edges is $O(mnk)$. Comparing pairwise all these elements takes $O(m^2n^2k^2)$ steps, and each comparison step takes $O(k)$ time. Thus, provided that all the *k* distance matrices are given, the multicriteria λ-cent-dian algorithm runs in $O(m^2n^2k^3)$.

Before the computational experience is presented, next we present a small example to illustrate how the algorithm performs on a multicriteria network.

function *MulticriteriaCentDian*(Network $N(V,E)$, DistanceMatrix *d*, Parameters *p*, *q*, λ)
{ *// Let P be the set of candidate points to be non-dominated.*
$P := \emptyset$
// Let S be the set of possible non-dominated segments.
$S := \emptyset$
for all edges $e := (v_s, v_t) \in E$ **do**
{ **for** $r := 1$ to q **do**
Compute $f^r_{\max}(x)$
for $s := 1$ to p **do**
for $r := 1$ to q **do**
Compute $f^{sr}_{\text{sum}}(x)$
for $s := 1$ to p **do**
for $r := 1$ to q **do**
Compute $F^{sr}_{\text{cd}}(\lambda, x) = \lambda f^r_{\max}(x) + (1-\lambda) f^{sr}_{\text{sum}}(x)$
Let $b_1, b_2, \ldots, b_j$ be the breakpoints of all the $k = p \times q$ λ-cent-dian functions
Sort these points in increasing order with respect to the first length
Let $v_s = x_0, x_1, \ldots, x_j, x_{j+1} = v_t$ be the sorted sequence of different points including the endnodes v_s and v_t
for $i := 0$ to j **do**
{ Let $[x_i, x_{i+1}]$ be a segment of edge *e*
if $F^{sr}_{\text{cd}}(\lambda, x_i) \le F^{sr}_{\text{cd}}(\lambda, x_{i+1})$, $\forall s = 1, \ldots, p \wedge \forall r = 1, \ldots, q$ **then** $P := P \cup \{x_i\}$
else if $F^{sr}_{\text{cd}}(\lambda, x_i) \ge F^{sr}_{\text{cd}}(\lambda, x_{i+1})$, $\forall s = 1, \ldots, p \wedge \forall r = 1, \ldots, q$ **then** $P := P \cup \{x_{i+1}\}$
else $P := P \cup \{x_i\} \cup \{x_{i+1}\}$ and $S := S \cup \{[x_i, x_{i+1}]\}$
}
}
Compare the points in *P* using Algorithm III.3 and store in set P_{ND} the non-dominated points obtained
Compare the segments in *S* using Algorithm III.7 and store in set S_{ND} the non-dominated segments obtained
Compare the points of P_{ND} with segments in S_{ND} using Algorithm III.4, storing what is non-dominated
return P_{ND} and S_{ND}
}

Algorithm IV.1: The multicriteria λ-cent-dian function.

IV.4 A brief example

A planar network with $n = 5$ nodes and $m = 9$ edges was randomly generated. On each node we place two weights, whereas each edge has associated two lengths. The network is drawn in Figure IV.1. We set parameter λ to 0.5.

Following the guidelines of Algorithm IV.1, first we compute the q unweighted center functions. Figure IV.2 (left) shows the unweighted center function for the first length of edge (v_1, v_3), which corresponds to the upper envelope of all weighted distance functions. Figure IV.2 (right) shows the two unweighted center functions related to the two lengths of this edge.

Next, we obtain the $p \times q$ weighted median functions. Figure IV.3 (left) shows the weighted median function with regards to the first weight and the first length of edge (v_1, v_3), whereas the right figure depicts the four weighted median functions of that edge. Note that there is one weighted median function generated for each combination of weights and lengths.

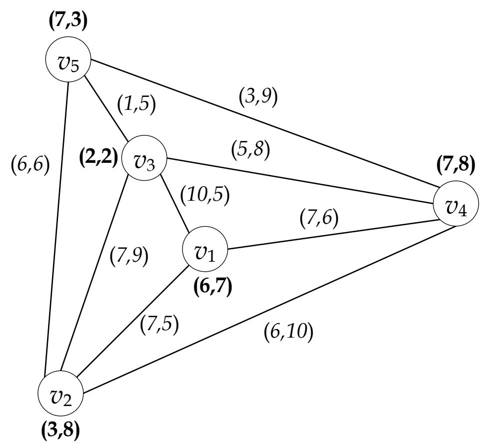


Figure IV.1: A network with two lengths per edge and two weights per node.

Once all the center and median functions have been obtained, we proceed to build up the λ-cent-dian functions from the convex combination of these two latter functions. Given $\lambda = 0.5$, Figure IV.4 (left) shows the λ-cent-dian for the first weight and the first length of edge (v_1, v_3). The right figure shows the four λ-cent-dian obtained for each combination of weights and lengths.

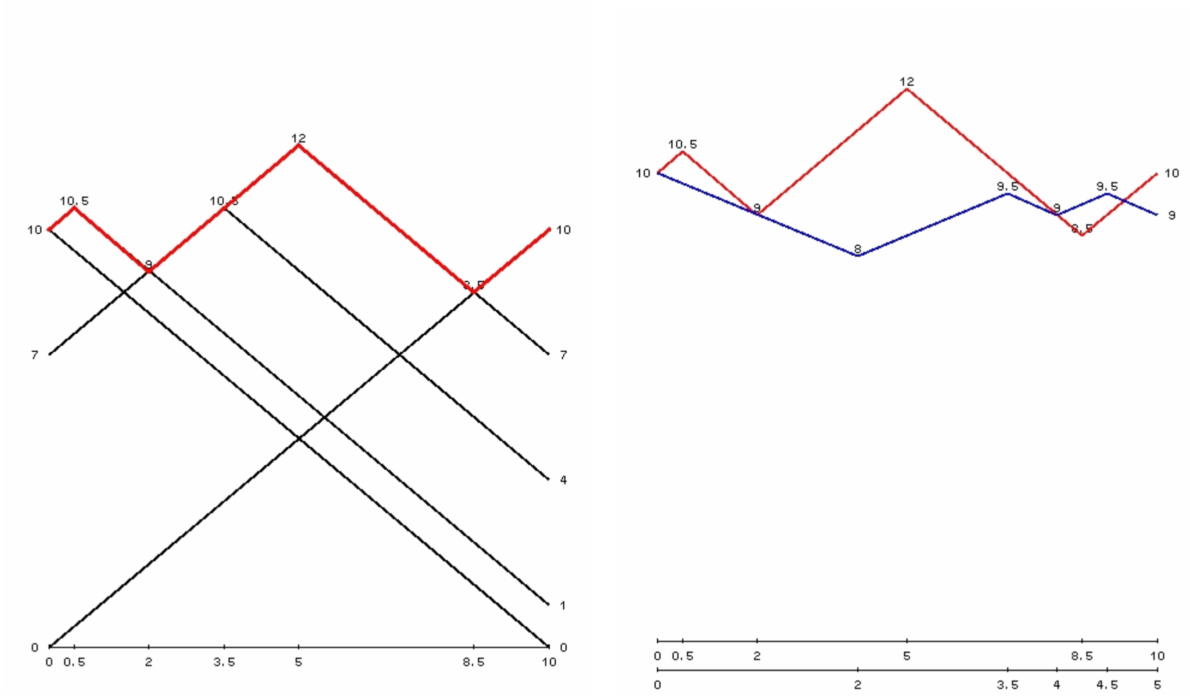


Figure IV.2: Unweighted center function for the first length (left) and the two unweighted center functions on edge (v_1, v_3) (right).

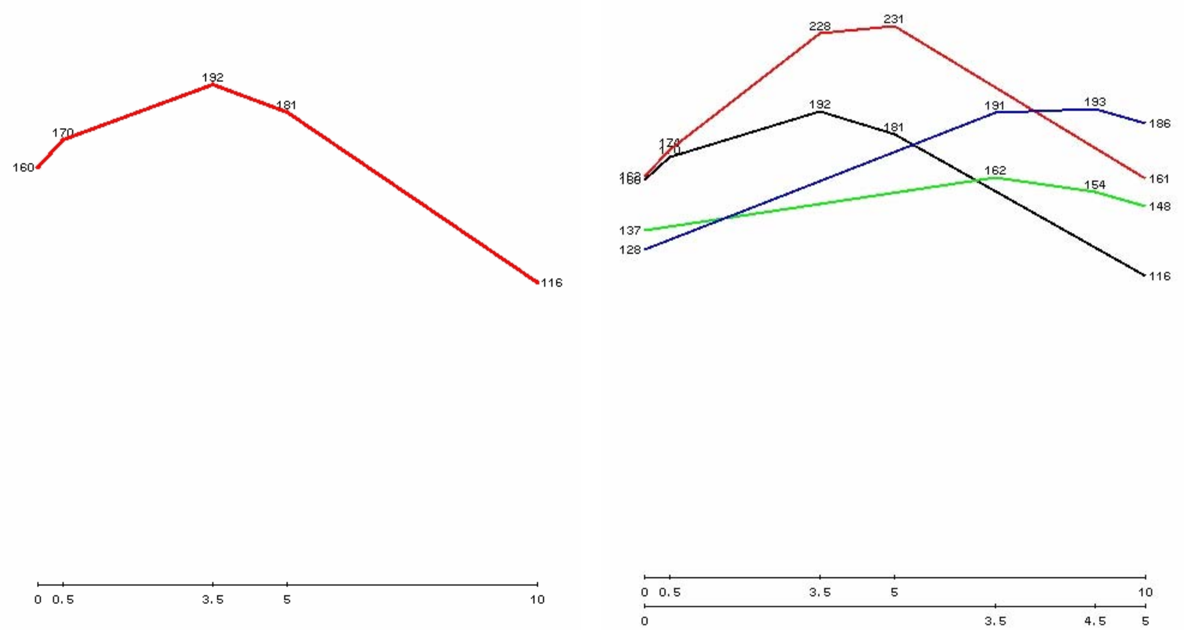


Figure IV.3: Weighted median function for the first length (left)
and the four weighted median functions on edge (v_1, v_3) (right).

Next, we split these λ-cent-dian functions to obtain the set of points P and the set of segments S. Henceforth, the segments in S and the points in P are compared pairwise. Figure IV.5 (left) illustrates a comparison between two segments of set S, whereas the right figure represents the comparison between a point of set P and a segment of set S.

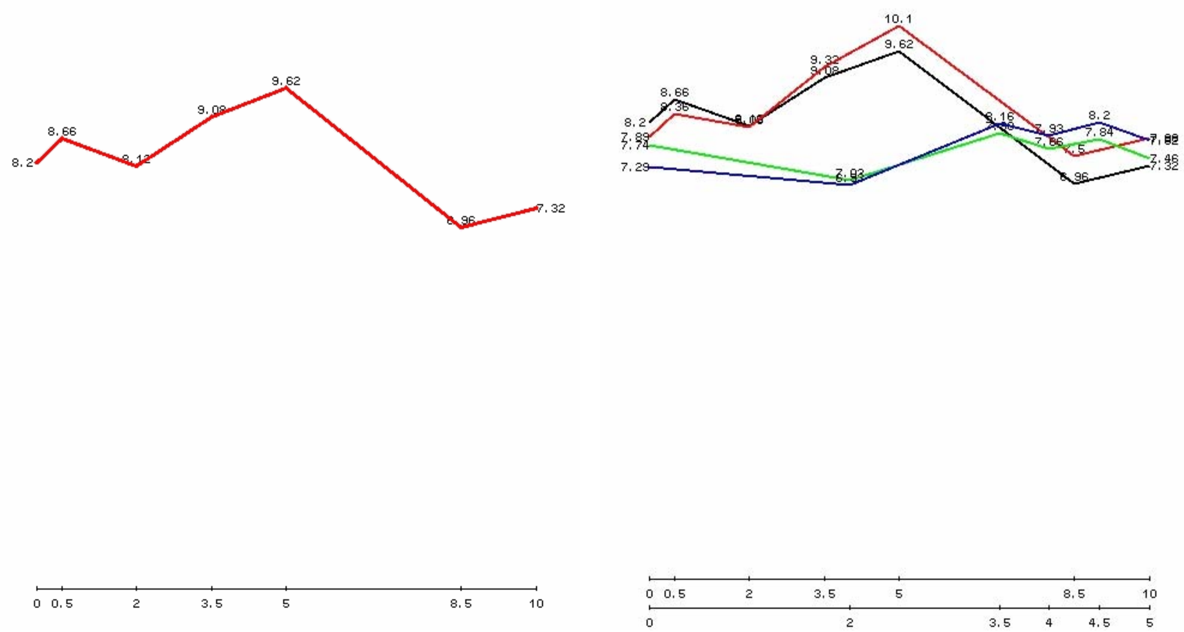


Figure IV.4: λ-cent-dian function for the first length (left)
and the four λ-cent-dian functions on edge (v_1, v_3) (right) with $\lambda = 0.5$.

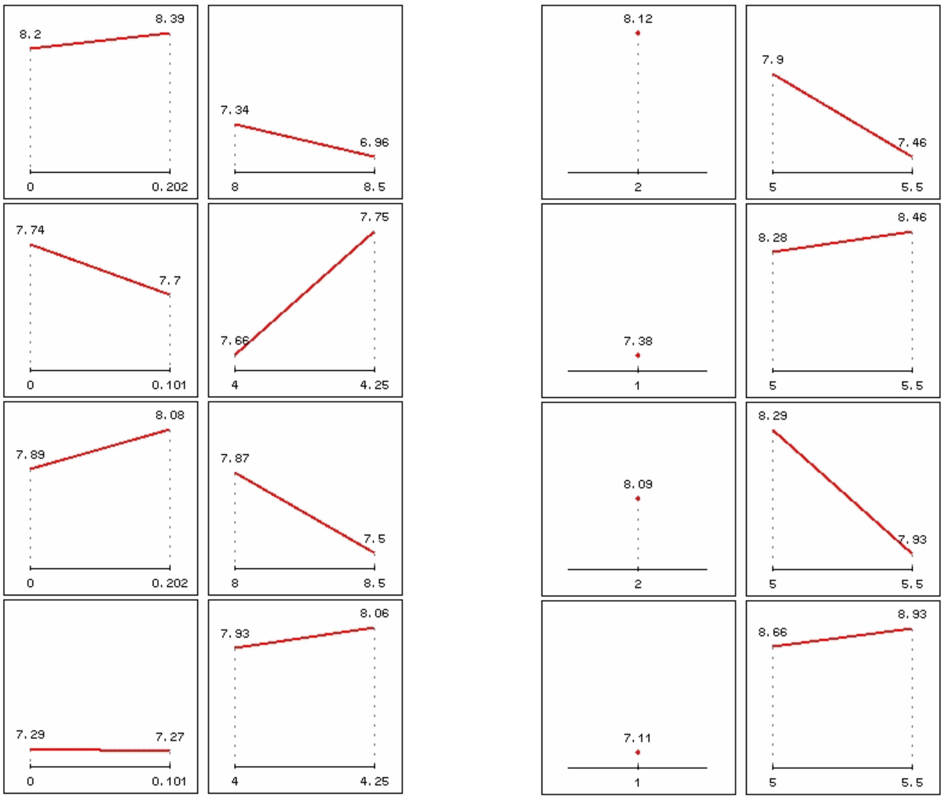


Figure IV.5: A segment comparison (left) and a point-segment comparison (right).

Edge	*Efficient points*
(v_1, v_3)	[0,0.202387], [2,4], [8.12501,8.5], [9.36646,10]
(v_1, v_4)	[6.5,7]
(v_2, v_1)	[1.89413,3.5]
(v_2, v_3)	[6.61111,7]
(v_2, v_4)	[0,0.3], [5.7,6]

Table IV.1: Efficient location points of the network of Figure IV.1.

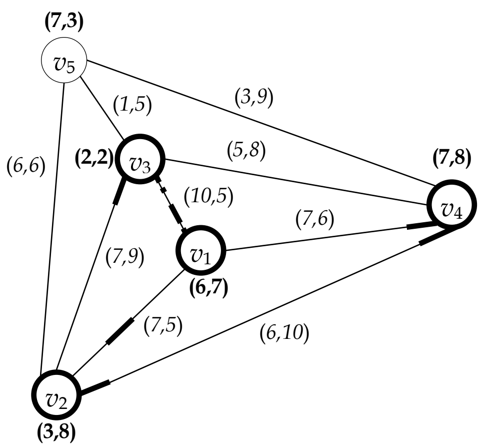


Figure IV.6: Efficient points are drawn in bold on the network.

Finally, the solution obtained is presented in Table IV.1. Likewise, these efficient points are also drawn on the original network in Figure IV.6. Note that the efficient location points are not only sited on some nodes of the network, but also they might be placed on any point of the edges.

Before stating the conclusions of this chapter, in the next section we present the computational results of the multicriteria λ-cent-dian algorithm.

IV.5 Computational results

The computational experiment was performed on a DEC with four Alpha 466 MHz processors and 2 Gb of RAM, running OSF Digital UNIX. The algorithm was programmed using GNU g++ 2.95.2 and LEDA 4.2.1 (*Library of Efficient Datatypes and Algorithms*).

Random planar networks ($m = 3n - 6$) were generated with $n = 10$ up to 100 nodes. Both the number of node weights p and the number of edge lengths q range from 1 to 3. The weight values vary uniformly between 1 and 10, whereas the length values are uniformly distributed from 1 to 50. Parameter λ ranges between 0 and 1 with a step of 0.25. Ten random instances were generated for each problem.

Table IV.2 shows the computing times obtained for each combination of n, λ, p and q. Given a fixed combination of n, p and q, note that the computing times remain almost the same independently of the value of parameter λ. Besides, the cases $\lambda = 0$ and $\lambda = 1$ correspond to the multicriteria median problem and to the multicriteria center problem, respectively.

Figure IV.7 shows the time graphics for $\lambda = 0$, 0.5 and 1. Obviously, the running times proportionally grow with respect to both the number of weights p and the number of lengths q.

IV.6 Conclusions

Following the model presented in Chapter II, and taking into account the approach to the multiobjective median problem proposed in Chapter III, we have developed a polynomial algorithm that solves the multicriteria λ-cent-dian problem for a given value of λ.

This model allows the solution to the multicriteria unweighted center problem to be obtained for the case of $\lambda = 1$. However, the model can be slightly changed to fit the multicriteria weighted center problem. On the other hand, when $\lambda = 0$, the multicriteria weighted median problem is solved, which is a generalization of the model presented in the last chapter.

In the subsequent chapters, we address several models for the location of undesirable facilities with regards to a single criterion as well as multiple criteria.

		$\lambda = 0$								
		$p = 1$			$p = 2$			$p = 3$		
n	m	$q = 1$	$q = 2$	$q = 3$	$q = 1$	$q = 2$	$q = 3$	$q = 1$	$q = 2$	$q = 3$
10	24	0.07	0.09	0.16	0.08	0.11	0.19	0.08	0.13	0.23
20	54	0.14	0.20	0.31	0.18	0.27	0.45	0.21	0.35	0.63
30	84	0.29	0.39	0.60	0.36	0.54	0.83	0.42	0.69	1.12
40	114	0.38	0.53	0.79	0.48	0.77	1.25	0.58	1.00	1.58
50	144	0.62	0.85	1.27	0.77	1.20	1.88	0.92	1.53	2.53
60	174	0.71	1.01	1.56	0.93	1.53	2.25	1.14	1.96	3.15
70	204	0.83	1.23	1.86	1.09	1.84	2.89	1.36	2.48	3.96
80	234	0.94	1.46	2.23	1.32	2.27	3.73	1.68	3.08	4.89
90	264	1.40	2.04	3.02	1.86	3.09	4.63	2.42	4.21	6.59
100	294	1.57	2.35	3.40	2.14	3.58	5.55	2.71	4.77	7.36

		$\lambda = 0.25$								
		$p = 1$			$p = 2$			$p = 3$		
n	m	$q = 1$	$q = 2$	$q = 3$	$q = 1$	$q = 2$	$q = 3$	$q = 1$	$q = 2$	$q = 3$
10	24	0.06	0.08	0.14	0.08	0.10	0.19	0.07	0.12	0.25
20	54	0.15	0.22	0.37	0.18	0.28	0.48	0.21	0.34	0.64
30	84	0.31	0.40	0.61	0.37	0.55	0.85	0.42	0.66	1.10
40	114	0.38	0.54	0.82	0.48	0.76	1.31	0.58	1.03	1.67
50	144	0.62	0.88	1.25	0.78	1.20	1.92	0.91	1.60	2.40
60	174	0.70	1.03	1.63	0.92	1.54	2.36	1.15	1.99	3.25
70	204	0.81	1.24	1.89	1.11	1.89	3.05	1.39	2.55	4.22
80	234	0.94	1.47	2.29	1.29	2.31	3.59	1.69	3.08	5.37
90	264	1.42	2.15	3.03	1.92	3.10	4.69	2.34	4.15	6.40
100	294	1.54	2.38	3.36	2.17	3.67	5.46	2.75	4.93	7.80

		$\lambda = 0.5$								
		$p = 1$			$p = 2$			$p = 3$		
n	m	$q = 1$	$q = 2$	$q = 3$	$q = 1$	$q = 2$	$q = 3$	$q = 1$	$q = 2$	$q = 3$
10	24	0.05	0.09	0.14	0.09	0.12	0.17	0.07	0.11	0.22
20	54	0.16	0.22	0.33	0.18	0.29	0.48	0.21	0.34	0.61
30	84	0.30	0.42	0.64	0.37	0.53	0.95	0.41	0.68	1.26
40	114	0.38	0.57	0.85	0.48	0.78	1.22	0.57	0.97	1.78
50	144	0.61	0.89	1.37	0.75	1.19	1.95	0.92	1.52	2.55
60	174	0.73	1.05	1.62	0.93	1.52	2.43	1.14	1.98	3.33
70	204	0.82	1.25	1.88	1.11	1.94	3.23	1.38	2.43	4.05
80	234	0.94	1.46	2.36	1.32	2.30	3.69	1.69	3.04	5.19
90	264	1.45	2.07	3.18	1.94	3.07	4.80	2.34	4.13	6.48
100	294	1.57	2.36	3.78	2.16	3.63	5.47	2.77	4.90	7.55

		$\lambda = 0.75$								
		$p = 1$			$p = 2$			$p = 3$		
n	m	$q = 1$	$q = 2$	$q = 3$	$q = 1$	$q = 2$	$q = 3$	$q = 1$	$q = 2$	$q = 3$
10	24	0.07	0.09	0.15	0.07	0.10	0.17	0.07	0.12	0.22
20	54	0.15	0.21	0.36	0.18	0.29	0.48	0.21	0.33	0.61
30	84	0.31	0.40	0.78	0.36	0.55	0.93	0.41	0.66	1.16
40	114	0.38	0.56	1.03	0.48	0.74	1.44	0.58	1.00	1.76
50	144	0.63	0.87	1.36	0.75	1.18	2.09	0.92	1.57	2.59
60	174	0.72	1.06	1.76	0.93	1.56	2.40	1.16	2.00	3.31
70	204	0.84	1.25	2.00	1.10	1.84	2.91	1.39	2.53	4.31
80	234	0.95	1.49	2.50	1.30	2.28	3.90	1.66	3.10	5.10
90	264	1.38	2.06	3.24	1.91	3.16	4.79	2.37	4.22	6.49
100	294	1.58	2.35	3.77	2.16	3.67	5.81	2.74	4.81	7.75

		$\lambda = 1$								
		$p = 1$			$p = 2$			$p = 3$		
n	m	$q = 1$	$q = 2$	$q = 3$	$q = 1$	$q = 2$	$q = 3$	$q = 1$	$q = 2$	$q = 3$
10	24	0.06	0.08	0.11	0.07	0.10	0.15	0.08	0.11	0.16
20	54	0.15	0.18	0.25	0.18	0.24	0.35	0.20	0.28	0.47
30	84	0.30	0.37	0.49	0.37	0.48	0.68	0.41	0.60	0.86
40	114	0.37	0.50	0.67	0.47	0.68	0.94	0.58	0.87	1.29
50	144	0.60	0.80	1.02	0.73	1.08	1.48	0.91	1.38	1.97
60	174	0.71	0.95	1.23	0.92	1.37	1.88	1.13	1.81	2.59
70	204	0.81	1.12	1.48	1.09	1.72	2.40	1.35	2.28	3.28
80	234	0.94	1.34	1.78	1.29	2.10	2.94	1.69	2.84	4.28
90	264	1.44	1.92	2.54	1.90	2.87	3.96	2.34	3.86	5.47
100	294	1.55	2.17	2.86	2.13	3.36	4.79	2.71	4.52	6.45

Table IV.2: Computing time results.

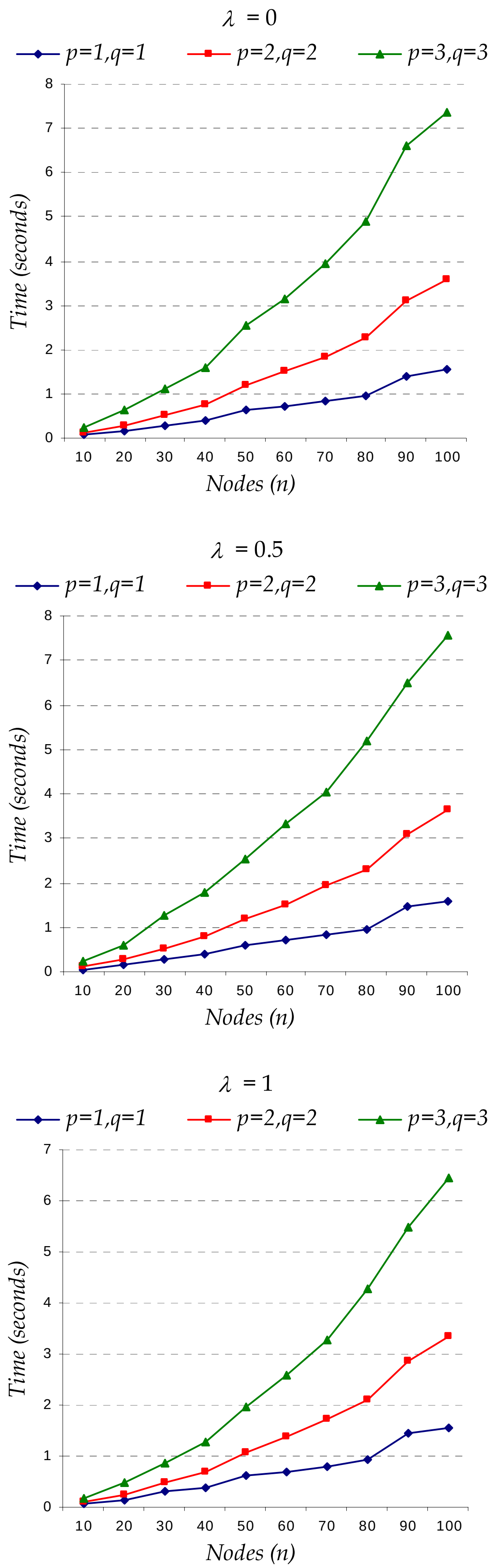


Figure IV.7: Computing time graphics for $\lambda = 0$, 0.5 and 1.

Chapter V

The undesirable center location problem on networks

"Things should be made as simple as possible,
but not any simpler"
A. EINSTEIN

V.1 Introduction

Network location problems deal with finding the right position where one or more facilities should be placed, in order to optimize a certain objective function which is related to the distance from the facility to the demand points (customers). Usually, the facilities to be located are *desirable*, that is, potential customers (nodes) try to attract them as closely as possible. For example, services such as police/fire stations, hospitals, schools or even shopping centers are typical desirable facilities.

Hakimi (1964) introduced the network location analysis, addressing the center problem (minimize the farthest distance) and the median problem (minimize the sum of distances). Later on, several authors have studied thoroughly these problems and they have proposed polynomial algorithms to solve them (see Minieka, 1981; Kariv and Hakimi, 1979a,b).

However, sometimes the facilities can be considered undesirable for the surrounding population, such as nuclear reactors, military installations, polluting plants, prisons, correctional centers and garbage dump sites. Erkut and Neuman (1989) distinguish between *noxious* (harmful, lethal) and *obnoxious* (annoying, unbearable) facilities. For the sake of clearness, we call them *undesirable*.

Even though location theory begins in the 17th century, location problems involving undesirable facilities have only been discussed since the early 1970s. This is due to the fact that undesirable facilities are the consequence of technology and industrialization. In this sense, nuclear reactors, power plants, dump sites and huge airports are all contemporary problems, whereas there have been desirable facilities, such as police stations, hospitals, schools and warehouses, for centuries.

There are not many papers devoted to undesirable location on networks. Church and Garfinkel (1978) studied the *one-facility maximum median* (*maxian*) problem, providing an $O(mn \log n)$ algorithm. This was improved by Tamir (1991), who briefly suggested an $O(mn)$ procedure. Minieka (1983) also proposed the *anticenter* (maxmax) and the *antimedian* (maxsum).

According to Erkut and Neuman (1989) and Cappanera (1999), there was no paper regarding the location of one undesirable center (*maximin*) in the location literature thus far. The first $O(mn)$ algorithm for the 1-maximin problem was briefly suggested by Tamir (1988) using Megiddo (1982) and Dyer (1984). In the particular cases in which the underlying graph is a path, a star or a tree, Burkard, Dollani, Lin and Rote (2001) have developed algorithms which improve those given by Tamir (1988). Lately, Melachrinoudis and Zhang (1999) have proposed another $O(mn)$ procedure based on upper bounds and on a minor modification to Dyer (1984). The most recent paper regarding this problem is written by Berman and Drezner (2000), who gave a linear programming approach in $O(mn)$ time. The algorithm we present computationally improves these former approaches.

The main purpose of this chapter is twofold. First, we tighten the upper bounds already proposed, reducing even more both the number of edges to be processed and, on each edge, the number of operations to get the optimal point. Secondly, we put forward a new algorithm in $O(mn)$ time for the undesirable 1-center on networks. This new approach relies on the intersection of the distance function lines with opposite sign slopes, and avoids the matching of superfluous lines. Even though the theoretical complexity is identical to the approaches formerly reported, the computing times of the new algorithm are normally smaller. This fact becomes quite outstanding when we want to test the problem several times in a sensitivity analysis. Likewise, some harder problems, such as multicriteria network location problems, require computing the solutions for each single criterion to get the set of local non-dominated points.

The rest of the chapter is structured as follows. First, we present the basic notation and the formulation of the undesirable 1-center problem, as well as the analysis of the unweighted case. The next section states new properties for the weighted undesirable 1-center problem. In the following section the latest approaches to this problem are analyzed, along with the new tightened upper bounds. Hence, we demonstrate that by reformulating the maximin problem in an easier way we can greatly improve the computational complexity. Finally, several graphics and tables are presented comparing the new algorithm with the two latest approaches. In the last section, we summarize the chapter.

V.2 Notation and model formulation

Let $N=(V,E)$ be a simple (no loops or multiple edges) undirected and connected network, $V=\{v_1,v_2,\dots,v_n\}$ being the set of nodes, and $E=\{(v_s,v_t):v_s,v_t\in V\}$ the set of edges, with $|E|=m$. On each node v_i, we set a positive weight (demand) w_i as a function $w:V\to\mathbb{R}_+$, $v_i\in V\to w(v_i)=w_i>0$. The lower the node weight, the farther the undesirable facility is located from that node.

Besides, each edge $e=(v_s,v_t)$ is labeled with a positive length (travel cost) l_e. So, we have a length function $l:E\to\mathbb{R}_+$, $e=(v_s,v_t)\in E\to l(e)=l_e>0$. Thus, a point $x\in e$ ranges in the interval $[0,l_e]$.

For each pair of nodes $v_i,v_j\in V$ we define the *distance* between two nodes $d(v_i,v_j)$ as the length of the shortest path between v_i and v_j.

Given any edge $e=(v_s,v_t)\in E$, $v_i\in V$ and an inner point $x\in e$, we define the distance between x and a node v_i as $d(x,v_i)=\min\{x+d(v_s,v_i),l_e-x+d(v_t,v_i)\}$. The point where $d(x,v_i)$ attains its equilibrium (i.e. $x+d(v_s,v_i)=l_e-x+d(v_t,v_i)$) is called a *bottleneck point*:

$$b_i=\frac{d(v_t,v_i)+l_e-d(v_s,v_i)}{2} \tag{V.1}$$

When b_i is located inside e, then $d(x,v_i)$ resembles Figure I.2c. Otherwise, the bottleneck point is located over one of the two ending nodes (see Figure I.2a and Figure I.2b).

Now, we are ready to formulate the undesirable 1-center (maximin) problem on networks. Given any point $x\in N$ we define $f(x)=\min_{v_i\in V} w_i d(x,v_i)$.

Then, the problem consists of calculating

$$\max_{x\in N}\min_{v_i\in V} w_i d(x,v_i)=\max_{x\in N} f(x) \tag{V.2}$$

and a point $x_N\in N$ is an undesirable 1-center point iff $f(x_N)=\max_{x\in N} f(x)$.

This problem is the opposite to the 1-center problem (minimax), so it could be called the *anti-center*. Unfortunately, this term was already coined by Minieka (1983) to define the *maxmax* problem. We instead propose the term 1-*uncenter* (undesirable center) to define the optimal location point. According to the classification scheme presented in section I.4, this problem is denoted as $1/\mathcal{G}/\bullet/d(\mathcal{V},\mathcal{G})/\max_{\text{obnox}}$.

If there is at least one vertex v_i such that $w_i=0$, then $f(x)=0,\forall x\in N$ and obviously any point on network N would be a 1-uncenter. Therefore, we consider only $w_i>0$, $\forall v_i\in V$.

Several interesting properties arise for this problem, all stated and proved in Melachrinoudis and Zhang (1999) and in Berman and Drezner (2000).

Property V.1. *For any edge $e=(v_s,v_t)\in E$, $x\in e$, the objective function $f(x)$, is continuous, piecewise linear and concave in the interval $[0,l_e]$, consisting of at most $2n$ strictly monotonic line segments. The value of the objective function is zero at the ends of the edge* (see Figure V.1).

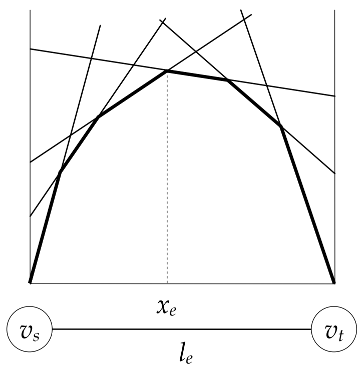


Figure V.1: Objective function $f(x)$, which is actually the lower envelope of all distance functions.

Let x_e be the point in edge $e=(v_s,v_t)\in E$ such that $f(x_e)=\max_{x\in e} f(x)$. This point x_e is called a *local* 1-*uncenter* on edge *e*.

Property V.2. *A unique local* 1*-uncenter* x_e *location exists on each edge e. Consequently, there are at most m* 1*-uncenter locations on a network.*

We now begin discussing in brief the unweighted case for its simplicity, and later we will analyze the weighted 1-uncenter problem.

When all the node weights are equal, $\forall v_i \in V$, $w_i = w$, the local 1-uncenter x_e is sited at the central point of edge *e*. Therefore, the unweighted 1-uncenter x_N is located in the middle of the longest edge(s) (see Melachrinoudis and Zhang, 1999; Berman and Drezner, 2000). This is done in $\mathrm{O}(m)$ time.

V.3 New properties for the weighted 1-uncenter problem

The previous properties allow us to reformulate the 1-uncenter problem over each edge $e = (v_s, v_t) \in E$ as follows: $x_N \in N$ is a 1-uncenter point iff $f(x_N) = \max_{e \in E} f(x_e)$.

Since the local 1-uncenter point is the maximum value of the concave objective function $f(x)$, it should be located at the intersection of two distance functions lines with opposite sign slopes. Our goal is to find these two lines and the intersection point between them.

The bottleneck point (V.1) can give us an idea about whether the distance function line is increasing or decreasing. Thus, given $e = (v_s, v_t) \in E$ and for all $v_i \in V$ we can get these relationships:

$$\begin{aligned} b_i > 0 &\Leftrightarrow \text{distance function line of vertex } v_i \text{ is increasing to the left of } b_i. \\ b_i < l_e &\Leftrightarrow \text{distance function line of vertex } v_i \text{ is decreasing to the right of } b_i. \end{aligned} \tag{V.3}$$

Replace b_i in (V.3), and let $d_i = d(v_s, v_i) - d(v_t, v_i)$. Then:

$$\begin{aligned} d_i < l_e &\Leftrightarrow \text{increasing distance function line.} \\ -d_i < l_e &\Leftrightarrow \text{decreasing distance function line.} \end{aligned} \tag{V.4}$$

We divide the set of nodes V into two sets, depending on whether the distance function increases or decreases from v_s:

$L = \{v_k \in V : d_k < l_e\}$: nodes with $d(x, v_k)$ increasing from the left-end node v_s (Figure I.2a,c).
$R = \{v_k \in V : -d_k < l_e\}$: nodes with $d(x, v_k)$ increasing from the right-end node v_t (Figure I.2b,c).

A node v_k may belong to both sets, and hence, $|L| + |R| \leq 2n$. For any node $v_i \in V$, we now define the functions $F_i^L(x)$ and $F_i^R(x)$ as:

$$F_i^L(x) = w_i(x + d(v_s, v_i))$$
$$F_i^R(x) = w_i(l_e - x + d(v_t, v_i))$$

For any pair of nodes $v_i \in L, v_j \in R$ we also define

$$X(v_i, v_j) = \frac{w_j(l_e + d(v_t, v_j)) - w_i d(v_s, v_i)}{w_i + w_j}$$

which computes the intersection point between two distance function lines with opposite sign slopes, that is, the point *x* where both $F_i^L(x)$ and $F_j^R(x)$ are equal. For the special case where $v_i = v_j$, we get the bottleneck point b_i.

Note that our goal is to find the two distance function lines (with opposite sign slopes) which cross at the maximum value of the objective function. Since there are at most n distance function lines in sets L and R, there are at most n^2 possible intersection points. Let P_e be the set containing such intersection points for a given edge $e \in E$:

$$P_e = \{X(v_i, v_j) : \forall v_i \in L,\ \forall v_j \in R\}, \quad |P_e| \leq n^2$$

and let P_N be the set obtained joining, for each edge, all the points belonging to P_e, that is

$$P_N = \bigcup_{e \in E} P_e, \quad |P_N| \leq mn^2$$

Hooker, Garfinkel and Chen (1991) defined the *arc bottleneck point set* $B_A = \{b_i : v_i \in V\}$, and the *center bottleneck point set* B_C. This set B_C contains points $x \in e$ such that, for any two distinct nodes $v_i, v_j \in V$, $w_i d(x, v_i) = w_j d(x, v_j)$, and besides, $d(x, v_i)$ and $d(x, v_j)$ do not both decrease when x is perturbed slightly in either direction. Obviously, $B_A \subset P_e$ and $B_C \subset P_e$.

Let $v_i \in L$ and $v_j \in R$. If $v_i = v_j$, then $X(v_i, v_i) = b_i \in B_A$. On the other hand, if $v_i \neq v_j$ then $X(v_i, v_j) \in B_C$. Hence, $P_e = B_A \cup B_C$.

Melachrinoudis and Zhang (1999) stated that the Finite Dominating Set (FDS) for the 1-maximin problem on networks with positive weights is $V \cup B_A \cup B_C$ (this result is also described more generally in Hooker, Garfinkel and Chen, 1991). Nevertheless, this is rather mistaken, and needs to be fixed. The following result determines the correct FDS.

Lemma V.1. *The Finite Dominating Set for the weighted 1-uncenter problem on networks is* P_N.

Proof. According to Property V.1, the value of the objective function is zero at the ends of the edges, so the maximum can never be at those points. On the other hand, this maximum value is unique on each edge (Property V.2), and must be attained at the crossing point of two distance function lines with opposite sign slopes. These points are in P_e. Therefore, the FDS for the weighted network 1-uncenter problem is P_N. ■

Taking into account these last results, we can get a new formulation for the 1-uncenter problem (V.2) as follows.

Given $e = (v_s, v_t) \in E$, let $F(x) = \{F_i^L(x) : \forall v_i \in L\}$ (or $F(x) = \{F_i^R(x) : \forall v_i \in R\}$) be the set of left (right) weighted distance functions on edge e. We define the point z_e on edge e such that $F(z_e) = \min_{x \in P_e} F(x)$.

Lemma V.2. *The local 1-uncenter point* x_e *on edge e is* z_e.

Proof. Property V.1 and Property V.2 state that $f(x)$ is a concave function and has a unique maximum x_e. This point is obtained intersecting one increasing line $F_i^L(x)$ with a decreasing line $F_j^R(x)$. Therefore x_e must belong to set P_e.

Now we show that $x_e = z_e$. By the definition of z_e, we always have $F_i^L(x_e) \geq F_i^L(z_e)$. If $x_e \neq z_e$, and since all weights w_i must be positive, the line segments of function $f(x)$ have non-zero slope, and thus $F_i^L(x_e) \neq F_i^L(z_e)$. Hence, we have $F_i^L(x_e) > F_i^L(z_e)$, which means that x_e would not be a local 1-uncenter point, and the result follows. ■

Recall from (V.2) that our goal is to find a point on the network which maximizes the minimum distance from that point to the closest one. Then, denoting F_e as the value $F(x_e) = F(z_e)$, the original problem is equivalent to the next one.

Theorem V.1. *The 1-uncenter problem on networks can be expressed as*

$$\max_{e \in E} \min_{x \in P_e} F(x)$$

and a point $x_N \in N$ *is a 1-uncenter point iff* $F(x_N) = \max_{e \in E} F_e$.

Proof. According to Lemma V.2, on each edge *e* the value of $\max_{e \in E} f(x_e)$ is F_e. Hence, the optimum value x_N on network *N* is the maximum of all F_e. That is, $\max_{e \in E} \min_{x \in P_e} F(x)$. ■

Taking into consideration the previous result, the initial continuous 1-uncenter problem (V.2) on networks becomes a discrete problem. Finally we remark that, despite the size of set P_N being at most mn^2, the 1-uncenter point can be found on a network in $O(mn)$ time. This result is proved in a subsequent section, where the new algorithm is presented. Previous to this, we briefly comment on the latest approaches and bounds cited in the literature, along with the new bounds that we propose.

V.4 Latest approaches and new bounds

As we mentioned in the introduction, few papers have been devoted to the 1-uncenter problem on networks thus far. One of the latest algorithms in $O(mn)$ time has been presented by Melachrinoudis and Zhang (1999).

Their approach relies on three upper bounds that significantly reduce the number of edges and, over each edge, the number of distance function lines. Given an edge $e = (v_s, v_t) \in E$, the first upper bound is defined as $x_{UB1} = X(v_s, v_t)$ and $F_{UB1} = F_s^L(x_{UB1}) = F_t^R(x_{UB1})$ (see Figure V.2).

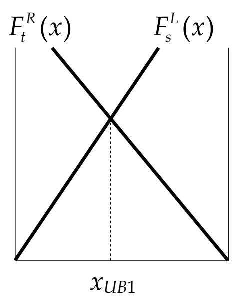


Figure V.2: F_{UB1}, the first upper bound.

This bound cannot be improved. Nevertheless, the next two bounds can be tightened. Let

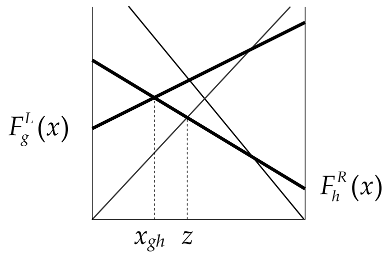
$$v_g \in V : F_g^L(0) = \min_{\substack{v_k \in V \\ v_k \neq v_s}} F_k^L(0), \qquad v_h \in V : F_h^R(l_e) = \min_{\substack{v_k \in V \\ v_k \neq v_t}} F_k^R(l_e) \tag{V.5}$$

be the nodes at which the distance functions attain their minimum value on each side. Ties are broken taking the node with the smallest weight *w*. The second upper bound is $x_{gh} = X(v_g, v_h)$ and $F_{gh} = F_g^L(x_{gh}) = F_h^R(x_{gh})$.

However, upper bound F_{gh} may be slightly improved in two special cases (see Figure V.3). So, we introduce a new point *z* and its ordinate, which are defined by:

$$(z, F_z) = \begin{cases} (X(v_s, v_h), F_s^L(X(v_s, v_h))) & \text{if } F_s^L(x_{gh}) \le F_{gh} \quad \text{(Figure 3a)} \\ (X(v_g, v_t), F_t^R(X(v_g, v_t))) & \text{if } F_t^R(x_{gh}) \le F_{gh} \quad \text{(Figure 3b)} \\ (0, \infty) & \text{otherwise} \end{cases} \qquad \text{(V.6)}$$

Then, we propose the new bound $F_{UB2} = \min\{F_{gh}, F_z, F_{UB1}\}$, and hence, x_{UB2} is equal to x_{gh}, z or x_{UB1}.



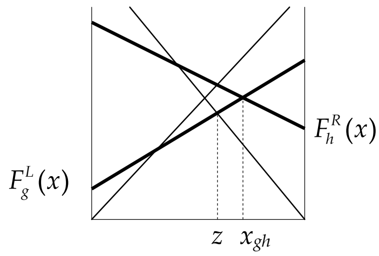


Figure V.3: Tighter bounds. The value of F_z is better than F_{gh}.

Any distance function line over F_{UB2} is redundant and, therefore, can be completely removed. Despite the upper bound F_{gh} has been tightened to F_{UB2}, the proof in Melachrinoudis and Zhang (1999) is valid for this result as well.

Likewise, the third upper bound is defined considering

$$v_p \in V : F_p^L(l_e) = \min_{\substack{v_k \in V \\ v_k \ne v_s}} F_k^L(l_e), \qquad v_q \in V : F_q^R(0) = \min_{\substack{v_k \in V \\ v_k \ne v_t}} F_k^R(0) \qquad \text{(V.7)}$$

with $x_{pq} = X(v_p, v_q)$ and $F_{pq} = F_p^L(x_{pq}) = F_q^R(x_{pq})$.

This bound F_{pq} can also be improved by establishing a new point *y* and its ordinate, which are defined by:

$$(y, F_y) = \begin{cases} (X(v_s, v_q), F_s^L(X(v_s, v_q))) & \text{if } F_s^L(x_{pq}) \le F_{pq} \\ (X(v_p, v_t), F_t^R(X(v_p, v_t))) & \text{if } F_t^R(x_{pq}) \le F_{pq} \\ (0, \infty) & \text{otherwise} \end{cases} \qquad \text{(V.8)}$$

Then, we propose the new bound $F_{UB3} = \min\{F_{pq}, F_y, F_{UB1}\}$ and x_{UB3} is updated accordingly to x_{pq}, y or x_{UB1}.

Before presenting the new algorithm which makes use of bounds (V.6) and (V.8), we now outline the rest of Melachrinoudis and Zhang's algorithm.

Once all the lines above $\min\{F_{UB1}, F_{gh}\}$ are deleted, the remaining lines are compared pairwise. For each pair of lines, either the intersection point is calculated or one of them is deleted (dominated). Then, the median value of the intersection points is projected on the maximin function (lowest lines). If the right and left gradients have opposite sign slopes, the maximin point is found. Otherwise, the gradients are used to delete a quarter of the paired lines. In the worst case, the procedure keeps on until two lines remain only.

The main disadvantage of this pairing algorithm is the matching of superfluous distance function lines, that is, lines that do not actually exist (see Figure V.4). These lines load the algorithm with useless computational effort and, therefore, they need to be excluded.

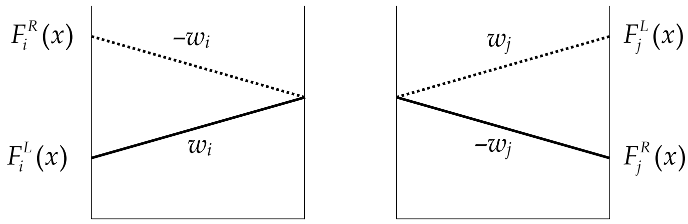


Figure V.4: Superfluous lines are plotted as dotted lines.

On the other hand, the most recent contribution to the 1-uncenter problem is due to Berman and Drezner (2000), who presented a brief paper on the location of an obnoxious facility on a network. They addressed this problem from a linear programming viewpoint, making use of the algorithm given in Megiddo (1982) to get an $O(mn)$ time procedure. However, this approach is not very fast (computationally speaking) since every single edge has to be checked to find the optimal value. This fact is proved later in the computational experience section.

All the improvements discussed above, together with the new upper bounds, are shown in the next algorithm that we propose to solve the 1-uncenter problem.

V.5 The algorithm

The algorithm has two main parts: the first one computes the three upper bounds; the second one seeks for the best point in the set of remaining distance function lines. For the sake of comprehensibility, we present the outlined algorithm (see Algorithm V.1), and in the following paragraphs we explain each block of code.

The function *UnCenter* needs only two inputs: the network $N=(V,E)$ and the distance matrix *d*, which can be computed in $O(mn+n^2\log n)$ time using Fredman and Tarjan (1987). The output is F_N and the set of points *S* where this value is attained.

The calculation of the first upper bound is easy. The second one is computed using (V.5) and (V.6), whereas (V.7) and (V.8) calculate the third upper bound.

Then, the pair (x_e,F_e) is set to the best upper bound. The purpose of the rest of the algorithm is to sharpen F_e until the optimal value is found.

Next, we divide set *V* into two sets *L* and *R*. The distance function lines belonging to these sets are then matched, so that the number of matchings must be equal to $\max\{|L|,|R|\}$. For example, let $L=\{v_1,v_3,v_4\}$ and $R=\{v_2,v_3,v_5,v_7,v_8\}$. Then, the specific matchings $(v_i\in L, v_j\in R)$ are (v_1,v_2), (v_3,v_3), (v_4,v_5), (v_1,v_7) and (v_3,v_8). In each pairing, the intersection point between the two lines and its related ordinate value are computed. Besides, any dominated line is immediately removed. The intersection point with minimal function value is stored in (x_e,F_e).

```
function UnCenter(Network N, Distance Matrix d)
{   // Current best value on network N.
    F_N := 0
    // Solution set.
    S := ∅
    for all edges e := (v_s, v_t) ∈ E do
        {   // Compute the upper bounds.
            x_UB1 := X(v_s, v_t)
            F_UB1 := F_s^L(x_UB1)
            if F_N > F_UB1 then continue to next edge
            Compute UB2 using (V.5) and (V.6)
            if F_N > F_UB2 then continue to next edge
            Compute UB3 using (V.7) and (V.8)
            if F_N > F_UB3 then continue to next edge
            // Set (x_e, F_e) to the best value found.
            if F_UB2 ≤ F_UB3 then   (x_e, F_e) := (x_UB2, F_UB2)
            else                    (x_e, F_e) := (x_UB3, F_UB3)
            Create sets L and R using (V.4). All lines must be below F_UB2.
            // Continue till the new value F_e cannot improve the current F_N,
            // or until one of the node sets becomes empty.
            while F_e ≥ F_N and ( L ≠ ∅ or R ≠ ∅ ) do
                {   Pair all nodes in L against R, using a max{|L|,|R|} matching
                    Store the intersection point with minimal function value in (x_e, F_e)
                    Project the value x_e on the lower envelope using (V.9) to get v_a and v_b
                    x_e := X(v_a, v_b)
                    F_e := F_a^L(x_e)
                    Remove from L and R all lines above the new value F_e
                }
            if F_e ≥ F_N then
                {   F_N := F_e
                    Store the pair (x_e, e) in S
                }
        }
    return (F_N, S)
}
```

Algorithm V.1: The uncenter function.

The value of x_e is projected on the objective function (lower envelope), and thus, we obtain a new value for (x_e, F_e). All lines above F_e are then deleted from *L* and *R*. The algorithm keeps going until either $F_e < F_N$, that is, this edge cannot improve the network optimum, or both *L* and *R* are empty. The complete code is shown in the Appendix.

The maximum matching assures a maximum of *n* paired lines, which is essential to delete as many lines as possible. The following lemma states this result.

Lemma V.3. *In each iteration of the 'while' loop, at least* $(\max\{|L|,|R|\})/2$ *nodes from L and R are removed.*

Proof. For each of the paired lines (v_i, v_j), $v_i \in L, v_j \in R$, let $Q_e = \{X(v_i, v_j)\}$ be such that $|Q_e| = \max\{|L|,|R|\}$, that is, Q_e contains all the intersection points of the line pairing. Let $F_e = \min_{\substack{x \in Q_e \\ v_i \in L}} F_i^L(x)$ and x_e be, respectively, the minimum value of all the paired lines and the point where this minimal value is attained.

The value F_e might be optimal. Obviously, all lines belonging to *L* and *R* are then deleted. Otherwise, let

$$v_a \in L : F_a^L(x_e) = \min_{v_k \in L} F_k^L(x_e), \qquad v_b \in R : F_b^R(x_e) = \min_{v_k \in R} F_k^R(x_e) \tag{V.9}$$

be the lowest lines (lower envelope) from *L* and *R* (ties are broken taking the lower weight *w*). Let $x_e = X(v_a, v_b)$ and $F_e = F_a^L(x_e)$. This F_e is a new upper bound. Also, since $F_a^L(x_e)$ or $F_b^R(x_e)$ belongs to the lower envelope, any line above F_e can be removed. Indeed, each pair of lines (v_i, v_j) can only have one single line under F_e, to be precise, either $F_i^L(x_e) < F_e$ or $F_j^R(x_e) < F_e$. Both lines v_i and v_j cannot be below F_e since that contradicts the fact that F_e is the minimal value. Then, in the worst case, one single node belonging to each pair (v_i, v_j) can be removed from *L* or *R*. Therefore, each removal process deletes at least $|Q_e|/2$ nodes (lines). ■

Given the distance matrix, the following theorem proves that the overall complexity of the new 1-uncenter algorithm is $O(mn)$.

Theorem V.2. *The previous algorithm solves efficiently the weighted 1-uncenter problem in* $O(mn)$ *time.*

Proof. The computation of the second and third upper bounds takes $O(n)$ time. The size of *L* and *R* is, in the worst case, $n \geq \max\{|L|,|R|\}$ nodes. According to Lemma V.3, each iteration of the 'while' loop deletes $n/2$ nodes. Therefore, the complexity of that loop is:

$$n + \frac{n}{2} + \frac{n}{4} + \cdots + \frac{n}{2^k} = n\left(\frac{2^k + 2^{k-1} + \cdots + 1}{2^k}\right) = \frac{n}{2^k}\sum_{i=0}^{k} 2^i = \frac{n}{2^k}(2^{k+1} - 1)$$

In the worst case, this loop keeps on till one single line remains in both *L* and *R*. Then $n/2^k = 2 \Rightarrow n = 2^{k+1}$, and consequently, $(n/2^k)(2^{k+1} - 1) = 2(n-1) < 2n \in O(n)$.

This process must be applied to all *m* edges. Thus, the overall complexity is $O(mn)$. ■

The time complexity given in Melachrinoudis and Zhang (1999) was bounded by 4*n*, and hence, this may explain why the new algorithm is much faster. Moreover, as you may have noticed, the 1-uncenter algorithm does not make use of the median algorithm. Next, we illustrate the proposed algorithm with a brief example.

V.6 An example

The network is depicted in Figure V.5. It has $n = 8$ nodes and $m = 18$ edges. The weights (in bold) on the nodes range randomly from 1 to 9, whereas the lengths (in italics) randomly vary from 1 to 49. The trace of the algorithm is summarized in Table V.1.

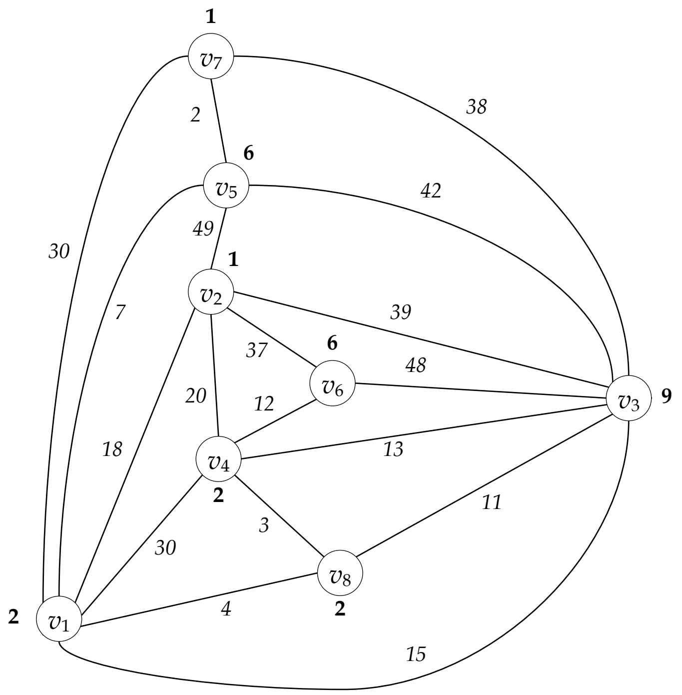


Figure V.5: Planar network with $n = 8$ and $m = 18$.

In the first iteration, the three upper bounds are computed. The best of them is *UB*3. Since *R* is empty, there is no line pairing. Thus, the first local 1-uncenter on edge (v_1, v_3) is located at $x_e = 12.6$, with $F_e = 21.6$. The solution set *S* and the value F_N are updated.

The best upper bound on edge (v_1, v_4) is $(17, 26)$. Again, there is no line pairing and, since $F_e = 26 > F_N$, the set *S* is updated. The next four edges cannot improve F_N.

Edge (v_2, v_3) updates the best network 1-uncenter value to $F_N = 31.5$. The next edge (v_2, v_4) leaves F_N and *S* unchanged, while in the iteration of edge (v_2, v_5) the algorithm steps to the following edge as soon as it checks that *UB*2 is worse than F_N.

The algorithm keeps on in the same way with edges (v_2, v_6) and (v_3, v_4), updating the network 1-uncenter value $F_N = 31.71$ and $S = \{(31.71, e_{26})\}$. The first lines paired arise in edge (v_3, v_5). The pairing is: (v_7, v_7) and (v_8, v_7), which provides a new $(x_e, F_e) = (10, 34)$, and hence, a new F_N and *S*.

In the next edge (v_3, v_6) the line matching cannot improve $(x_e, F_e) = (26, 50)$. Given that no remaining edge provides a better value, $F_N = 50$ becomes the 1-uncenter value at $S = \{(26, e_{36})\}$.

Note that the algorithm processes only 6 out of 18 potential edges, with only 5 pairings. For the same example, the maximin algorithm by Melachrinoudis and Zhang (1999) needs to process 7 edges, and computes 26 pairings. Even though these numbers may not seem important, they will be quite relevant when the network size gets bigger (both in nodes and edges), as shown in the next section.

Edge	F_N	(x_{UB1}, F_{UB1})	(x_{UB2}, F_{UB2})	(x_{UB3}, F_{UB3})	(x_e, F_e)	*L*	*R*	*S*
$e_{13}=(v_1, v_3)$	0	(12.27, 24.54)	(12.27, 24.54)	(12.6, 21.6)	(12.6, 21.6)	$\{v_7\}$	∅	$\{(12.6, e_{13})\}$
$e_{14}=(v_1, v_4)$	21.6	(15, 30)	(15, 30)	(17, 26)	(17, 26)	$\{v_7\}$	∅	$\{(17, e_{14})\}$
$e_{15}=(v_1, v_5)$	26	(5.25, 10.5)	–	–	–	–	–	$\{(17, e_{14})\}$
$e_{17}=(v_1, v_7)$	26	(10, 20)	–	–	–	–	–	$\{(17, e_{14})\}$
$e_{18}=(v_1, v_8)$	26	(2, 4)	–	–	–	–	–	$\{(17, e_{14})\}$
$e_{21}=(v_2, v_1)$	26	(12, 12)	–	–	–	–	–	$\{(17, e_{14})\}$
$e_{23}=(v_2, v_3)$	26	(35.1, 35.1)	(33.33, 33.33)	(31.5, 31.5)	(31.5, 31.5)	∅	$\{v_7, v_8\}$	$\{(31.5, e_{23})\}$
$e_{24}=(v_2, v_4)$	31.5	(13.33, 13.33)	–	–	–	–	–	$\{(31.5, e_{23})\}$
$e_{25}=(v_2, v_5)$	31.5	(42, 42)	(25.5, 25.5)	–	–	–	–	$\{(31.5, e_{23})\}$
$e_{26}=(v_2, v_6)$	31.5	(31.71, 31.71)	(31.71, 31.71)	(31.71, 31.71)	(31.71, 31.71)	–	–	$\{(31.71, e_{26})\}$
$e_{34}=(v_3, v_4)$	31.71	(2.36, 21.27)	–	–	–	–	–	$\{(31.71, e_{26})\}$
$e_{35}=(v_3, v_5)$	31.71	(16.8, 151.2)	(7.33, 36.66)	(10, 34)	(10, 34)	$\{v_7, v_8\}$	$\{v_7\}$	$\{(10, e_{35})\}$
$e_{36}=(v_3, v_6)$	34	(19.2, 172.8)	(24.5, 71)	(26, 50)	(26, 50)	$\{v_2, v_7, v_8\}$	$\{v_2, v_4, v_7\}$	$\{(26, e_{36})\}$
$e_{37}=(v_3, v_7)$	50	(3.8, 34.2)	–	–	–	–	–	$\{(26, e_{36})\}$
$e_{38}=(v_3, v_8)$	50	(2, 18)	–	–	–	–	–	$\{(26, e_{36})\}$
$e_{46}=(v_4, v_6)$	50	(9, 18)	–	–	–	–	–	$\{(26, e_{36})\}$
$e_{48}=(v_4, v_8)$	50	(1.5, 3)	–	–	–	–	–	$\{(26, e_{36})\}$
$e_{57}=(v_5, v_7)$	50	(0.28, 1.71)	–	–	–	–	–	$\{(26, e_{36})\}$

Table V.1: Trace of the 1-uncenter algorithm for the network of Figure V.5.

V.7 Computational results

The computational results were developed using GNU g++ 2.95.2 programming language and LEDA (*Library of Efficient Datatypes and Algorithms*; see Melhorn and Näher, 1999), on a PC AMD K6-III 400 Mhz under Red Hat Linux 6.1 (Cartman). The sources were built using the g++ compiler optimizing option '–O'.

The distance matrix was computed using an algorithm developed in LEDA, which is claimed to run in $O(mn + n^2 \log n)$ time.

For the sake of a homogeneous comparison with the algorithm reported by Melachrinoudis and Zhang (1999), we keep the same node weight range from 1 to 9, edge length ranges from 1 to 49, and the edge density $d = m/(n(n-1)/2)$ equal to 1/2, 1/4, 1/8 and 1/16. However, the sizes of the networks were too small for such a fast computer, since they provided computational times near to zero seconds. Thus, we decided to run the experiments from $n = 100$ up. The networks were created using the random graph generators provided by LEDA.

Before the comparison with the algorithm by Melachrinoudis and Zhang (1999), we present the results obtained for the comparison between the new algorithm and the linear programming approach proposed by Berman and Drezner (2000). For this task, we made use of the free linear solver *lp_solve* (available at *ftp.es.ele.tue.nl/pub/lp_solve*). Since their method relies on an LP process over each and every edge, we decided to test the algorithms on low density networks. Thus, we created planar networks with $m = 3n - 6$ and $n = 100$ to 500, in steps of 25 nodes. Ten instances were generated for each value of *n*. Table V.2 illustrates the average processed edges and the average computing time for the three experiments accomplished. The label "B & D" stands for Berman and Drezner.

The first column in Table V.2 shows the results for the original approach by Berman and Drezner (2000). These times are extremely high, since their method has to run over all existing edges. The next column shows the results for the same approach including the new upper bounds proposed in this chapter. These bounds remarkably reduce the number of processed edges, and hence, the overall computing times. Finally, the third column presents the computing results of the new algorithm, which achieves faster computing times than the bounded version of Berman and Drezner. The time reduction percent between these two latter procedures is shown in the last column.

n	*B & D*		*B & D (with UBs)*		*New algorithm*		
	Processed edges	*Time (sec.)*	*Processed edges*	*Time (sec.)*	*Processed edges*	*Time (sec.)*	*Reduction (%)*
100	294	1.611	6	0.046	6	0.010	78
125	369	2.859	5	0.055	5	0.014	75
150	444	4.593	7	0.095	7	0.020	79
175	519	6.902	6	0.112	6	0.023	79
200	594	11.608	7	0.183	7	0.033	82
225	669	16.453	6	0.203	6	0.047	77
250	744	26.371	9	0.321	9	0.054	83
275	819	28.585	8	0.375	8	0.068	82
300	894	37.029	7	0.380	7	0.076	80
325	969	47.419	8	0.496	8	0.085	83
350	1044	57.553	8	0.570	8	0.101	82
375	1119	70.416	8	0.625	8	0.114	82
400	1194	82.602	7	0.678	7	0.134	80
425	1269	103.021	8	0.867	8	0.133	85
450	1344	110.540	8	0.758	8	0.130	83
475	1419	144.851	8	0.892	8	0.155	83
500	1494	169.766	7	0.864	7	0.159	82

Table V.2: Processed edges and computing times of Berman & Drezner's procedure and the new algorithm for planar networks ($m = 3n - 6$) with $n = 100$ to 500 nodes.

Regarding the comparison with Melachrinoudis and Zhang's procedure, three kinds of experiments were performed. In the first one, n varies from 100 to 500 nodes in steps of 25, with d equal to 1/2, 1/4, 1/8 and 1/16. In the second, the number of nodes ranges from 525 to 1000 in steps of 25 nodes, with $d = 1/8$ and 1/16. In the last experiment, random planar ($m = 3n - 6$) networks were generated for $n = 1000$ up to 5000, with a step of 250 nodes. In all cases, ten instances of each combination were run. The comparison is based on the average value of the processed edges, line pairings and computing time. The label "M & Z" stands for Melachrinoudis and Zhang.

Figure V.6 and Figure V.7 show the processed edges, line pairings and computing times for different number of nodes and edge densities. Due to the tighter bounds, there are fewer edges processed by the 1-uncenter algorithm than by the maximin procedure. Besides, the number of paired lines is much less in our algorithm. Likewise, the 1-uncenter algorithm beats the maximin in all the computing time graphics. Finally, in Figure V.8 we also describe the results for random planar networks. It seems that the 1-uncenter algorithm behaves even better than compared to the maximin procedure when the number of edges m is $O(n)$. In this particular case, the gap between the two algorithms is quite large.

In Table V.3 we show an overall summary of numerical results obtained for the different set of densities as well as for planar networks. In all cases, the number of edges processed by our algorithm, and the number of matchings (line crossings) is fewer than Melachrinoudis and Zhang, gaining in some instances a reduction of over 50%. As a consequence of all this, the computing times of the new algorithm are better, achieving in some cases a reduction of 80%. Besides, the reduction augments as the number of nodes n increases.

V.8 Concluding remarks

The location of an undesirable facility under the max-min criterion is addressed. As it was stated in the introduction, there are only a few references to this problem in the literature. One of the latest is by Melachrinoudis and Zhang (1999), who proposed a $O(mn)$ time algorithm based on three upper bounds and on a modified procedure of Dyer (1984). However, we show that their upper bounds can be tightened, and that pairing superfluous lines is not needed. The other paper by Berman and Drezner (2000) approaches the problem in a linear programming way. Though it has the same theoretical complexity, its running times are extremely high, since the algorithm has to process every single edge.

Hence, using tighter bounds and eliminating the superfluous line pairing by means of a more convenient problem formulation, we propose a new $O(mn)$ time algorithm. Besides, the algorithm needs no median procedure. As a result of all this, the proposed algorithm is more straightforward and its running times are faster than the ones already reported by Melachrinoudis and Zhang (1999).

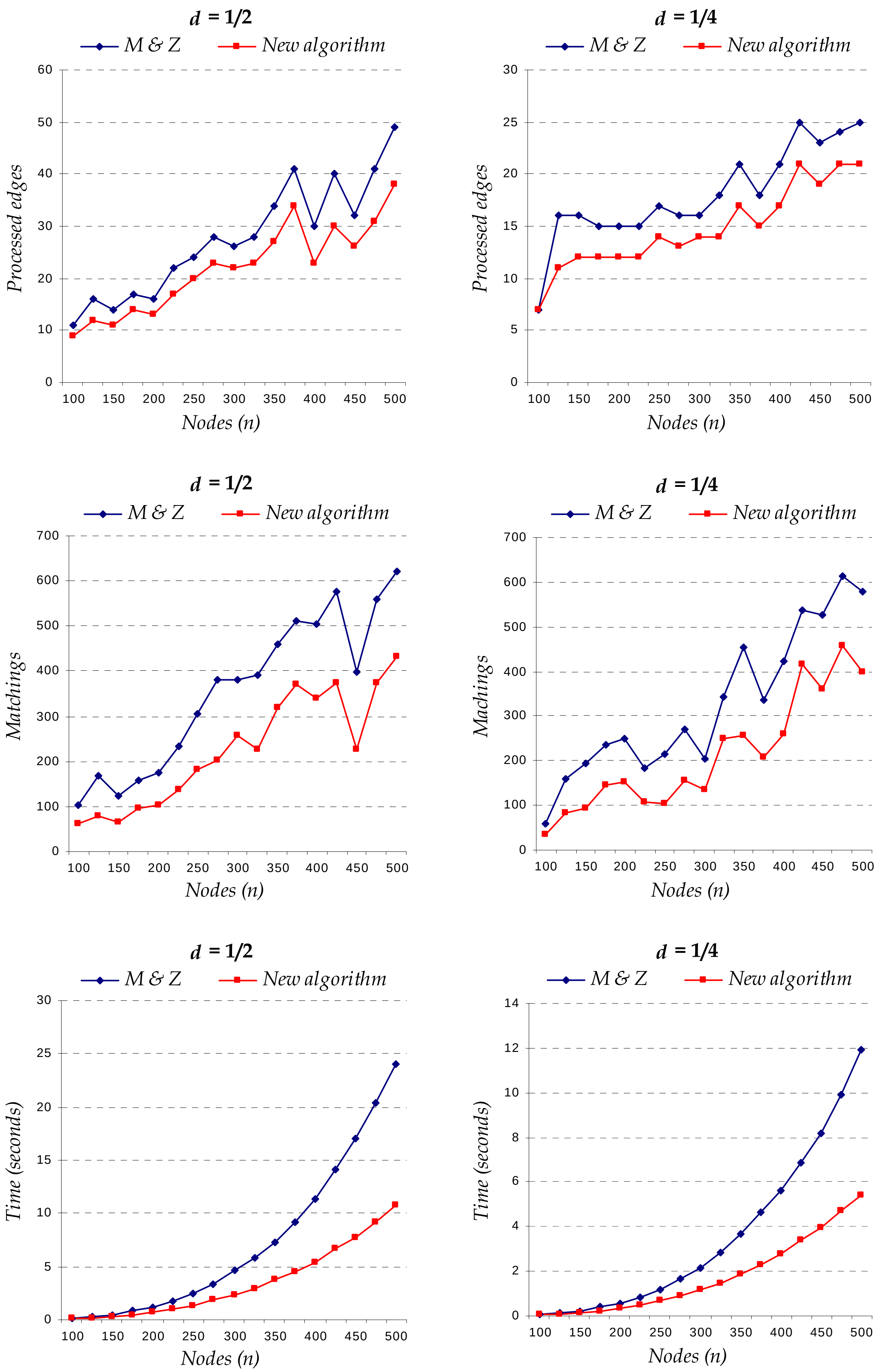


Figure V.6: Processed edges, line pairings (matchings) and computing times for $d = 1/2$ and $d = 1/4$ with $n = 100$ to 500.

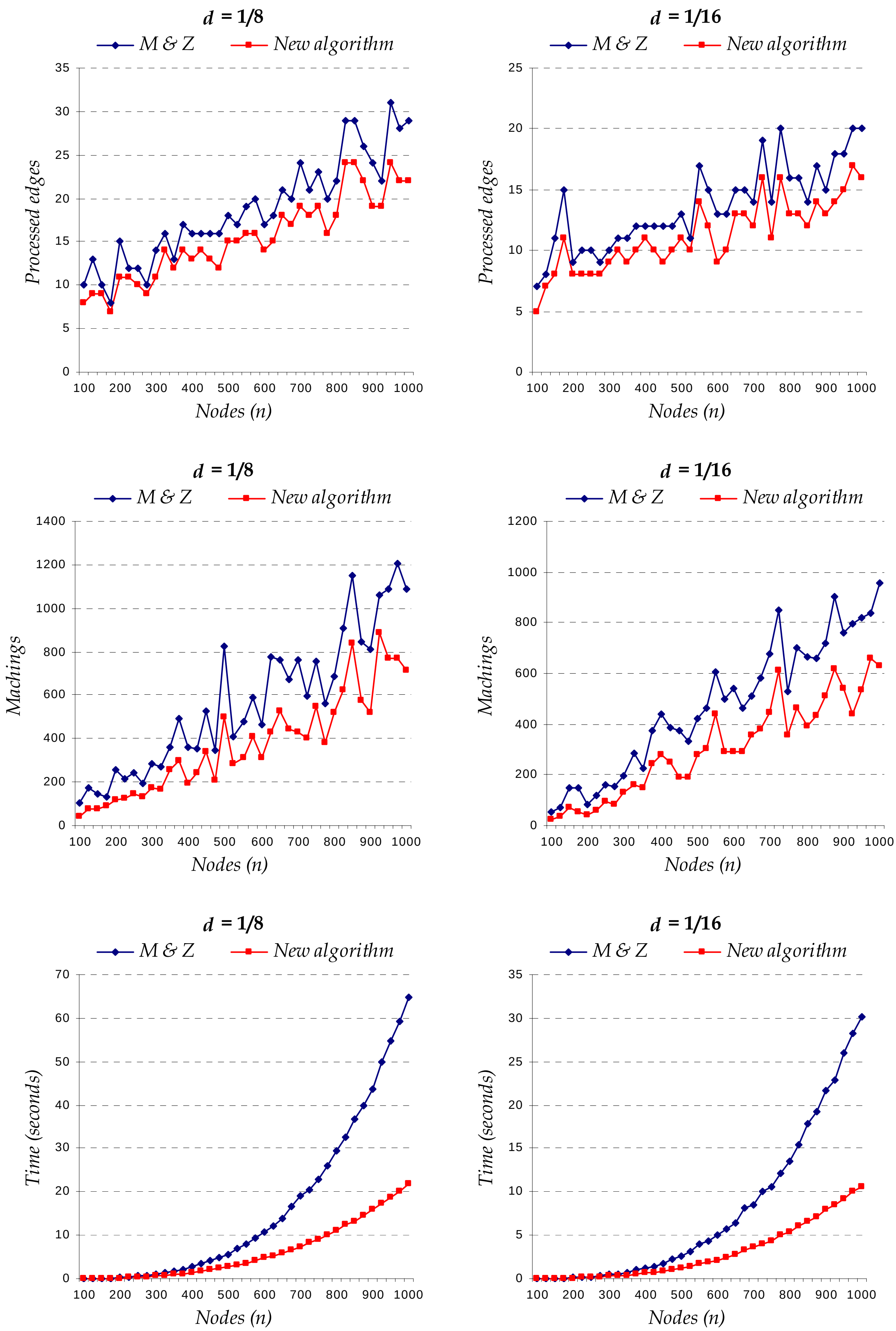


Figure V.7: Processed edges, line pairings (matchings) and computing times for $d = 1/8$ and $d = 1/16$ with $n = 100$ to 1000.

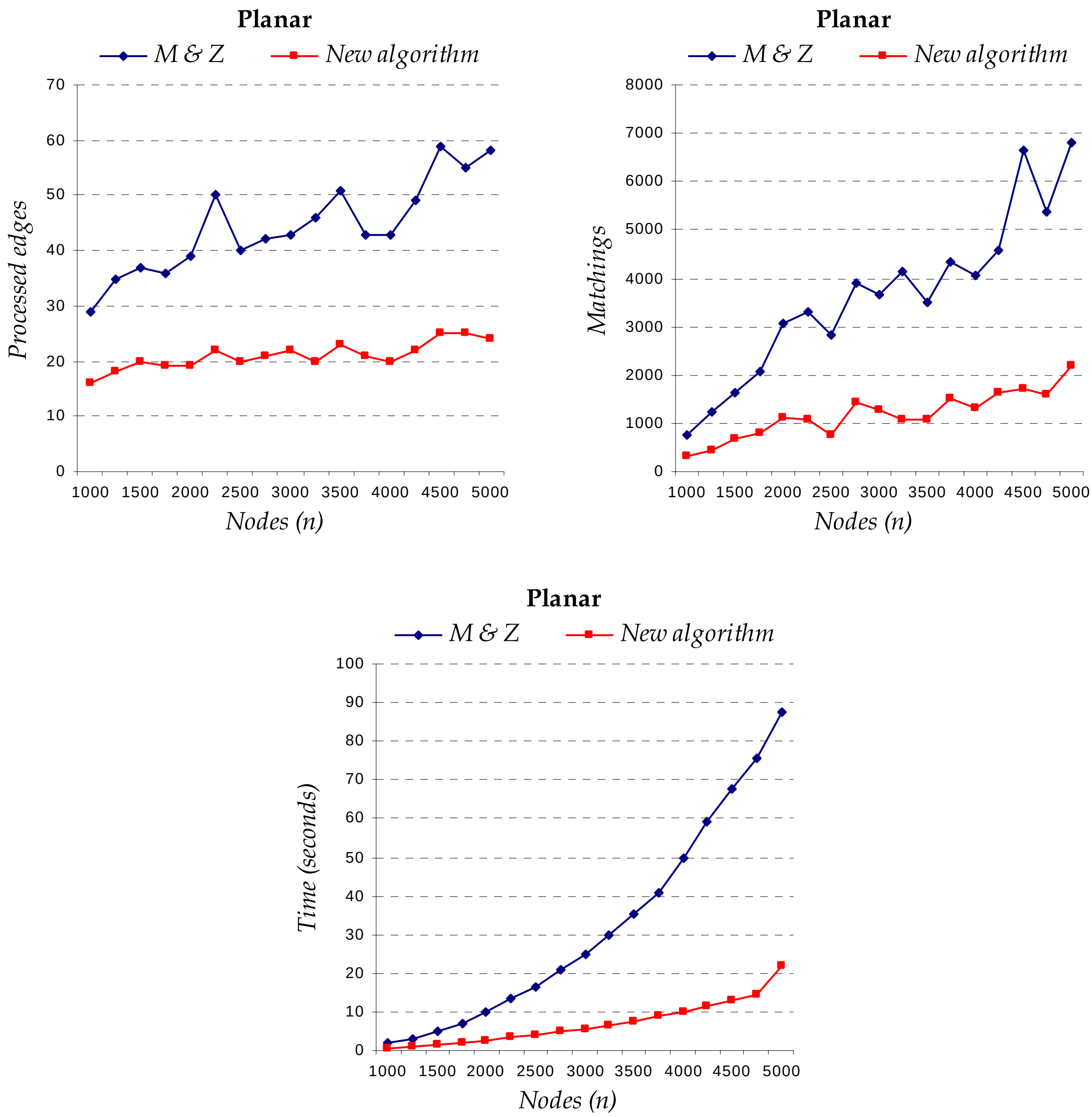


Figure V.8: Processed edges, line pairings and computing time for planar networks ($m = 3n - 6$) and $n = 1000$ to 5000 vertices.

d	n	Processed edges			Matchings			Time		
		M & Z	*New algorithm*	*Reduction (%)*	*M & Z*	*New algorithm*	*Reduction (%)*	*M & Z*	*New algorithm*	*Reduction (%)*
1/2	100	11	9	18	103	61	41	0.144	0.092	36
	200	16	13	19	175	102	42	1.209	0.708	41
	300	26	22	15	382	259	32	4.595	2.399	48
	400	30	23	23	503	341	32	11.364	5.451	52
	500	49	38	22	620	434	30	24.042	10.813	55
1/4	100	7	7	0	60	33	45	0.070	0.046	34
	200	15	12	20	249	152	39	0.585	0.350	40
	300	16	14	13	203	135	33	2.154	1.148	47
	400	21	17	19	422	261	38	5.584	2.753	51
	500	25	21	16	579	397	31	11.955	5.389	55
1/8	125	13	9	31	172	75	56	0.061	0.044	28
	250	12	10	17	243	148	39	0.562	0.331	41
	375	17	14	18	492	296	40	2.215	1.130	49
	500	18	15	17	825	496	40	5.576	2.662	52
	625	18	15	17	775	433	44	12.208	5.276	57
	750	23	19	17	752	547	27	22.744	9.010	60
	875	26	22	15	847	578	32	39.812	14.669	63
	1000	29	22	24	1085	714	34	64.705	21.680	66
1/16	125	8	7	13	73	33	55	0.024	0.016	33
	250	10	8	20	160	93	42	0.238	0.138	42
	375	12	10	17	373	246	34	0.976	0.522	47
	500	13	11	15	424	280	34	2.560	1.270	50
	625	13	10	23	461	294	36	5.736	2.505	56
	750	14	11	21	530	355	33	10.644	4.359	59
	875	17	14	18	904	620	31	19.295	7.144	63
	1000	20	16	20	954	629	34	30.149	10.623	65
Planar	1000	29	16	45	774	304	61	1.846	0.670	64
	1500	37	20	46	1639	658	60	5.209	1.513	71
	2000	39	19	51	3071	1112	64	9.770	2.542	74
	2500	40	20	50	2813	766	73	16.255	3.752	77
	3000	43	22	49	3675	1289	65	25.009	5.525	78
	3500	51	23	55	3516	1072	70	35.133	7.503	79
	4000	43	20	53	4079	1296	68	49.626	9.944	80
	4500	59	25	58	6655	1694	75	67.624	12.884	81
	5000	58	24	59	6809	2191	68	87.341	21.936	75

Table V.3: Summary of the processed edges, line pairings (matchings) and computing times for $d = 1/2, 1/4, 1/8$ and $1/16$, and for planar networks ($m = 3n - 6$).

Chapter VI

The undesirable median and anti-cent-dian location problems on networks

"Work on undesirable facility location models represents one of the major fields of research nowadays"
H.A. EISELT & G. LAPORTE

VI.1 Introduction

Modern network location theory was originally introduced by Hakimi (1964). It basically deals with finding an optimal point on the network where one or more facilities can be established, so that the service demand of users (nodes) is completely satisfied. Usually, the facilities to be located are considered "desirable" for the customers, for instance, shopping centers, emergency services, schools, etc.

However, there are some services which are not so desirable, and might be considered annoying (*obnoxious*), such as garbage dump sites, oil plants or prisons. Some of them might be even harmful (*noxious*) for the surrounding population, for instance, nuclear reactors, chemical industries and polluting plants. Anyhow, we just consider all of them *undesirable*.

The literature on undesirable network location began in the mid 70s with Church and Garfinkel (1978), who defined and solved the 1-maxisum (maxian) problem in $\mathrm{O}(mn \log n)$ time, being n the number of nodes and m the number of edges.

Later, Minieka (1983) addressed the *anti-center* (maxmax) and the *anti-median* (maxsum), which is a similar approach to the unweighted case described in Church and Garfinkel (1978). Soon after, Ting (1984) developed an algorithm in linear time for the 1-maxisum problem on trees.

Regarding the maximin problem, Tamir (1988) hinted at a method in $\mathrm{O}(mn)$ time using Megiddo (1982) and Dyer (1984). Lately, Melachrinoudis and Zhang (1999) and Berman and Drezner (2000) make use of, respectively, Dyer (1984) and Megiddo (1982), to devise analogous algorithms. The most recent approach is addressed by Colebrook, Gutiérrez, Alonso and Sicilia (2001). For a deeper and state-of-the-art survey on undesirable location, the reader is referred to Erkut and Neuman (1989) and Cappanera (1999).

Tamir (1991) briefly suggested that the 1-maxisum problem could be solved in $\mathrm{O}(mn)$ time using an algorithm given by Zemel (1984). However, to the best of our knowledge, there is no reference in the literature directly describing such an algorithm for the network 1-maxisum problem thus far. In this chapter we present a new algorithm which solves this problem in $O(mn)$ time.

The remainder of the chapter is structured as follows. First, we introduce the notation and the general properties of the 1-maxisum problem. Section VI.3 addresses a new approach to the problem based on the right and left slopes of the objective function at the end nodes of each edge. In section VI.4 we propose a new upper bound to this problem which speeds up the search of the optimal point. The next two sections describe, respectively, the new method and the $O(mn)$ algorithm. In section VI.7 a small trace is developed. The computational experience on low and high dense networks as well as planar networks is presented in section VI.8. Finally, taking into account the 1-uncenter model (Chapter IV) and the 1-maxisum model here described, we have developed a new algorithm in $O(mn)$ time for the anti-cent-dian problem. The chapter ends with the conclusions.

VI.2 Notation and general properties

Let $N=(V,E)$ be a simple (no loops or multiple edges), undirected, finite and connected network with n nodes (vertices) $V=\{v_1,v_2,\ldots,v_n\}$, and m edges $E=\{(v_s,v_t):v_s,v_t\in V\}$, with $|E|=m$. A function $w:V\to\mathbb{R}$, $w(v_i)=w_i\geq 0$ is defined, which denotes the number of customers situated at v_i who will make use of the facility's services. Obviously, we assume that not all $w_i=0$.

On the other hand, we define a function $l:E\to\mathbb{R}_+$, $l(e)=l_e>0$ that indicates the length of edge *e*. Thus, a point $x\in e$ ranges in the interval $[0,l_e]$.

Given any pair of nodes $v_i,v_j\in V$, the *distance* between these two nodes $d(v_i,v_j)$ was defined in section I.3.1 as the length of the shortest path between v_i and v_j. Then, for any $e=(v_s,v_t)\in E$ and given an inner point $x\in e$, the distance between *x* and a node v_i is

$$d(x,v_i)=\min\{x+d(v_s,v_i),l_e-x+d(v_t,v_i)\} \quad \text{(VI.1)}$$

The point on *e* where $d(x,v_i)$ attains its equilibrium, i.e. $x+d(v_s,v_i)=l_e-x+d(v_t,v_i)$, is called a *bottleneck point*:

$$b_i=\frac{d(v_t,v_i)-d(v_s,v_i)+l_e}{2} \quad \text{(VI.2)}$$

The function $d(x,v_i)$ is linear and concave with at most one bottleneck point, as shown in Figure I.2.

Let $B_e=\bigcup_{v_i\in V} b_i$ be the set of all bottleneck points on edge *e*, and let $B_N=\bigcup_{e\in E} B_e$ be the set of all bottleneck points on network *N*.

Given any point *x* on network *N*, we define

$$f(x)=\sum_{v_i\in V} w_i d(x,v_i) \quad \text{(VI.3)}$$

as the sum of weighted distances from point *x* to all the nodes of the network.

The undesirable *one-facility maximum* (maxian) *problem* is expressed as

$$\max_{x \in N} f(x) \qquad \text{(VI.4)}$$

and a point $x_N \in N$ is a *maxian* point iff $f(x_N) = \max_{x \in N} f(x)$. Several interesting properties arise for this problem.

Property VI.1. *For any edge* $e = (v_s, v_t) \in E$, *and given a point* $x \in e$, *the objective function* $f(x)$, *is continuous, piecewise linear and concave in the interval* $[0, l_e]$, *with at most* $n+1$ *monotonic line segments. Each breakpoint of function* $f(x)$ *corresponds to a bottleneck point* b_i *for a node* $v_i \in V$.

This first property follows directly from the definition of the distance function $d(x, v_i)$.

Property VI.2. *On each edge* $e = (v_s, v_t) \in E$, *there exists at least one point* x_e *that maximizes* $f(x)$, *such that* $x_e \in B_e \cup \{v_s, v_t\}$. *If function* $f(x)$ *reaches its maximum value at two consecutive points* x_e^1 *and* x_e^2, *then all points inside* $[x_e^1, x_e^2]$ *maximize function* $f(x)$ (see Figure VI.1).

This property is a direct consequence of Property VI.1.

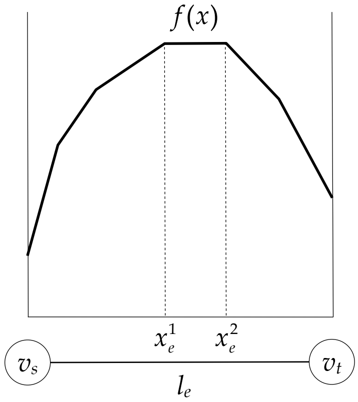


Figure VI.1: Interval $[x_e^1, x_e^2]$ maximizes function $f(x)$.

Property VI.3. *The Finite Dominating Set (FDS) for problem* (VI.4) *is* $B_N \cup V$. *The size of this set is* $n(m+1)$ (see proof in Church and Garfinkel, 1978).

According to the first two properties, problem (VI.4) can be formulated over each edge *e* as follows:

$$f(x_e) = \max_{x \in e} f(x) \qquad \text{(VI.5)}$$

and a point $x_N \in N$ is a maxian point iff $f(x_N) = \max_{e \in E} f(x_e)$. According to the classification scheme described in section I.4, this problem is classified as $1/\mathcal{G}/\bullet/d(\mathcal{V},\mathcal{G})/\sum_{\text{obnox}}$.

Property VI.3 stated that the FDS contains $O(mn)$ points. Hence, a direct evaluation of (VI.3) over all these points is performed in $O(mn^2)$ time. Despite this, in this chapter we present an algorithm that efficiently solves problem (VI.4) in $O(mn)$ time.

VI.3 A new approach

Given an edge $e=(v_s,v_t)\in E$, for all nodes $v_i\in V$, let $d_i=d(v_t,v_i)-d(v_s,v_i)$ be the difference of weighted distances from nodes v_s and v_t to node v_i. Obviously, from (VI.2) it follows that $-l_e\le d_i\le l_e$. Using d_i we have $b_i=(d_i+l_e)/2$. In particular, for $d=-l_e$, we get $b_i=0=v_s$, whereas for $d=l_e$, we obtain $b_i=l_e=v_t$.

We define the following sets

$$A=\{v_i\in V:-l_e<d_i\le l_e\},\quad B=\{v_i\in V:d_i=-l_e\}$$
$$C=\{v_i\in V:-l_e\le d_i<l_e\},\quad D=\{v_i\in V:d_i=l_e\}$$

Note that $B\subseteq C$, $D\subseteq A$ and $A\cup B=C\cup D=V$.

Let $W=\sum_{v_i\in V}w_i$ be the sum of all the weights, and let W_s be the right slope of function $f(x)$ at node v_s, that is

$$W_s=\sum_{v_i\in A}w_i-\sum_{v_i\in B}w_i=W-2\sum_{v_i\in B}w_i=2\sum_{v_i\in A}w_i-W \tag{VI.6}$$

Likewise, let W_t be the opposite sign value of the left slope of $f(x)$ at v_t,

$$W_t=\sum_{v_i\in C}w_i-\sum_{v_i\in D}w_i=2\sum_{v_i\in C}w_i-W=W-2\sum_{v_i\in D}w_i \tag{VI.7}$$

Obviously, $W_s,W_t\le W$. When $W_s\le 0$ or $W_t\le 0$, problem (VI.5) is easily solved using the following result.

Theorem VI.1. *Given an edge $e=(v_s,v_t)\in E$, we get a solution to (VI.5) in the following cases:*

a) *If $W_s=W_t=0$, the solution is the interval $[v_s,v_t]$.*
b) *If $W_s=0$ and $W_t\ne 0$, the solution interval is $[v_s,\min_{b_i\ne 0}b_i]$.*
c) *If $W_t=0$ and $W_s\ne 0$, the solution interval is $[\max_{b_i\ne l_e}b_i,v_t]$.*
d) *If $W_s<0$ and $W_t\ne 0$ the optimum point is v_s.*
e) *If $W_t<0$ and $W_s\ne 0$ the optimum point is v_t.*

Proof. Taking into account Property VI.1 and Property VI.2, the proof is as follows:

a) Since $f(x)$ is concave (Property VI.1) and $W_s=W_t=0$, then $f(x)$ must be constant along edge *e*, and therefore the solution is the interval $[v_s,v_t]$.
b) The case $W_s=0$ and $W_t<0$ is not feasible. In fact, if $W_s=0$ and $W_t<0$ then $W=2\sum_{v_i\in A}w_i<2\sum_{v_i\in D}w_i$, which is a contradiction to $D\subseteq A$. On the other hand, if $W_s=0$ and $W_t>0$, due to Property VI.1 the solution must be the interval $[v_s,\min_{b_i\ne 0}b_i]$ (see Figure VI.2).
c) Analogous to the proof of b), the solution obtained is $[\max_{b_i\ne l_e}b_i,v_t]$.
d) The case $W_s<0$ and $W_t<0$ is not possible, since $2\sum_{v_i\in C}w_i<W<2\sum_{v_i\in B}w_i$ is not true for $B\subseteq C$. On the other hand, if $W_s<0$ and $W_t>0$, as a result of Property VI.1, the solution is v_s.
e) In a similar way to the proof of d), if $W_t<0$ and $W_s\ne 0$ then the solution is v_t. ■

Unfortunately, when W_s and W_t are both strictly positive, problem (VI.5) is not so easy to solve. However, these two values can be used to define a new upper bound, which allows simplifying the search procedure.

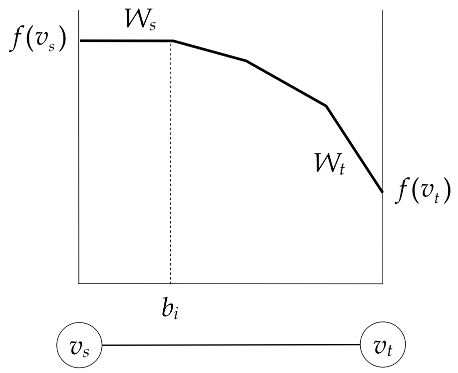


Figure VI.2: Case b) of Theorem VI.1.

VI.4 Lower and upper bounds

On any edge $e=(v_s,v_t)\in E$, a simple lower bound $LB(e)=\max(f(v_s),f(v_t))$ was proposed by Church and Garfinkel (1978). They also gave an upper bound for the unweighted maxian problem, which can be used to derive an initial upper bound for the weighted maxian problem as follows

$$UB(e)=\frac{f(v_s)+f(v_t)+Wl_e}{2} \tag{VI.8}$$

This bound is computed in $O(n)$ time. However, this upper bound can be improved with the same time complexity as follows.

We consider both W_s and W_t to be strictly positive. Now, we compute the intersection point z such that $f(v_s)+zW_s=f(v_t)+W_t(l_e-z)$, and its ordinate value $y(z)$ (see Figure VI.3).

$$z=\frac{f(v_t)-f(v_s)+W_tl_e}{W_s+W_t}, \qquad y(z)=\frac{W_sf(v_t)+W_tf(v_s)+W_sW_tl_e}{W_s+W_t}$$

Let $NUB(e)=y(z)$ be the new upper bound. Obviously, since $f(x)$ is a concave function, $f(x)\le NUB(e), \forall x\in e$. We are going to prove that the new upper bound is at least as good as (VI.8). Thus, first we need to state the following Lemma.

Lemma VI.1. $f(v_t)\le f(v_s)+W_sl_e$.

Proof. Taking into account that $f(v_s)=\sum_{v_i\in V}w_id(v_s,v_i)$ and $f(v_t)=\sum_{v_i\in V}w_id(v_t,v_i)$, then $f(v_t)-f(v_s)=\sum_{v_i\in V}w_i(d(v_t,v_i)-d(v_s,v_i))=\sum_{v_i\in V}w_id_i$. From (VI.6) we get $\sum_{v_i\in A}w_i=\frac{W+W_s}{2}$ and $\sum_{v_i\in B}w_i=\frac{W-W_s}{2}$. Since $A\cup B=V$, then

$$\sum_{v_i\in V}w_id_i=\sum_{v_i\in A}w_id_i+\sum_{v_i\in B}w_id_i=\sum_{v_i\in A}w_id_i+\sum_{v_i\in B}w_i(-l_e)=\sum_{v_i\in A}w_id_i-l_e\left(\frac{W-W_s}{2}\right)$$

On the other hand, $\sum_{v_i \in A} w_i d_i \leq \sum_{v_i \in A} w_i l_e = l_e \sum_{v_i \in A} w_i = l_e \left(\frac{W + W_s}{2} \right)$. Replacing this result in the previous expression we get

$$f(v_t) - f(v_s) = \sum_{v_i \in V} w_i d_i = \sum_{v_i \in A} w_i d_i - l_e \left(\frac{W - W_s}{2} \right) \leq l_e \left(\frac{W + W_s}{2} \right) - l_e \left(\frac{W - W_s}{2} \right) = l_e W_s$$

Therefore, $f(v_t) \leq f(v_s) + W_s l_e$. ■

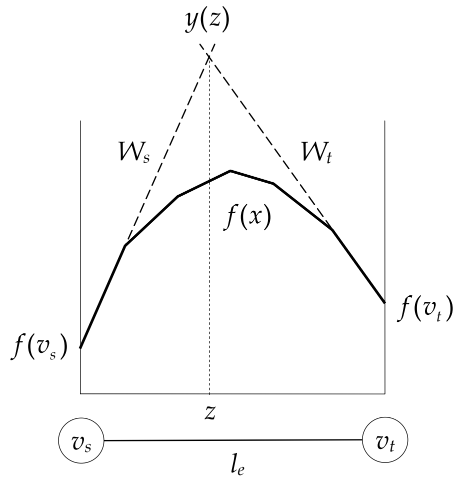


Figure VI.3: $NUB(e) = y(z)$ is the new upper bound.

Proposition VI.1. *For any edge* $e = (v_s, v_t) \in E$, $NUB(e) \leq UB(e)$.

Proof. If $W_s = W_t = W$, then $NUB(e) = \frac{W(f(v_s) + f(v_t) + W l_e)}{2W} = UB(e)$.

When $W_s = W_t < W$, we then get

$$NUB(e) = \frac{W_s(f(v_t) + f(v_s) + W_s l_e)}{2W_s} = \frac{W_t(f(v_t) + f(v_s) + W_t l_e)}{2W_t} < UB(e)$$

In the case of $W_s, W_t \leq W$ and $W_s \neq W_t$, we start from the following expression

$$f(v_s)(W_t - W_s) + f(v_t)(W_s - W_t)$$

By virtue of Lemma VI.1, we have that

$$f(v_s)(W_t - W_s) + f(v_t)(W_s - W_t) \leq f(v_s)(W_t - W_s) + (f(v_s) + W_s l_e)(W_s - W_t) = W_s l_e (W_s - W_t)$$

Adding and subtracting W inside the parentheses of the last term we obtain

$$W_s l_e (W_s - W_t + W - W) = W_s l_e (W - W_t) + W_s l_e (W_s - W)$$

and since $W_s - W \leq 0$, then

$$W_s l_e (W - W_t) + W_s l_e (W_s - W) \leq W_s l_e (W - W_t) \leq W_s l_e (W - W_t) + W_t l_e (W - W_s)$$

and hence

$$f(v_s)(W_t - W_s) + f(v_t)(W_s - W_t) \leq W_s l_e (W - W_t) + W_t l_e (W - W_s)$$

Arranging this expression we get

$$W_s f(v_t) + W_t f(v_s) + 2 W_s W_t l_e \leq W_s f(v_s) + W_t f(v_t) + (W_s + W_t) W l_e$$

$$2(W_s f(v_t) + W_t f(v_s) + W_s W_t l_e) \le (W_s + W_t)(f(v_s) + f(v_t) + W l_e)$$

and therefore, $NUB(e) \le UB(e)$. ■

Despite these two bounds $UB(e)$ and $NUB(e)$ are equal when $W_s = W_t = W$, there is a special case in which we can determine the minimum difference between them. If the distance between the nodes of an edge is equal to its length, the following result can be stated.

Corollary VI.1. *Given any edge* $e = (v_s, v_t) \in E$ *such that* $d(v_s, v_t) = l_e$, *the minimum difference between* $NUB(e)$ *and* $UB(e)$ *is*

$$\frac{(w_s - w_t)(f(v_s) - f(v_t)) + (W(w_s + w_t) - 4w_s w_t)l_e}{2(W - w_s - w_t)}$$

Proof. If edge *e* satisfies $d(v_s, v_t) = l_e$, then $W_s = W - 2w_t$ and $W_t = W - 2w_s$. We only have to replace these two values in $UB(e) - NUB(e)$, and the result follows. ■

The method described in the next section makes use of this new upper bound $NUB(e)$. Moreover, this bound will be updated in each iteration of the search procedure. Hence, we define the new upper bound on edge *e* as a function G_{UB} of five parameters:

$$G_{UB}(e, F_j, W_j, F_k, W_k) = \frac{W_j F_k + W_k F_j + W_j W_k l_e}{W_j + W_k}$$

Thus, $NUB(e) = G_{UB}(e, f(v_s), W_s, f(v_t), W_t)$.

VI.5 The method proposed when W_s and W_t are strictly positive

Church and Garfinkel (1978) devised an $O(mn\log n)$ algorithm to solve the maxian problem. Theorem VI.1 provides directly the solution when W_s or W_t is nonpositive. In this section we show how to obtain the optimal points in $O(mn)$ time when W_s and W_t are strictly positive.

Let $e = (v_s, v_t) \in E$. We begin by replacing (VI.1) in (VI.3) to get

$$f(x) = \sum_{v_i \in V} w_i \min\{x + d(v_s, v_i), l_e - x + d(v_t, v_i)\}$$

Given a point *x* on *e*, the following two sets are defined:

$$L(x) = \{v_i \in V : b_i < x\}, \qquad R(x) = \{v_i \in V : b_i \ge x\}$$

The set $L(x)$ contains the nodes with their bottleneck point b_i lying to the left of *x*, whereas $R(x)$ includes the nodes with their bottleneck point greater or equal to *x*.

Function $f(x)$ is then divided into two summations

$$f(x) = \sum_{L(x)} w_i (l_e - x + d(v_t, v_i)) + \sum_{R(x)} w_i (x + d(v_s, v_i))$$

Arranging the expression we obtain

$$f(x) = \sum_{L(x)} w_i (l_e + d(v_t, v_i)) + \sum_{R(x)} w_i d(v_s, v_i) + x\left(\sum_{R(x)} w_i - \sum_{L(x)} w_i\right) \qquad \text{(VI.9)}$$

Recall that from Property VI.1 function $f(x)$ is continous, concave and piecewise linear, $\sum_{R(x)} w_i - \sum_{L(x)} w_i$ being the different values of the successive slopes of $f(x)$. Let $W_L(x) = \sum_{L(x)} w_i$ and since $\sum_{R(x)} w_i + \sum_{L(x)} w_i = W$ then $\sum_{R(x)} w_i = W - W_L(x)$. Replace this value in (VI.9), and let $H(x)$ be equal to the first two summations,

$$f(x) = H(x) + x(W - 2W_L(x))$$

For any $x \in e$, function $H(x)$ is always positive. The second term will remain positive as long as $2W_L(x) < W$. This means that $f(x)$ is growing to the right of point x. Once $2W_L(x) = W$ function $f(x)$ has reached its maximum value with a null slope (see Property VI.2). Finally, function $f(x)$ is decreasing when $2W_L(x) > W$.

Following this scheme, we could evaluate $W - 2W_L(x)$ at several particular points x to check whether $f(x)$ is increasing, decreasing or remains flat. The points to evaluate are the set of edge bottleneck points B_e.

Let $l = 1$ and $r = n$ be the lowest and highest indices in B_e, respectively. Let d_q be the median value of all the differences d_i ($l \le i \le r$), that is, the value for which half of the values are smaller or equal, and the other half are greater or equal. This can be computed in $O(n)$ time using Hoare (1961). This algorithm performs a permutation of the elements in B_e such that $d_l, \ldots, d_{q-1}$ are smaller or equal to d_q, and $d_{q+1}, \ldots, d_r$ are greater or equal. Let b_q and w_q be, respectively, the bottleneck point and the weight related to d_q.

Let $W_L(b_q) = \sum_{L(b_q)} w_i$. Since $b_i < b_q$ for $l \le i < q$, we can set $W_L = W_L(b_q) = \sum_{i=l}^{q-1} w_i$. Likewise, let $W_R = \sum_{i=q+1}^{r} w_i = W - W_L - w_q$. Note that the left slope of point b_q is $(W_R + w_q) - W_L$, whereas the right slope is $W_R - (W_L + w_q)$. Following the analysis given above, the next result is achieved.

Theorem VI.2. *There exists a solution to (VI.5) in the next three cases:*

a) *If $W_L + w_q = W_R$, then the solution is $[b_q, \min_{q<i\le r} b_i]$.*

b) *If $W_L = W_R + w_q$, then the solution is $[\max_{l\le i<q} b_i, b_q]$.*

c) *If $W_R - w_q < W_L < W_R + w_q$, then the solution is point b_q.*

Proof.

a) $W_L + w_q = W_R \Rightarrow W_L + w_q = W - W_L - w_q \Rightarrow 2(W_L + w_q) = W \Rightarrow 2\sum_{i=l}^{q} w_i = W$.

This result implies that $f(x)$ has attained its maximum at point b_q with a null slope to the following point (Property VI.2). Thus, the solution is the interval $[b_q, \min_{q<i\le r} b_i]$.

b) $W_L = W_R + w_q \Rightarrow W_L = W - W_L - w_q + w_q \Rightarrow 2W_L = W \Rightarrow 2\sum_{i=l}^{q-1} w_i = W$.

This is analogous to the preceding case, but shifted one place to the left. Then the solution is $[\max_{l\le i<q} b_i, b_q]$.

c) $W_L < W_R + w_q \Rightarrow 2W_L < W$, so $f(x)$ is increasing at point b_{q-1}.

If $W_L + w_q > W_R \Rightarrow 2(W_L + w_q) > W$, then $f(x)$ is decreasing at b_q. Therefore, $f(b_q)$ is the maximum value of $f(x)$. ■

These three cases are mutually exclusive, though there are still two more alternatives in which the solution is not achieved, since the maximal value is not attained at point b_q. In this case, we must move either to the right (case d) or to the left (case e) to search for the optimum.

d) $W_L + w_q < W_R$: function $f(x)$ is increasing at point b_q. It follows that the maximum must be to the right of this point. All points b_i such that $l \le i \le q$ can be discarded. The search resumes with $l = q+1$.

e) $W_L > W_R + w_q$: implies that $f(x)$ is decreasing and, therefore, all points b_i with $q \le i \le r$ can be removed. The search continues with $r = q-1$.

The next section outlines the new algorithm considering Theorem VI.1 and Theorem VI.2, and the previous cases d) and e). Besides, we introduce an improvement by dynamically updating the new upper bound $NUB(e)$ over point b_q in each iteration. In this way, the search process can be finished as soon as the value of $NUB(e)$ is less than the network optimum stored thus far.

VI.6 The new algorithm

In this section, we bring together all the results previously stated. First, we outline the new algorithm in Algorithm VI.1, and then we prove its complexity. Finally, the unweighted case is analyzed.

The dynamic calculation of the new bound using point b_q is performed by function $G_{UB}(e, F_j, W_j, F_k, W_k)$. The values F_j and F_k depend on $f(b_q)$, which can be obtained from (VI.9). Replacing $f_L(b_q) = \sum_{L(b_q)} w_i d(v_t, v_i)$ and $f_R(b_q) = \sum_{R(b_q)} w_i d(v_s, v_i)$ we get

$$f(b_q) = f_L(b_q) + f_R(b_q) + b_q(W - W_L(b_q)) + (l_e - b_q)W_L(b_q)$$

If a new median is computed in the next iteration, say for example d_p with bottleneck point b_p, then the value of $f(b_p)$ can be determined from $f(b_q)$ in the following way:

- If $b_p < b_q$, then $f_L(b_p) = f_L(b_q) - \sum_{i=p}^{r} w_i d(v_t, v_i)$ and $f_R(b_p) = f_R(b_q) + \sum_{i=p}^{r} w_i d(v_s, v_i)$, so $$f_L(b_p) + f_R(b_p) = f_L(b_q) + f_R(b_q) + \sum_{i=p}^{r} w_i (d(v_s, v_i) - d(v_t, v_i)) = f_L(b_q) + f_R(b_q) - \sum_{i=p}^{r} w_i d_i .$$
- If $b_p > b_q$, then $f_L(b_p) = f_L(b_q) + \sum_{i=l}^{p-1} w_i d(v_t, v_i)$ and $f_R(b_p) = f_R(b_q) - \sum_{i=l}^{p-1} w_i d(v_s, v_i)$, so $$f_L(b_p) + f_R(b_p) = f_L(b_q) + f_R(b_q) + \sum_{i=l}^{p-1} w_i (d(v_t, v_i) - d(v_s, v_i)) = f_L(b_q) + f_R(b_q) + \sum_{i=l}^{p-1} w_i d_i .$$

The computation of $W_L(b_p)$ is figured out using the same approach:

$$W_L(b_p) = W_L(b_q) + \begin{cases} -\sum_{i=p}^{r} w_i, & \text{if } b_p < b_q \\ \sum_{i=l}^{p-1} w_i, & \text{if } b_p > b_q \end{cases}$$

```
function NewAlgorithm(Network N, Distance Matrix d)
{   // f_N : Current best value on network N.
    f_N := 0
    // Solution set.
    S := ∅
    for all edges e := (v_s, v_t) ∈ E do
      {  Compute W_s and W_t by (VI.6) and (VI.7)
         X_e := ∅   // Let X_e represent either a single point x or an interval [x^1, x^2].
         if Theorem VI.1 holds then Store solution in X_e
         else
           {  F_j := f(v_s), W_j := W_s
              F_k := f(v_t), W_k := W_t
              // Compute initial value of the new upper bound.
              NUB(e) := G_UB(e, F_j, W_j, F_k, W_k)
              if NUB(e) < f_N then continue to next edge
              l := 1, r := n
              while X_e = ∅ and NUB(e) ≥ f_N do
                {  d_q := Median value of all d_i with l ≤ i ≤ r
                   b_q := (d_q + l_e)/2
                   Compute W_L and W_R
                   if cases a), b) or c) of Theorem VI.2 hold then Store solution in X_e
                   else
                     {  // Search for the optimum to the left or right, cases d) or e).
                        if case d) holds then   l := q+1, update F_j, W_j, W_L, f(b_q)
                        else                    r := q-1, update F_k, W_k
                        // Update the upper bound at point b_q
                        NUB(e) := G_UB(e, F_j, W_j, F_k, W_k)
                     }
                }
           }
         if X_e ≠ ∅ and f(X_e) ≥ f_N then
           {  f_N := f(X_e)
              Store the pair (X_e, e) in S
           }
      }
    return (f_N, S)
}
```

Algorithm VI.1: The new algorithm for the maxisum problem.

Finally, when cases d) or e) are satisfied, the values F_j, W_j and F_k, W_k must be update accordingly:

- If case d) is fulfilled, update $W_j = W - 2(W_L(b_q) + w_q)$ and $F_j = f(b_q) - W_j b_q$. Besides, since we move to the right, we must set $W_L = W_L + w_q$ and $f(b_q) = f(b_q) + w_q d_q$.

- Else, update $W_k = 2W_L(b_q) - W$ and $F_k = f(b_q) - W_k(l_e - b_q)$, leaving W_L and $f(b_q)$ unchanged.

Likewise, the values of l and r are also updated, and thus, we can delete half of the values d_i. This result proves the following Lemma.

Lemma VI.2. *In each iteration of the while loop, $q = (l+r)/2$ points are removed from B_e.*

This Lemma helps in proving the overall complexity of the new algorithm.

Theorem VI.3. *Provided that the distance matrix is given, the new algorithm solves the undesirable 1-median problem on networks in $\mathrm{O}(mn)$ time.*

Proof. For each edge, the initial new bound $NUB(e)$ is computed in $\mathrm{O}(n)$ time. According to the preceding Lemma, in each iteration of the 'while' loop, the size of B_e is reduced to a half. Thus, the number of points processed is

$$n + \frac{n}{2} + \frac{n}{4} + \cdots + \frac{n}{2^k} = n\left(\frac{2^k + 2^{k-1} + \cdots + 1}{2^k}\right) = \frac{n}{2^k}\sum_{i=0}^{k} 2^i = \frac{n}{2^k}(2^{k+1} - 1)$$

The loop keeps on, in the worst case, till l and r are consecutive. Then, $n/2^k = 1 \Rightarrow n = 2^k$, and consequently, $(n/2^k)(2^{k+1} - 1) = 2n - 1 \in \mathrm{O}(n)$. This must be applied to all m edges. Thus, the overall complexity is $\mathrm{O}(mn)$. ■

The algorithm described works on networks with weighted vertices. However, sometimes the network might have all node weights equal. The next section analyzes this particular situation.

VI.6.1 The unweighted case

When all nodes v_i have the same weight $w_i = w$, the underlying network can be considered unweighted. In this case, Church and Garfinkel (1978) suggested a method in $\mathrm{O}(mn\log n)$ time to obtain the optimal point.

The following result states that the new algorithm directly solves the unweighted case in $\mathrm{O}(mn)$ time as well.

Proposition VI.2. *If all weights w_i, $\forall v_i \in V$ are equal, then either Theorem VI.1 holds, or only cases b) or c) of Theorem VI.2 are fulfilled in the first iteration of the 'while' loop.*

Proof. Since all weights are equal, we can assume $w_i = w$, for all nodes v_i. Thus, $w_q = w$ and $W = \sum_{v_i \in V} w_i = nw$. Besides, $W_L = \lfloor nw/2 \rfloor$ and $W_R = wn - \lfloor wn/2 \rfloor - w = \lceil wn/2 \rceil - w$.

If Theorem VI.1 is held, we promptly obtain the solution. With regards to Theorem VI.2, we check all possible choices:

a) Case $W_L + w_q = W_R$ is not possible, since $\lfloor wn/2 \rfloor + w \neq \lceil wn/2 \rceil - w$.

b) If $W_L = W_R + w_q$ then $\lfloor wn/2 \rfloor = \lceil wn/2 \rceil - w + w$, which is true if n is even.

c) If $W_L + w_q > W_R$ then $\lfloor wn/2 \rfloor + w > \lceil wn/2 \rceil - w$, and if $W_L < W_R + w_q$ then $\lfloor wn/2 \rfloor < \lceil wn/2 \rceil - w + w$. Both are true for n odd.

d) Case $W_L + w_q < W_R$ is not feasible, because always $\lfloor wn/2 \rfloor + w \geq \lceil wn/2 \rceil - w$.

e) Case $W_L > W_R + w_q$ is not possible, since $\lfloor wn/2 \rfloor \leq \lceil wn/2 \rceil - w + w$.

Therefore, if n is even then case b) is fulfilled. Otherwise, case c) is satisfied. As neither case d) nor e) are true, the 'while' loop iterates once, and the solution is hence obtained in $O(mn)$ time.■

Before presenting the computational experience, next we show a small example to illustrate how the new algorithm runs.

VI.7 An example

Consider the network in Figure VI.4, with $n = 7$ nodes and $m = 15$ edges. The node weights (in bold) are integers randomly generated between 1 and 9, whereas the edge lengths range between 1 and 25. The total weight *W* is equal to 24. Table VI.1 summarizes the trace.

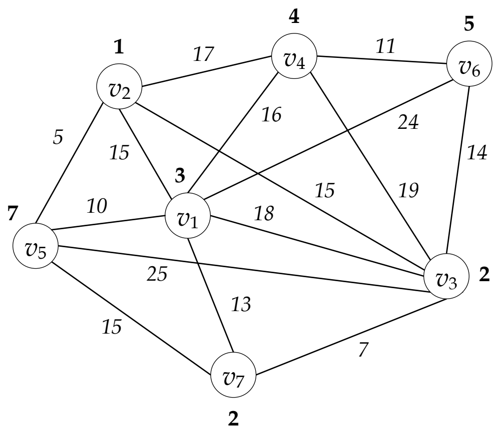


Figure VI.4: Weighted network with seven nodes and fifteen edges.

We begin with edge (v_1, v_3). The slopes of function $f(x)$ at nodes v_s and v_t are, respectively, $W_s = 20$ and $W_t = 18$. Thus, the new upper bound is $NUB(e) = 521.526$. We have also included in the table the upper bound $UB(e)$ and the difference between this bound and the new one. Since the current best value f_N on the network is equal to zero, we proceed to search for the optimum value inside the edge. In the first iteration of the loop, case d) is satisfied, which sets $f(b_q) = 449$. We now move to the right with $l = q + 1 = 5$.

In the second iteration, case e) is fulfilled with $f(b_q) = 441$. Then, we go to the left, $r = q - 1 = 5$. Since case c) of Theorem VI.2 holds, the loop ends with the solution on edge (v_1, v_3) set to $(X_e, f(X_e)) = (10.5, 455)$. The network optimum f_N and the solution *S* are also updated accordingly. The value of $NUB(e) = 482.059$ at (v_1, v_4) is greater than $f_N = 455$. Despite case a) of Theorem VI.2 is held, the solution $f(b_q) = 398$ is less than f_N, which remains unchanged.

In the following edge, the value of $NUB(e) = 394.692$ is less than f_N, and therefore, the edge is not processed. This is not the case for the upper bound $UB(e) = 464.5$, which would have made the search process to be run.

Case c) of Theorem VI.2 is held in edge (v_1, v_6), updating the network solution to $f_N = 479$ and $S = \{(v_1, v_6), 16\}$. The next two edges (v_1, v_7) and (v_2, v_1) cannot improve this optimum. Likewise, the two following edges (v_2, v_3) and (v_2, v_4) yield an upper bound $NUB(e)$ smaller than f_N. Again, the bound $UB(e)$ of these two edges would have led them to be uselessly processed.

Edge (v_2, v_5) generates a peculiar instance. Both W_s and W_t are zero, and thus case a) of Theorem VI.1 is satisfied. However, the value $f(X_e) = 358$ is less than $f_N = 479$. The network solution is updated to $f_N = 500$ and $S = \{(v_3, v_4), [8.5, 10.5]\}$ in the next edge. None of the five subsequent edges, (v_3, v_5) through (v_5, v_7), can improve f_N. Note that the value of the new upper bound $NUB(e)$ in (v_4, v_6) and (v_5, v_7) allows avoiding, once again, the search process.

The optimum value of $f(x)$ in this example is $f_N = 500$, which is attained in the interval $[8.5, 10.5]$ at edge (v_3, v_4). We finally remark that, due to the new upper bound $NUB(e)$, we have only processed 8 of the 15 total edges. Using $UB(e)$ the algorithm would have run over 13 edges. This enhancement allows a substantial saving of time with regard to Church and Garfinkel's algorithm, as we show in the next section.

VI.8 Computational results

The computational experience was performed using C++ programming language (GNU g++ 2.95.2) and LEDA (*Library of Efficient Datatypes and Algorithms*, see Melhorn and Näher, 1999), on a DEC with four alpha 466 Mhz processors and 2 Gb of RAM, running OSF Digital UNIX.

Despite Tamir (1991) having briefly stated that a solution to the network 1-maxisum problem can be obtained in $O(mn)$ time using the general algorithms proposed by Zemel (1984), the procedure is not directly described. As a result of this, we decided to compare the new algorithm with the one proposed by Church and Garfinkel (1978), which was accurately programmed following the results and the original procedure given in their paper. Besides, we added the weighted version $UB(e)$ of their original unweighted upper bound to the algorithm to make the comparison as fair as possible. The distance matrix was computed using an $O(mn + n^2 \log n)$ algorithm devised in LEDA. We remark that the time spent in calculating this matrix is not included in the total computing time of the algorithms.

Two types of experiment were performed. In both of them the node weights are random integers between 1 and 10, whereas the edge lengths vary from 1 to 50. In the first test, random networks were generated with $n = 100$ up to 1000 nodes with a step of 50, and different edge densities $d = 1/2$, 1/4 and 1/8, being $d = m/(n(n-1)/2)$. These networks are complete graphs with, respectively, a half, a quarter and an eighth of the total number of edges. The second experiment generated planar networks with $m = 3n - 6$ edges and $n = 1000$ up to 8000 in steps of 500 nodes. In all cases, ten instances of each network were created using the LEDA random graph generators. Label "C & G" stands for Church and Garfinkel.

Figure VI.5 shows the average time results and processed edges for the two algorithms when density d is equal to 1/2, 1/4 and 1/8. In the three cases, the new algorithm is much faster than Church and Garfinkel's. The same happens in the first graphic of Figure VI.6 for planar networks. Finally, Table VI.2 presents the reduction percentage in the number of edges and the time gained with the new algorithm for the planar networks generated in the second experiment.

Edge	*Upper bounds*					*Search process*								*Edge solution*			*Network solution*	
e	W_s	W_t	$UB(e)$	$NUB(e)$	$UB(e) - NUB(e)$	l	r	q	W_L	W_R	*Case*	b_q	$f(b_q)$	X_e	$f(X_e)$		f_N	S
				521.526	44.474	1	7	4	9	14	d)	9	449	$\varnothing$	–			
(v_1,v_3)	20	18	566	463.909	102.091	5	7	6	14	3	e)	14	441	$\varnothing$	–			
				455	111	5	5	5	10	10	Th. 2, c)	10.5	455	10.5	455	>	0	$\varnothing$
(v_1,v_4)	16	18	539.5	482.059	57.441	1	7	4	11	12	Th. 2, a)	9	398	[9,14]	398	<	455	$\{(v_1,v_3),10.5\}$
(v_1,v_5)	8	18	464.5	394.692	69.808	–	–	–	–	–	–	–	–	–	–		455	$\{(v_1,v_3),10.5\}$
(v_1,v_6)	14	18	676	569.875	106.125	1	7	4	11	11	Th. 2, c)	16	479	16	479	>	455	$\{(v_1,v_3),10.5\}$
(v_1,v_7)	20	18	515	483.632	31.368	1	7	4	9	8	Th. 2, c)	9	439	9	439	<	479	$\{(v_1,v_6),16\}$
(v_2,v_1)	18	22	524.5	494.35	30.15	1	7	4	10	10	Th. 2, c)	7	457	7	457	<	479	$\{(v_1,v_6),16\}$
(v_2,v_3)	20	8	543.5	451.571	91.929	–	–	–	–	–	–	–	–	–	–		479	$\{(v_1,v_6),16\}$
(v_2,v_4)	6	8	565	418.857	146.143	–	–	–	–	–	–	–	–	–	–		479	$\{(v_1,v_6),16\}$
(v_2,v_5)	0	0	418	358	60	–	–	–	–	–	Th. 1, a)	v_2	358	$[v_2, v_5]$	358	<	479	$\{(v_1,v_6),16\}$
(v_3,v_4)	16	16	594.5	518.5	76	1	7	4	12	5	Th. 2, b)	10.5	500	[8.5,10.5]	500	>	479	$\{(v_1,v_6),16\}$
(v_3,v_5)	24	24	663.5	663.5	0	1	7	4	11	9	Th. 2, c)	14	498	14	498	<	500	$\{(v_3,v_4), [8.5,10.5]\}$
(v_3,v_6)	14	16	575	509	66	1	7	4	12	5	Th. 2, b)	13.5	453	[10,13.5]	453	<	500	$\{(v_3,v_4), [8.5,10.5]\}$
(v_3,v_7)	20	2	462	398.091	63.909	–	–	–	–	–	–	–	–	–	–		500	$\{(v_3,v_4), [8.5,10.5]\}$
(v_4,v_6)	14	0	536.5	445	91.5	–	–	–	–	–	Th. 1, c)	v_6	445	v_6	445	<	500	$\{(v_3,v_4), [8.5,10.5]\}$
(v_5,v_7)	20	8	552.5	464.429	88.071	–	–	–	–	–	–	–	–	–	–		500	$\{(v_3,v_4), [8.5,10.5]\}$

Table VI.1: Trace of the new algorithm on the network of Figure VI.4.

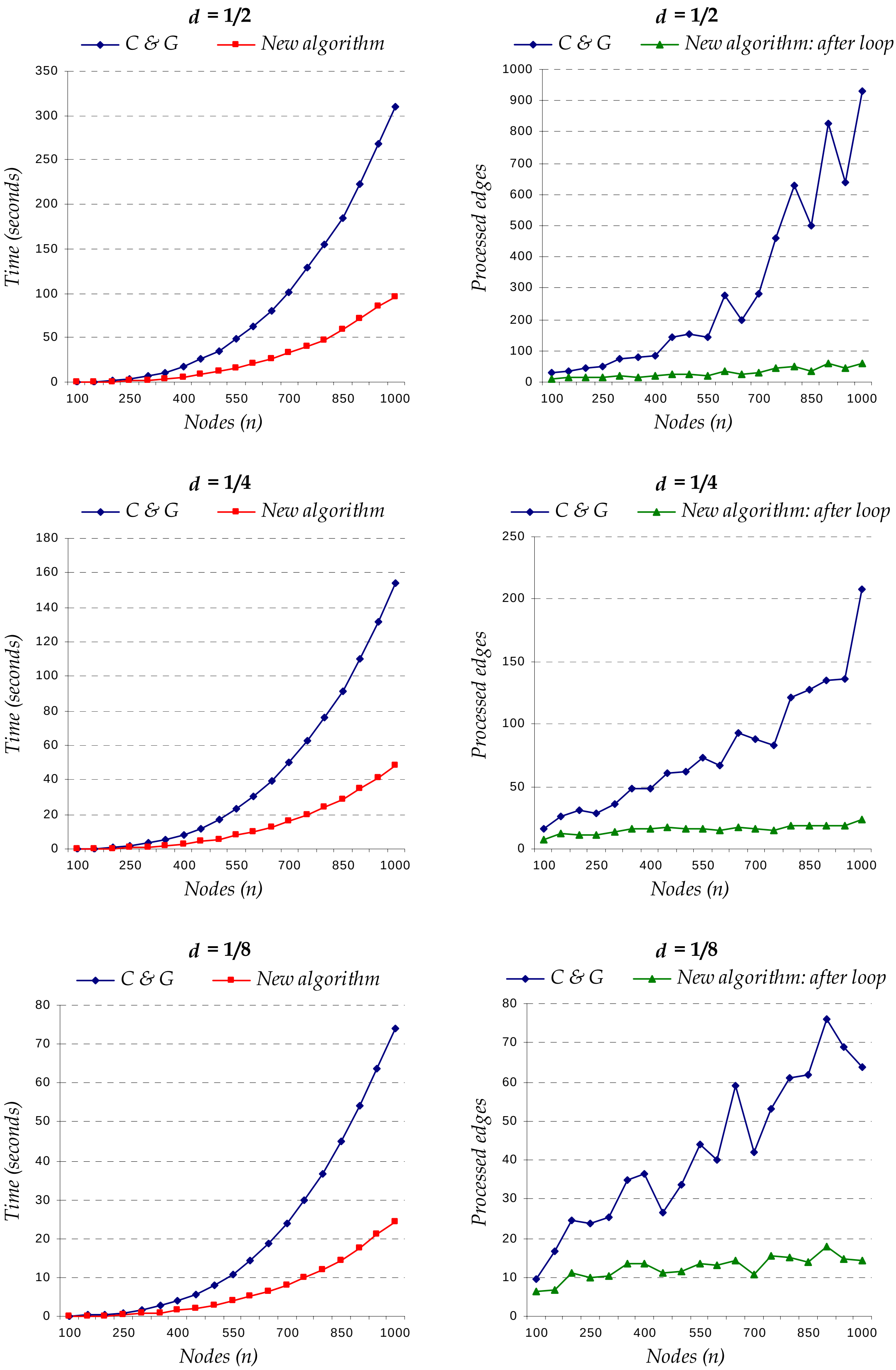


Figure VI.5: Time results and processed edges for networks with $n = 100$ to 1000 nodes and density d equal to 1/8, 1/4 and 1/2.

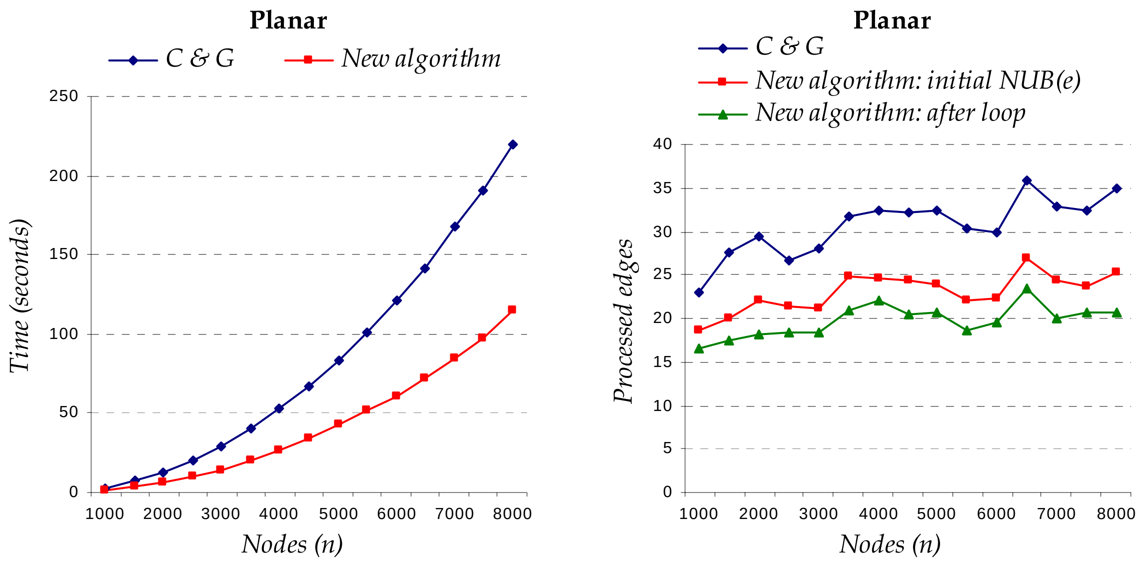


Figure VI.6: Time results and edges processed for planar networks with $n = 1000$ to 8000 nodes.

Nodes	*Time*			*Processed edges*		
n	*C & G*	*New algorithm*	*Reduction %*	*C & G*	*New algorithm: initial NUB(e)*	*Reduction %*
1000	3.05	1.50	50.76	22.9	18.7	18.34
1500	7.06	3.40	51.75	27.5	20	27.27
2000	12.93	6.26	51.59	29.5	22.1	25.08
2500	20.27	10.14	49.95	26.7	21.3	20.22
3000	29.51	14.49	50.87	28.1	21.1	24.91
3500	40.35	20.34	49.58	31.8	24.8	22.01
4000	52.83	26.22	50.38	32.3	24.6	23.84
4500	67.33	34.11	49.33	32.2	24.3	24.53
5000	83.64	42.66	48.99	32.5	23.8	26.77
5500	100.88	52.33	48.12	30.3	22.1	27.06
6000	121.58	61.21	49.65	30	22.3	25.67
6500	141.12	71.81	49.11	35.9	26.9	25.07
7000	167.63	84.32	49.70	32.8	24.3	25.91
7500	191.23	96.89	49.33	32.5	23.6	27.38
8000	219.19	114.65	47.69	34.9	25.2	27.79

Table VI.2: Reduction percentage in time and in number of edges for planar networks with $n = 1000$ to 8000.

Regarding the number of edges, the graphics to the right of Figure VI.5 show the processed edges for the three values of density *d*. For each edge, the initial values of the upper bounds $UB(e)$ and $NUB(e)$ are almost the same. This means that, in the beginning, they override the same number of edges. We have also plotted the number of edges completely processed by the new algorithm, that is, the number of edges for which the dynamic update of $NUB(e)$ could not skip out of the 'while' loop. The difference between these two last values denotes the number of edges that were rejected in the 'while' loop due to the update of $NUB(e)$.

Likewise, in Figure VI.6 we show the processed edges for planar networks. The label "initial $NUB(e)$" denotes the number of edges processed by the new algorithm when the initial upper bound is applied. In this case, the initial value of $NUB(e)$ is much better than $UB(e)$, and hence, the new algorithm discards more edges than Church and Garfinkel's procedure. On the other hand, label "after loop" stands for the number of edges totally processed by the new algorithm. The difference between the values of "after loop" and "initial $NUB(e)$" represents the number of edges discarded by the 'while' loop applying the dynamic computation of the new bound $NUB(e)$.

VI.9 Combining the uncenter with the maxian: an improved algorithm for the anti-cent-dian problem

In previous sections we have addressed the 1-uncenter (maximin) problem and the 1-maxian (maxisum) problem on networks. Now we are going to combine these two objectives to obtain a location criterion called the *anti-cent-dian*.

The network anti-cent-dian model considers the convex combination of the maximin and the maxisum criteria. Moreno-Pérez and Rodríguez-Martín (1999) developed two algorithms that provide, respectively, the optimal location for a fixed λ that determines the convex combination, and the set of optimal locations for all convex combinations. Both of them run in $O(mn\log n)$ time. In the following sections we show that the complexity of the first algorithm can be reduced to $O(mn)$. According to the classification scheme introduced in Chapter I, this problem is denoted as $1/\mathcal{G}/\bullet/d(\mathcal{V},\mathcal{G})/\mathrm{CD}_{\mathrm{obnox}}$.

VI.9.1 Notation and properties

Let $N=(V,E)$ be a simple, undirected and connected network with n nodes (vertices) $V=\{v_1,v_2,\ldots,v_n\}$, and m edges $E=\{(v_s,v_t):v_s,v_t\in V\}$, with $|E|=m$. For the sake of simplicity, we follow the same notation introduced in section VI.2 for the node weight function w and the edge length function l, as well as for the distance $d(\cdot,\cdot)$ and the bottleneck points b_i.

Let $B_e=\bigcup_{v_i\in V} b_i$ be the set of all bottleneck points on edge e, and let $B_N=\bigcup_{e\in E} B_e$ be the set of all bottleneck points on network N.

Let Q_e be a set containing points $x\in e$ such that, for any two distinct nodes $v_i,v_j\in V$, $d(x,v_i)=d(x,v_j)$ and besides, $d(x,v_i)$ and $d(x,v_j)$ do not both decrease when x is perturbed slightly in either direction. Let $Q_N=\bigcup_{e\in E} Q_e$.

We now define the unweighted uncenter (maximin) function and the maxian (maxisum) function. Given any point x on network N, we define

$$f_{\min}(x)=\min_{v_i\in V} d(x,v_i)$$

as the minimum unweighted distance from point x to all nodes of the network. Recall that a point $y_N\in N$ is an uncenter point iff $f_{\min}(y_N)=\max_{x\in N} f_{\min}(x)$. When all node weights w_i are equal, the point y_N is located in the middle of the longest edge. Then, the uncenter point for any edge $e=(v_s,v_t)$ is $y_e=l_e/2$, and hence $f_{\min}(y_e)=l_e/2$. Thus, the local optimum can be obtained in $O(1)$.

On the other hand, given $W=\sum_{v_i\in V} w_i$ and a point $x\in N$, we now define

$$f_{\text{sum}}(x)=\frac{1}{W}\sum_{v_i\in V} w_i d(x,v_i)$$

as the average sum of weighted distances from point *x* to all the nodes of the network. A point $z_N\in N$ is a maxian point iff $f_{sum}(z_N)=\max_{x\in N} f_{sum}(x)$. The local maxian point on edge *e* is denoted by z_e.

Finally, the *anti-cent-dian* function is defined as

$$f_{\text{acd}}(\lambda,x)=\lambda f_{\min}(x)+(1-\lambda)f_{\text{sum}}(x) \tag{VI.10}$$

and any point $x_N\in N$ maximizing $f_{\text{acd}}(\lambda,x)$ for a particular value of λ, $0\le\lambda\le 1$, is called a *λ-anti-cent-dian* point. In particular, if $\lambda=0$, the anti-cent-dian is equal to the maxian; whereas for $\lambda=1$, we obtain the uncenter. Figure VI.7 shows a typical plot of function $f_{\text{acd}}(\lambda,x)$ over edge *e*. For $\lambda=0$ the anti-cent-dian function is $f_{\text{sum}}(x)$. As parameter λ grows, the anti-cent-dian function makes a *morphing* to the $f_{\min}(x)$ function.

Combining the properties of the uncenter and maxian problems we get the following properties originally stated in Moreno-Pérez and Rodríguez-Martín (1999).

Property VI.4. *Given an edge $e=(v_s,v_t)\in E$ and a value λ, $0\le\lambda\le 1$, for any point $x\in e$ the objective function $f_{\text{acd}}(\lambda,x)$ is a continuous, concave and piecewise linear function,*

a) *having a finite number of breakpoints, all belonging to $B_e\cup y_e$,*
b) *with a finite number of locally maximum values, all attained at the points belonging to the set $A=\{v_s,v_t\}\cup B_e\cup y_e$,*
c) *having value zero at the ends of the edge for $\lambda=1$, and*
d) $f_{\text{acd}}(\lambda,v_s)=(1-\lambda)f_{\text{sum}}(v_s)$ *and* $f_{\text{acd}}(\lambda,v_t)=(1-\lambda)f_{\text{sum}}(v_t)$.

Property VI.5. *Given a value of λ, $0\le\lambda\le 1$, there exists at least one point $x_e\in A=\{v_s,v_t\}\cup B_e\cup y_e$ on each edge $e=(v_s,v_t)\in E$ that maximizes $f_{\text{acd}}(\lambda,x)$. If function $f_{\text{acd}}(\lambda,x)$ reaches its maximum value at two consecutive points $a,b\in A$, then all points inside $[a,b]$ maximize function $f_{\text{acd}}(\lambda,x)$.*

Property VI.6. *The Finite Dominating Set (FDS) for the anti-cent-dian problem is $V\bigcup B_N\bigcup Q_N$.*

Property VI.7. *There exist two finite sets $\{\lambda_i\in[0,1]:i=0,\dots,r\}$ and $\{x_i\in N:i=0,\dots,r\}$ with the following properties:*

a) $\lambda_i<\lambda_{i+1}$ *for all i, with* $\lambda_0=0$ *and* $\lambda_r=1$.
b) x_1 *is a maxian point and* x_r *is an uncenter point.*
c) $f_{\text{acd}}(\lambda)=f_{\text{acd}}(\lambda,x_{i+1})$ *for* $\lambda_r\le\lambda\le\lambda_{r+1}$, $i=0,\dots,r-1$.

Since the maximin and maxisum objective functions are both concave, we can derive a new property concerning the set of candidate points inside an edge.

Property VI.8. *Let $e=(v_s,v_t)\in E$, y_e the uncenter point on edge e and $[a,b]$ the maxian points. Given a value of λ, $0\le\lambda\le 1$, the anti-cent-dian points are inside the interval $[\min(y_e,a),\max(y_e,b)]$.*

Proof. When $\lambda = 0$ and $\lambda = 1$, the anti-cent-dian points are, respectively, $[a,b]$ and y_e. Due to the convex combination of the two concave functions $f_{\text{min}}(x)$ and $f_{\text{sum}}(x)$, for any other value of λ the anti-cent-dian point must be attained at a point between y_e and $[a,b]$. Thus, this point is inside the interval $[\min(y_e, a), \max(y_e, b)]$. ■

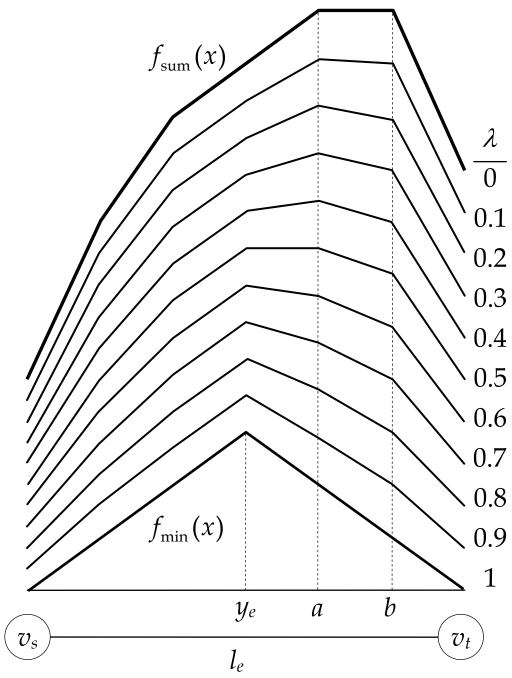


Figure VI.7: Plots of $f_{\text{acd}}(\lambda, x)$ for different values of λ.

Taking into account the first two properties and given a value of λ, $0 \le \lambda \le 1$, problem (VI.10) can be formulated over each edge *e* as follows:

$$f_{\text{acd}}(\lambda, x_e) = \max_{x \in e} f_{\text{acd}}(\lambda, x) \qquad \text{(VI.11)}$$

and a point $x_N \in N$ is a λ-anti-cent-dian point iff $f_{\text{acd}}(\lambda, x_N) = \max_{e \in E} f(\lambda, x_e)$.

A method to determine all λ-anti-cent-dian points for any value of $\lambda \in [0,1]$ in $O(mn \log n)$ time was proposed by Moreno-Pérez and Rodríguez-Martín (1999). It derives from an $O(mn \log n)$ algorithm by Hansen, Labbé and Thisse (1991). This complexity cannot be reduced since the algorithm is based on the computation of a convex hull of $O(mn)$ points, which is done in $O(mn \log n)$ time (see Hershberger, 1989).

On the other hand, Moreno-Pérez and Rodríguez-Martín (1999) also presented an $O(mn \log n)$ procedure to obtain the anti-cent-dian point when λ is fixed to a particular value. Nevertheless, we can achieve an $O(mn)$ time algorithm, as shown in section VI.9.4. Previous to this, in the next section we analyze some properties of the problem and we determine a new upper bound.

VI.9.2 Problem analysis and new upper bound

Let $e = (v_s, v_t) \in E$ be an edge. In the following paragraphs, we characterize the solution to (VI.11). Obviously, when $\lambda = 1$, the solution to (VI.11) is $x_e = y_e$, and y_e is located at the middle of edge *e*. On the other hand, if $\lambda = 0$ then $x_e = z_e$. In section VI.6 we have already proposed an

$O(mn)$ algorithm to determine z_e. Therefore, the analysis will be now focused in the case when $0 < \lambda < 1$.

Following a similar development to section VI.3, for all nodes $v_i \in V$, we consider the differences $d_i = d(v_t, v_i) - d(v_s, v_i)$ and the following sets

$$A = \{v_i \in V : -l_e < d_i \le l_e\} \quad B = \{v_i \in V : d_i = -l_e\}$$
$$C = \{v_i \in V : -l_e \le d_i < l_e\} \quad D = \{v_i \in V : d_i = l_e\}$$

with $B \subseteq C$, $D \subseteq A$ and $A \cup B = C \cup D = V$.

Let W_s be the right slope of function $f_{\text{sum}}(x)$ at node v_s, and let W_t be the opposite sign value of the left slope of $f_{\text{sum}}(x)$ at v_t, that is,

$$\begin{aligned} W_s &= \sum_{v_i \in A} w_i - \sum_{v_i \in B} w_i = W - 2\sum_{v_i \in B} w_i = 2\sum_{v_i \in A} w_i - W \\ W_t &= \sum_{v_i \in C} w_i - \sum_{v_i \in D} w_i = 2\sum_{v_i \in C} w_i - W = W - 2\sum_{v_i \in D} w_i \end{aligned} \quad \text{(VI.12)}$$

Since the anti-cent-dian function is a convex combination of functions $f_{\min}(x)$ and $f_{\text{sum}}(x)$, the right and left slopes of $f_{\text{acd}}(\lambda, x)$ at nodes v_s and v_t should be, respectively, as follows

$$W_s' = \lambda + (1-\lambda)\frac{W_s}{W}, \qquad W_t' = \lambda + (1-\lambda)\frac{W_t}{W} \quad \text{(VI.13)}$$

Since $W_s, W_t \le W$, then $W_s', W_t' \le 1$. If $W_s' \le 0$ or $W_t' \le 0$, problem (VI.11) is easily solved using the following result.

Theorem VI.4. *Given a value of* λ, $0 \le \lambda \le 1$, *and given an edge* $e = (v_s, v_t) \in E$, *we get a solution to* (VI.11) *in the following cases:*

a) *If* $\lambda = W_s / (W_s - W) = W_t / (W_t - W)$, *the solution is the interval* $[v_s, v_t]$.
b) *If* $\lambda = W_s / (W_s - W) \ne W_t / (W_t - W)$, *the solution interval is* $[v_s, \min\{y_e, \min_{b_i \ne 0} b_i\}]$.
c) *If* $\lambda = W_t / (W_t - W) \ne W_s / (W_s - W)$, *the solution interval is* $[\max\{y_e, \max_{b_i \ne l_e} b_i\}, v_t]$.
d) *If* $\lambda > W_s / (W_s - W)$ *and* $\lambda \ne W_t / (W_t - W)$ *the optimum point is* v_s.
e) *If* $\lambda > W_t / (W_t - W)$ *and* $\lambda \ne W_s / (W_s - W)$ *the optimum point is* v_t.

Proof. Taking into account Property VI.4 and Property VI.5, the proof is as follows:

a) Since $f_{\text{acd}}(\lambda, x)$ is concave (Property VI.4) and $\lambda = W_s / (W_s - W) = W_t / (W_t - W)$, then $W_s' = W_t' = 0$ and $f_{\text{acd}}(\lambda, x)$ must be constant along edge *e*, and therefore the solution is the interval $[v_s, v_t]$.
b) The case $\lambda = W_s / (W_s - W)$ and $\lambda > W_t / (W_t - W)$ is not feasible. Indeed, if $W_s' = 0$ and $W_t' < 0$, we obtain $1 = 2(\lambda + (1-\lambda)\sum_{v_i \in A} w_i / W)$ and $1 < 2((1-\lambda)\sum_{v_i \in D} w_i / W)$, then $\lambda + (1-\lambda)\sum_{v_i \in A} w_i / W < (1-\lambda)\sum_{v_i \in D} w_i / W$, which is a contradiction to $D \subseteq A$. On the other hand, if $W_s' = 0$ and $W_t' > 0$, due to Property VI.4 the solution must be the interval $[v_s, \min\{y_e, \min_{b_i \ne 0} b_i\}]$.
c) Analogous to the proof of b), the solution obtained is $[\max\{y_e, \max_{b_i \ne l_e} b_i\}, v_t]$.
d) The case $\lambda > W_s / (W_s - W)$ and $\lambda > W_t / (W_t - W)$ implies $W_s' < 0$ and $W_t' < 0$. This is not feasible, since $1 < 2((1-\lambda)\sum_{v_i \in B} w_i / W)$ and $2(\lambda + (1-\lambda)\sum_{v_i \in C} w_i / W) < 1$, then

$\lambda + (1-\lambda)\sum_{v_i \in C} w_i / W < (1-\lambda)\sum_{v_i \in B} w_i / W$, which is not true for $B \subseteq C$. On the other hand, if $W_s' < 0$ and $W_t' > 0$, as a result of Property VI.4, the solution is v_s.

e) In a similar way to the proof of d), for the case $\lambda > W_t / (W_t - W)$ and $\lambda \neq W_s / (W_s - W)$, we get the solution at v_t. ■

In the case where W_s' and W_t' are both strictly positive, problem (VI.11) is not so easy to analyze. In section VI.9.3, we present a new approach to solve the anti-cent-dian problem. Previous to this, we are going to improve the following upper bound proposed in Moreno-Pérez and Rodríguez-Martín (1999):

$$UB(\lambda, e) = \lambda UB_{\min}(e) + (1-\lambda) UB_{\text{sum}}(e) \qquad \text{(VI.14)}$$

with $UB_{\text{sum}}(e) = (f_{\text{sum}}(v_s) + f_{\text{sum}}(v_t) + l_e)/2$ and $UB_{\min}(e) = (f_{\min}(v_s) + f_{\min}(v_t) + l_e)/2$. Note that the value of function $f_{\min}(x)$ is zero at the ends of any edge *e*, so $UB_{\min}(x) = l_e / 2$. Moreover, this is also deduced from the fact that the local optimum for $f_{\min}(x)$ is located at $y_e = l_e / 2$ with $f_{\min}(y_e) = l_e / 2$. Replacing $UB_{\text{sum}}(e)$ and $UB_{\min}(e)$ in (VI.14) we get

$$UB(\lambda, e) = \lambda \frac{l_e}{2} + (1-\lambda)\frac{f_{\text{sum}}(v_s) + f_{\text{sum}}(v_t) + l_e}{2} = \frac{(1-\lambda)(f_{\text{sum}}(v_s) + f_{\text{sum}}(v_t)) + l_e}{2} \qquad \text{(VI.15)}$$

This bound is computed in $O(n)$ time. We can now improve this upper bound with the same time complexity as follows.

We assume both W_s' and W_t' to be strictly positive. The intersection point *x* such that $f_{\text{acd}}(\lambda, v_s) + xW_s' = f_{\text{acd}}(\lambda, v_t) + W_t'(l_e - x)$, and its ordinate value $y(x)$ are computed.

$$x = \frac{f_{\text{acd}}(\lambda, v_t) - f_{\text{acd}}(\lambda, v_s) + W_t' l_e}{W_s' + W_t'}, \qquad y(x) = \frac{f_{\text{acd}}(\lambda, v_t) W_s' + f_{\text{acd}}(\lambda, v_s) W_t' + W_s' W_t' l_e}{W_s' + W_t'}$$

According to Property VI.4, the anti-cent-dian function at the end nodes of the edge is equal to $(1-\lambda) f_{\text{sum}}(x)$. Then, replacing $f_{\text{acd}}(\lambda, v_s)$ and $f_{\text{acd}}(\lambda, v_t)$ by, respectively, $(1-\lambda) f_{\text{sum}}(v_s)$ and $(1-\lambda) f_{\text{sum}}(v_t)$ yields

$$y(x) = \frac{(1-\lambda)(f_{\text{sum}}(v_t) W_s' + f_{\text{sum}}(v_s) W_t') + W_s' W_t' l_e}{W_s' + W_t'}$$

Let $NUB(\lambda, e) = y(x)$ be the new upper bound. Since $f_{\text{acd}}(\lambda, x)$ is a concave function, obviously $f_{\text{acd}}(\lambda, x) \leq NUB(\lambda, e)$, $\forall x \in e, 0 \leq \lambda \leq 1$. We are now going to prove that the new upper bound is at least as good as (VI.15). Previously, we need to state the following Lemma.

Lemma VI.3. $(1-\lambda) f_{\text{sum}}(v_t) \leq (1-\lambda) f_{\text{sum}}(v_s) + W_s' l_e$.

Proof. Lemma VI.1 stated that $f(v_t) \leq f(v_s) + W_s l_e$, where $f(x) = \sum_{v_i \in V} w_i d(x, v_i)$. If we now divide this expression by *W*, we get $f(v_t)/W \leq f(v_s)/W + W_s l_e / W$. Thus, $f_{\text{sum}}(v_t) \leq f_{\text{sum}}(v_s) + W_s l_e / W$. From (VI.13) we have that $f_{\text{sum}}(v_t) \leq f_{\text{sum}}(v_s) + (W_s' - \lambda) l_e / (1-\lambda)$. Multiplying all the expression by $(1-\lambda)$ we obtain $(1-\lambda) f_{\text{sum}}(v_t) \leq (1-\lambda) f_{\text{sum}}(v_s) + (W_s' - \lambda) l_e$, and therefore $(1-\lambda) f_{\text{sum}}(v_t) \leq (1-\lambda) f_{\text{sum}}(v_s) + W_s' l_e$. ■

Proposition VI.3. *For any edge* $e = (v_s, v_t) \in E$, $NUB(\lambda, e) \leq UB(\lambda, e)$.

Proof. Recall that $W_s', W_t' \leq 1$ always holds. So, if $W_s = W_t = W \Rightarrow W_s' = W_t' = \lambda + \frac{1-\lambda}{W} W = 1$, then

$$NUB(\lambda,e)=\frac{(1-\lambda)(f_{\text{sum}}(v_s)+f_{\text{sum}}(v_t))+l_e}{2}=UB(\lambda,e)$$

When $W_s'=W_t'<1$, we get

$$NUB(\lambda,e)=\frac{W_s'[(1-\lambda)(f_{\text{sum}}(v_s)+f_{\text{sum}}(v_t))+W_s'l_e]}{2W_s'}=\frac{(1-\lambda)(f_{\text{sum}}(v_s)+f_{\text{sum}}(v_t))+W_s'l_e}{2}<UB(\lambda,e)$$

In the case of $W_s',W_t'\le 1$ and $W_s'\ne W_t'$, we make use of the following expression

$$(1-\lambda)[f_{\text{sum}}(v_s)(W_t'-W_s')+f_{\text{sum}}(v_t)(W_s'-W_t')] \tag{VI.16}$$

By virtue of Lemma VI.3, we have that $(1-\lambda)f_{\text{sum}}(v_t)\le(1-\lambda)f_{\text{sum}}(v_s)+W_s'l_e$. Replacing this expression in (VI.16) we obtain

$$\begin{aligned}&(1-\lambda)[f_{\text{sum}}(v_s)(W_t'-W_s')+f_{\text{sum}}(v_t)(W_s'-W_t')]\le\\&\qquad(1-\lambda)[f_{\text{sum}}(v_s)(W_t'-W_s')+f_{\text{sum}}(v_s)(W_s'-W_t')]+W_s'l_e(W_s'-W_t')=W_s'l_e(W_s'-W_t')\end{aligned}$$

Adding and subtracting 1 inside the parentheses of the last term we obtain

$$W_s'l_e(W_s'-W_t'+1-1)=W_s'l_e(1-W_t')+W_s'l_e(W_s'-1)$$

and since $W_s'-1\le 0$, then

$$W_s'l_e(1-W_t')+W_s'l_e(W_s'-1)\le W_s'l_e(1-W_t')\le W_s'l_e(1-W_t')+W_t'l_e(1-W_s')$$

and hence

$$(1-\lambda)[f_{\text{sum}}(v_s)(W_t'-W_s')+f_{\text{sum}}(v_t)(W_s'-W_t')]\le W_s'l_e(1-W_t')+W_t'l_e(1-W_s')$$

Arranging this expression we get

$$(1-\lambda)[f_{\text{sum}}(v_t)W_s'+f_{\text{sum}}(v_s)W_t']+2W_s'W_t'l_e\le(1-\lambda)[f_{\text{sum}}(v_s)W_s'+f_{\text{sum}}(v_t)W_t']+(W_s'+W_t')l_e$$

Adding in both sides of the inequality the term $(1-\lambda)[f_{\text{sum}}(v_t)W_s'+f_{\text{sum}}(v_s)W_t']$, we get

$$2((1-\lambda)(f_{\text{sum}}(v_t)W_s'+f_{\text{sum}}(v_s)W_t')+W_s'W_t'l_e)\le(W_s'+W_t')((1-\lambda)(f_{\text{sum}}(v_s)+f_{\text{sum}}(v_t))+l_e)$$

and therefore, $NUB(\lambda,e)\le UB(\lambda,e)$. ■

Finally, in order to compute the value of this new upper bound on edge *e*, we denote such bound as a function G_{UB} of six parameters:

$$G_{UB}(\lambda,e,F_j,W_j,F_k,W_k)=\frac{(1-\lambda)(W_jF_k+W_kF_j)+W_jW_kl_e}{W_j+W_k}$$

Hence, $NUB(\lambda,e)=G_{UB}(\lambda,e,f_{\text{sum}}(v_s),W_s',f_{\text{sum}}(v_t),W_t')$.

VI.9.3 Solving the anti-cent-dian problem

In Theorem VI.4 we prove how to get a solution to the problem when $W_s'\le 0$ or $W_t'\le 0$. From here on, we show how the anti-cent-dian problem for a particular value of λ, $0<\lambda<1$, can be solved also when $W_s'>0$ and $W_t'>0$.

Let $e=(v_s,v_t)\in E$. Given a point x on *e*, recall the two following sets defined in section VI.5,

$$L(x)=\{v_i\in V:b_i<x\},\qquad R(x)=\{v_i\in V:b_i\ge x\}$$

Making use of the same approach to the maxian problem, we can formulate the function $f_{\text{sum}}(x)$ as

$$f_{\text{sum}}(x) = \frac{1}{W}\left(\sum_{L(x)} w_i(l_e + d(v_t, v_i)) + \sum_{R(x)} w_i d(v_s, v_i)\right) + \frac{x}{W}\left(\sum_{R(x)} w_i - \sum_{L(x)} w_i\right)$$

Let $H(x)$ be equal to the first two summations, which is always positive for any value of x. Hence,

$$f_{\text{sum}}(x) = H(x) + \frac{x}{W}\left(\sum_{R(x)} w_i - \sum_{L(x)} w_i\right)$$

From Property VI.1, $f_{\text{sum}}(x)$ is continuous, concave and piecewise linear, being the values $\frac{1}{W}\left(\sum_{R(x)} w_i - \sum_{L(x)} w_i\right)$ the successive slopes of function $f_{\text{sum}}(x)$.

On the other hand, we have that $f_{\min}(x) = \min_{v_i \in V} d(x, v_i)$. Furthermore, in section VI.9.1 we stated that the local solution to this problem is $y_e = l_e / 2$, with $f_{\min}(y_e) = l_e / 2$ as well. So then, for any $x \in e$ we have

$$f_{\min}(x) = \begin{cases} x & \text{if } x \le y_e \\ l_e - x & \text{if } x > y_e \end{cases} \tag{VI.17}$$

Finally, the anti-cent-dian function $f_{\text{acd}}(\lambda, x)$ over edge *e* is defined as

$$f_{\text{acd}}(\lambda, x) = (1-\lambda)\left(H(x) + \frac{x}{W}\left(\sum_{R(x)} w_i - \sum_{L(x)} w_i\right)\right) + \lambda\begin{cases} x & \text{if } x \le y_e \\ l_e - x & \text{if } x > y_e \end{cases} \tag{VI.18}$$

Since $f_{\text{acd}}(\lambda, x)$ is a concave, continuous and piecewise linear function, we can integrate the slopes of function $f_{\min}(x)$ with the slopes of $f_{\text{sum}}(x)$ to determine the slopes of $f_{\text{acd}}(\lambda, x)$. Then, following the scheme of the maxian problem, we evaluate the slope of $f_{\text{acd}}(\lambda, x)$ at a particular point x to check whether it is increasing, decreasing or remains flat. The points to evaluate are the set of edge bottleneck points $B_e \cup \{y_e\}$.

Let $B'_e = B_e \cup \{y_e\}$ with $|B_e| = n$, that is, we have added y_e as the last point of B'_e, setting $b_{n+1} = y_e$ ($d_{n+1} = 0$) and $w_{n+1} = 0$. Let $l = 1$ and $r = n+1$ be, respectively, the lowest and highest index in B'_e. Let d_q be the median value of all the differences d_i ($l \le i \le r$), that is, the value for which half of the values are smaller, and the other half are greater or equal. This can be computed in O(n) time using the algorithm proposed by Hoare (1961). This algorithm performs a permutation of the elements in B_e such that $d_l, \ldots, d_{q-1}$ are smaller or equal to d_q, and $d_{q+1}, \ldots, d_r$ are greater or equal. Let b_q and w_q be, respectively, the bottleneck point and the weight related to d_q.

We now focus the analysis at the particular median point b_q. Let $W_L = W_L(b_q) = \sum_{L(b_q)} w_i = \sum_{i=l}^{q-1} w_i$. Besides, let $W_R = \sum_{i=q+1}^{r} w_i = W - W_L - w_q$. These are the same variables we used for the maxian problem.

However, we need to define new variables for the anti-cent-dian problem as follows:

$$W'_L = (1-\lambda)\frac{W_L}{W}, \quad W'_R = (1-\lambda)\frac{W_R}{W}, \quad w'_q = (1-\lambda)\frac{w_q}{W} \tag{VI.19}$$

Our goal is to analyze the slopes of the anti-cent-dian function $f_{\text{acd}}(\lambda,x)$. Note that if $y_e \geq x$, then point y_e is in $R(x)$ and, otherwise, is in $L(x)$. So, the slope λ is always related to any of these two sets. Thus, we can express the left slope of this function at a particular point b_q as

$$(W'_R + w'_q) - W'_L + \lambda\alpha_q \qquad \text{(VI.20)}$$

being $\alpha_q = \begin{cases} 1, & \text{if } b_q \leq y_e \\ -1, & \text{if } b_q > y_e \end{cases}$. On the other hand, the right slope is

$$W'_R - (W'_L + w'_q) + \lambda\beta_q \qquad \text{(VI.21)}$$

being $\beta_q = \begin{cases} 1, & \text{if } b_q < y_e \\ -1, & \text{if } b_q \geq y_e \end{cases}$.

As in the maxian problem, we can now check whether this function is increasing, decreasing or becomes flat. But first, we have to add the slopes of the function $f_{\min}(x)$ to the slopes of $f_{\text{sum}}(x)$. So, we define the new variables W_L^*, W_R^*, and w_q^* as follows:

- If $b_q < y_e$ ($d_q < 0$) then let $W_R^* = W'_R + \lambda$, $W_L^* = W'_L$ and $w_q^* = w'_q$.
- If $b_q > y_e$ ($d_q > 0$) then let $W_L^* = W'_L + \lambda$, $W_R^* = W'_R$ and $w_q^* = w'_q$.
- Otherwise ($d_q = 0$), let $W_L^* = W'_L$, $W_R^* = W'_R$ and $w_q^* = w'_q + \lambda$.

Following the analysis given above, the following result is achieved.

Lemma VI.4. *The left slope of function $f_{\text{acd}}(\lambda,x)$ at point b_q is $1-2W_L^*$, while the right slope is $1-2(W_L^* + w_q^*)$.*

Proof. Since $\sum_{R(x)} w_i + \sum_{L(x)} w_i = W$, we have that $\frac{1}{W}\left(\sum_{R(x)} w_i + \sum_{L(x)} w_i\right) = 1$, and thus, $\frac{1-\lambda}{W}\left(\sum_{R(x)} w_i + \sum_{L(x)} w_i\right) + \lambda = 1$. Then, $\frac{1-\lambda}{W}(W_R + w_q + W_L) + \lambda = 1$.

Replacing in the previous expression the new values given in (VI.19) yields

$$W'_R + w'_q + W'_L + \lambda = 1$$

and considering the values recently defined W_L^*, W_R^*, and w_q^*, we have

$$W_R^* + w_q^* + W_L^* = 1 \qquad \text{(VI.22)}$$

since only one of these variables includes λ.

From (VI.20), the left slope of $f_{\text{acd}}(\lambda,x)$ function at point b_q is

$$(W'_R + w'_q) - W'_L + \lambda\alpha_q = (W_R^* + w_q^*) - W_L^*$$

Replacing (VI.22) in the preceding expression we have that the left slope is $1-2W_L^*$.

Likewise, from (VI.21), the right slope is

$$W'_R - (W'_L + w'_q) + \lambda\beta_q = W_R^* - (W_L^* + w_q^*)$$

Taking into account (VI.22) the right slope is $1-2(W_L^* + w_q^*)$. ■

Using the previous Lemma, the following result characterizes the optimal solution to the anti-cent-dian problem in several cases.

Theorem VI.5. *There exists a solution to* (VI.11) *in the next three cases:*

a) *If* $W_L^* + w_q^* = W_R^*$, *then the solution is* $[b_q, \min_{q<i\leq r} b_i]$.

b) *If* $W_L^* = W_R^* + w_q^*$, *then the solution is* $[\max_{l\leq i<q} b_i, b_q]$.

c) *If* $W_R^* - w_q^* < W_L^* < W_R^* + w_q^*$, *then the solution is point* b_q.

Proof.

a) From (VI.22) we have that $W_R^* = 1 - w_q^* - W_L^*$. Then, $W_L^* + w_q^* = 1 - w_q^* - W_L^* \Rightarrow 2(W_L^* + w_q^*) = 1$. This result implies that $f_{\text{acd}}(\lambda, x)$ has attained its maximum at point b_q with a null slope to the following point (Property VI.5). Thus, the solution is the interval $[b_q, \min_{q<i\leq r} b_i]$.

b) $W_L^* = W_R^* + w_q^* \Rightarrow W_L^* = 1 - W_L^* - w_q^* + w_q^* \Rightarrow 2W_L^* = 1$

This is analogous to the preceding case, but shifted one place to the left. Then the solution is $[\max_{l\leq i<q} b_i, b_q]$.

c) $W_L^* < W_R^* + w_q^* \Rightarrow 2W_L^* < 1$, so $f_{\text{acd}}(\lambda, x)$ is increasing at point b_{q-1}. If $W_L^* + w_q^* > W_R^* \Rightarrow 2(W_L^* + w_q^*) > 1$, then $f_{\text{acd}}(\lambda, x)$ is decreasing at b_q. Therefore, $f_{\text{acd}}(\lambda, b_q)$ is the maximum value of $f_{\text{acd}}(\lambda, x)$. ■

These three cases are mutually exclusive, although all possibilities are not considered. There are two more alternatives in which the solution is not achieved.

d) $W_L^* + w_q^* < W_R^*$: function $f_{\text{acd}}(\lambda, x)$ is increasing at point b_q. It follows that the maximum must be to the right of this point. All points b_i such that $l \leq i \leq q$ can be discarded. The search resumes with $l = q + 1$.

e) $W_L^* > W_R^* + w_q^*$: implies that $f_{\text{acd}}(\lambda, x)$ is decreasing and, therefore, all points b_i with $q \leq i \leq r$ can be removed. The search continues with $r = q - 1$.

The next section outlines the new algorithm considering Theorem VI.4 and Theorem VI.5, and the previous cases d) and e). Besides, we introduce an improvement by dynamically updating the new upper bound $NUB(\lambda, e)$ over the point b_q in each iteration. In this way, the search process can be finished as soon as the value of $NUB(\lambda, e)$ is less than the network optimum stored thus far.

VI.9.4 The anti-cent-dian algorithm for a particular value of λ

In this section, we bring together all the results previously stated, outlining the new method in Algorithm VI.2 and proving its complexity.

```
function AntiCentDian(Network N, Distance Matrix d, Parameter λ)
{   // f_N is the current best value on network N, and S is the solution set.
    f_N := 0 , S := ∅
    for all edges e := (v_s, v_t) ∈ E do
    {   y_e := l_e / 2
        X_e := ∅   // Let X_e represent either a single point x or an interval [x^1, x^2].
        if λ = 1 then X_e := y_e
        else
        {   Compute W'_s and W'_t by (VI.13)
            if Theorem VI.4 holds then Store solution in X_e
            else
            {   F_j := (1 − λ) f_sum(v_s) , W_j := W'_s
                F_k := (1 − λ) f_sum(v_t) , W_k := W'_t
                // Compute initial value of the new upper bound.
                NUB(λ, e) := G_UB(λ, e, F_j, W_j, F_k, W_k)
                if NUB(λ, e) < f_N then continue to next edge
                if λ ≠ 0 then   { d_{n+1} := 0 , w_{n+1} := 0 }
                l := 1 , r := n + 1
                while X_e = ∅ and NUB(λ, e) ≥ f_N do
                {   d_q := Median value of all d_i with l ≤ i ≤ r , b_q := (d_q + l_e) / 2
                    Compute W_L, W_R and W'_L, W'_R
                    if d_q < 0 then        W*_R = W'_R + λ , W*_L = W'_L , w*_q = w'_q
                    else if d_q > 0 then   W*_L = W'_L + λ , W*_R = W'_R , w*_q = w'_q
                    else                   W*_L = W'_L , W*_R = W'_R , w*_q = w'_q + λ
                    if a), b) or c) of Theorem VI.5 hold then Store solution in X_e
                    else
                    {   // Search for the optimum to the left or right, cases d),e).
                        if case d) then
                                l := q + 1 , update F_j , W_j , W_L , f_acd(λ, b_q)
                        else    r := q − 1 , update F_k , W_k
                        // Update the upper bound at point b_q
                        NUB(λ, e) := G_UB(λ, e, F_j, W_j, F_k, W_k)
                    }
                }
            }
        }
        if X_e ≠ ∅ and f_acd(λ, X_e) ≥ f_N then
        {   f_N := f_acd(λ, X_e)
            Store the pair (X_e, e) in S
        }
    }
    return (f_N, S)
}
```

Algorithm VI.2: The new algorithm for the λ-anti-cent-dian problem.

The dynamic calculation of the new bound using point b_q is performed by function $G_{UB}(\lambda, e, F_j, W_j, F_k, W_k)$. The values F_j and F_k depend on the value of $f_{\text{acd}}(\lambda, b_q)$, which is

$$f_{\text{acd}}(\lambda, b_q) = (1-\lambda) f_{\text{sum}}(b_q) + \lambda f_{\min}(b_q) =$$

$$= (1-\lambda)\left(\frac{1}{W}\sum_{L(b_q)} d(v_t, v_i) + \frac{1}{W}\sum_{R(b_q)} w_i d(v_s, v_i) + \frac{l_e}{W}\sum_{L(b_q)} w_i + \frac{b_q}{W}\left(\sum_{R(b_q)} w_i - \sum_{L(b_q)} w_i\right)\right) + \lambda f_{\min}(b_q)$$

Replacing $f^L_{\text{sum}}(b_q) = \sum_{L(b_q)} w_i d(v_t, v_i)/W$ and $f^R_{\text{sum}}(b_q) = \sum_{R(b_q)} w_i d(v_s, v_i)/W$ we get

$$f_{\text{acd}}(\lambda, b_q) = (1-\lambda)(f^L_{\text{sum}}(b_q) + f^R_{\text{sum}}(b_q)) + l_e W'_L + b_q((W'_R + w'_q) - W'_L) + \lambda f_{\min}(b_q) =$$

$$= (1-\lambda)(f^L_{\text{sum}}(b_q) + f^R_{\text{sum}}(b_q)) + (W'_R + w'_q) b_q + W'_L(l_e - b_q) + \lambda \begin{cases} b_q & \text{if } b_q \le y_e \\ l_e - b_q & \text{if } b_q > y_e \end{cases}$$

If a new median is computed in the next iteration, say for example d_p with bottleneck point b_p, the value of $f_{\text{acd}}(\lambda, b_p)$ can be determined from $f_{\text{acd}}(\lambda, b_q)$ in a similar way to the maxian problem:

- If $b_p < b_q$ then

$$f^L_{\text{sum}}(b_p) + f^R_{\text{sum}}(b_p) = f^L_{\text{sum}}(b_q) + f^R_{\text{sum}}(b_q) + \frac{1}{W}\sum_{i=p}^{r} w_i (d(v_s, v_i) - d(v_t, v_i)) =$$

$$= f^L_{\text{sum}}(b_q) + f^R_{\text{sum}}(b_q) - \frac{1}{W}\sum_{i=p}^{r} w_i d_i$$

- If $b_p > b_q$ then

$$f^L_{\text{sum}}(b_p) + f^R_{\text{sum}}(b_p) = f^L_{\text{sum}}(b_q) + f^R_{\text{sum}}(b_q) + \frac{1}{W}\sum_{i=l}^{p-1} w_i (d(v_t, v_i) - d(v_s, v_i)) =$$

$$= f^L_{\text{sum}}(b_q) + f^R_{\text{sum}}(b_q) + \frac{1}{W}\sum_{i=l}^{p-1} w_i d_i$$

Likewise, the computation of $W_L(b_p)$ is figured out using the same approach of the maxian problem. Finally, each time cases d) or e) are satisfied, the values F_j, W_j and F_k, W_k must be update accordingly:

- If case d) is fulfilled, update $W_j = 1 - 2(W^*_L + w^*_q)$ and $F_j = f_{\text{acd}}(\lambda, b_q) - W_j b_q$. Besides, since we move to the right, we must set $W_L = W_L + w_q$ and $f_{\text{acd}}(\lambda, b_q) = f_{\text{acd}}(\lambda, b_q) + (1-\lambda) w_q d_q / W$.
- Else, update $W_k = 2W^*_L - 1$ and $F_k = f_{\text{acd}}(\lambda, b_q) - W_k(l_e - b_q)$, leaving W_L and $f_{\text{acd}}(\lambda, b_q)$ unchanged.

As in the maxian problem, each iteration of the 'while' loop deletes $q = (l + r)/2$ points from B'_e. Thus, the complexity of the algorithm is the same of the maxian problem.

Theorem VI.6. *Provided that the distance matrix is given, the new algorithm solves the network λ-anti-cent-dian problem for a given λ, $0 \le \lambda \le 1$, in $O(mn)$ time.*

Proof. Given any edge *e*, the initial new bound $NUB(\lambda, e)$ is computed in $O(n)$ time. In the same way as the maxian algorithm, each iteration of the 'while' loop diminishes the size of B'_e to a half. Thus, the number of points processed is

$$n+\frac{n}{2}+\frac{n}{4}+\cdots+\frac{n}{2^k}=n\left(\frac{2^k+2^{k-1}+\cdots+1}{2^k}\right)=\frac{n}{2^k}\sum_{i=0}^{k}2^i=\frac{n}{2^k}(2^{k+1}-1)$$

This loop runs, in the worst case, until l and r are consecutive. Hence, $n/2^k=1\Rightarrow n=2^k$, and consequently, $(n/2^k)(2^{k+1}-1)=2n-1\in O(n)$. This must be applied to all m edges. Thus, the overall complexity is $O(mn)$. ■

VI.10 Conclusions

The main purpose of this chapter was twofold. In the first part, the 1-maxisum location problem (maxian) problem on networks is analyzed. From Church and Garfinkel (1978), an initial upper bound $UB(e)$ is derived, which is improved with a new upper bound $NUB(e)$. Likewise, this bound can be dynamically updated without increasing the total computational time.

We have developed a new algorithm in $O(mn)$ to solve this problem. The procedure makes use of the new upper bound, and hence, allows skipping out from the search process as soon as the upper bound is less than the global optimum. This new algorithm has been compared with the procedure by Church and Garfinkel (1978), including the initial bound $UB(e)$, on low and high dense networks, as well as on planar networks. In all cases, the new algorithm accomplishes a better performance in the computing times.

On the other hand, the second part addresses the λ-anti-cent-dian. A new upper bound $NUB(\lambda, e)$ has been proposed, as well as a new $O(mn)$ algorithm which improves the former $O(mn\log n)$ method by Moreno-Pérez and Rodríguez-Martín (1999).

Chapter VII

Undesirable facility location problems on multicriteria networks

"The real world problem of locating an undesirable facility
is clearly a multiobjective decision problem"
E. ERKUT & S. NEUMAN

VII.1 Introduction

As we stated in the introductory chapter, most of the huge literature on Location Analysis deals with the sitting of facilities such as shopping stores, emergency services and educational centers. All of these facilities are *desirable* (attractive) to the nearby inhabitants which try to have them as close as possible.

However, there are some other facilities such as garbage dump sites, landfills, chemical plants, nuclear reactors, military installations and polluting (noise/gas) plants that turn out to be *undesirable* (repulsive) for the surrounding population, which avoids them and tries to stay away from them. In this sense, Erkut and Neuman (1989) distinguish between *noxious* (hazardous) and *obnoxious* (nuisance) facilities, although both can be simply regarded as *undesirable*.

Despite these undesirable facilities being necessary, in general, to the community, for instance garbage dump sites, gas stations, electrical plants, etc., the location of such facilities might cause a certain disagreement in the population. Such disagreement has become a true opposition of people to the installation of undesirable facilities close to them. Moreover, in the last decade, a new nomenclature has been developed to define this opposition: NIMBY (*Not In My Back Yard*), NIMNBY (*Not In My Neighbor's Back Yard*), NIABY (*Not In Anyone's Back Yard*), NIMTOO/NIMTOF (*Not In My Term of Office*), NOPE (*Not On Planet Earth*), LULU (*Locally Unwanted Land Use*), BANANA (*Build Absolutely Nothing Anywhere Near Anyone*).

The classical location criteria *minimax* (center) and *minisum* (median) are useless to locate this type of facility. Thus, the *maximin/maxmax* and the *maxisum* criteria arose to model, respectively, the undesirable center problem and the undesirable median problem. By placing the new facility away from existing facilities, the maximin criterion minimizes the effect on the worst impacted existing facility, whereas the maxisum criterion minimizes the collective effect (average) on the existing facilities.

Likewise, some facilities might be considered *semi-desirable* since they provide a main service to the community but they can also cause inconveniences to the neighboring areas, for instance, an airport, a train station, or any other noisy facility. These problems can be perfectly modeled combining the minimax/minisum criteria and the maximin/maxisum criteria.

In this sense, the undesirable facility location models analyzed in previous chapters are basically single-criterion, and they were related to the papers by Melachrinoudis and Zhang (1999), Berman and Drezner (2000), Minieka (1983), Church and Garfinkel (1978), and Tamir (1988, 1991). Nevertheless, Erkut and Neuman (1989) emphasized the need for multiobjective approaches to the siting of undesirable facilities when they stated that (p. 289): *"Current models can be used to generate a small number of candidate sites, but the final selection of a site is a complex problem and should be approached using multiobjective decision making tools"*. Daskin (1995) and Zhang (1996) also pointed out not only the need to include multiple criteria in undesirable facility location problems, but also the fact that poor attention has been paid by researchers to these problems and hence, little research has been done in this promising field.

Thus, the literature on multicriteria/multiobjective undesirable facility location on networks starts in the late eighties and is rather scarce. Ratick and White (1988) proposed a multiobjective model for the location of undesirable facilities considering three objectives. List and Mirchandani (1991) presented a combined routing/siting model that can be used for siting decisions of waste treatment facilities. Rahman and Kuby (1995) examine the tradeoffs between minimizing costs and public opposition in the location of a solid waste transfer station. Giannikos (1998) presented a multiobjective model for locating disposal facilities and transporting hazardous waste along the links of a network considering four objectives. Zhang and Melachrinoudis (2001) considered the problem of locating an obnoxious facility on a general network using two objectives, maximizing the minimum weighted distance from the point to the vertices and maximizing the sum of weighted distances between the point and the vertices. Skriver and Andersen (2001) modeled a semi-obnoxious facility location problem as a bicriterion problem in both the plane and the network case. Finally, Hamacher, Labbé, Nickel, and Skriver (2002) presented a polynomial time algorithm for the location of a semi-obnoxious facility on networks.

Accordingly, in this chapter we present a multicriteria undesirable facility location model on networks with several weights on the nodes and several lengths on the edges, combining the maximin and maxisum criteria by a parameter λ. Such a model can be considered the opposite to the multicriteria network λ-cent-dian problem presented in Chapter VI and hence, it can be described as the *multicriteria λ-anti-cent-dian* problem on networks. According to the classification scheme of Chapter I, this problem is denoted as $1/\mathcal{G}/\bullet/d(\mathcal{V},\mathcal{G})/Q-\mathrm{CD}_{\text{obnox-par}}$. This model generalizes the anti-cent-dian model presented in section VI.9.

The remainder of the chapter is structured as follows. In the first section we introduce some basic definitions and the notation. In the two following sections, we analyze both the uncenter and maxian problems considering firstly two criteria and then extending those results to the multicriteria case. Section VII.5 is devoted to the multicriteria λ-anti-cent-dian problem, whereas in section VII.6 the algorithm proposed to solve this problem is devised and commented upon. To illustrate this algorithm, in section VII.7 we develop a brief example. Finally, we present the computational experience in section VII.8, and the chapter ends with the conclusions and the discussion.

VII.2 Notation and basic definitions

Let $N = (V, E)$ be an undirected, simple and connected network, with node set $V = \{v_1, v_2, \dots, v_n\}$, and $E = \{(v_s, v_t) : v_s, v_t \in V\}$ being the set of edges. Let *p* be the number of weights associated with each node, and *q* the number of lengths (costs) attached to each edge. For each vertex in *V*, we define the following weight function

$$\begin{array}{llll} w: & V & \longrightarrow & \mathbb{R}^p \\ & v_i \in V & \longrightarrow & w(v_i) = w_i = (w_i^1, \dots, w_i^p) \end{array}$$

Similarly, over each edge in *E* we define the next length function

$$\begin{array}{llll} l: & E & \longrightarrow & \mathbb{R}^q \\ & e = (v_s, v_t) \in E & \longrightarrow & l(e) = l_e = (l_e^1, \dots, l_e^q) \end{array}$$

Let *r* be a length index, with $1 \le r \le q$, and let $x \in e = (v_s, v_t)$ be a point within *e*. We define $c_e^r(x, v_s)$ as the length of the line segment between *x* and v_s regarding length *r*, with $0 \le c_e^r(x, v_s) \le l_e^r$ and $c_e^r(x, v_t) = l_e^r - c_e^r(x, v_s)$. For any two nodes $v_a, v_b \in V$, the distance between such nodes, denoted by $d^r(v_a, v_b)$, is defined as the length of any shortest path on *N* joining v_a and v_b concerning length *r*.

In the same way, given any point $x \in N$ and any node $v_i \in V$, let $d^r(x, v_i) = \min\{c_e^r(x, v_s) + d(v_s, v_i), c_e^r(x, v_t) + d(v_t, v_i)\}$ be the distance between point *x* and node v_i considering length *r*. The point on edge *e* where $d^r(x, v_i)$ attains its equilibrium is called a *bottleneck point*, which is defined as $b_i^r = (d^r(v_t, v_i) - d^r(v_s, v_i) + l_e^r)/2$. Given a length index *r*, the set of all bottleneck points on edge *e* is denoted by $B_e^r = \bigcup_{v_i \in V} b_i^r$, whereas the set of all bottleneck points on network *N* is denoted by $B_N^r = \bigcup_{e \in E} B_e^r$.

Given a weight index *s* and a length index *r*, let Q_e^{sr} be a set containing points $x \in e$ such that, for two distinct nodes $v_i, v_j \in V$, $w_i^s d^r(x, v_i) = w_j^s d^r(x, v_j)$ and, besides, $d^r(x, v_i)$ and $d^r(x, v_j)$ do not both decrease when *x* is perturbed slightly in either direction. Let $Q_N^{sr} = \bigcup_{e \in E} Q_e^{sr}$.

Now, we are ready to define both the weighted undesirable center (*uncenter*) function and the undesirable median (*maxian*) function on multicriteria networks, and to present new properties as well as some rules to remove inefficient edges.

VII.3 The multicriteria uncenter problem

Given any point $x \in N$, any weight *s* ($1 \le s \le p$) and any length *r* ($1 \le r \le q$), let $f_{\min}^{sr}(x) = \min_{v_i \in V} w_i^s d^r(x, v_i)$ be the minimum weighted distance from *x* to the set of nodes. Recall from Chapter V that given an edge $e \in E$, a point $y_e^{sr} \in Q_e^{sr}$ is a *local uncenter point* on edge *e* iff $f_{\min}^{sr}(y_e^{sr}) = \max_{x \in e} f_{\min}^{sr}(x)$, for any values (s, r), with $1 \le s \le p$ and $1 \le r \le q$. Likewise, a point $y_N^{sr} \in Q_N^{sr}$ is a *network uncenter point* iff $f_{\min}^{sr}(y_N^{sr}) = \max_{x \in N} f_{\min}^{sr}(x) = \max_{e \in E} f_{\min}^{sr}(y_e^{sr})$, for any value of indices *s* and *r*.

The uncenter function $f_{\min}^{sr}(x)$ is a continuous, concave and piecewise linear function with a unique uncenter point y_e^{sr} on each edge $e \in E$. Besides, the value of this function is zero at the ends of edge *e*. For the formal description of the uncenter properties the reader is referred to section V.2.

Obviously, in the multicriteria case there can be at most $k = p \times q$ uncenter functions. Given a point $x \in N$, let $F_{\min}(x) = (f_{\min}^{11}(x), f_{\min}^{12}(x), \ldots, f_{\min}^{pq}(x)) \in \mathbb{R}^{p \times q}$ be the vector of values of the uncenter function $f_{\min}^{sr}(x)$ for all combinations of the weight indices $s = 1, \ldots, p$ with the length indices $r = 1, \ldots, q$. To make the notation easier, from now on we denote the uncenter functions as $f_{\min}^{i}(x)$, with $i = 1, \ldots, k$.

A set of points $Y_N \subset N$ is an efficient set for the multicriteria uncenter problem iff $F_{\min}(Y_N) = \max_{x \in N} F_{\min}(x)$. In this sense, being this model a maximization problem, the efficiency is expressed in a different way from that in which it was initially defined in Chapter I. Thus, given two points $x, y \in N$, we say that x dominates point y, and is denoted by $x \succ y$, if $f_{\min}^{i}(x) \geq f_{\min}^{i}(y)$, $\forall i = 1, \ldots, k$, with at least one of the inequalities strict. Then, a point $x \in N$ is an *efficient* or *Pareto-optimal* point for the multicriteria uncenter problem if there is no other point $y \in N$ such that $y \succ x$.

We now state some basic properties for the multicriteria uncenter problem when $k = 2$, and then we extend these properties for any value of *k*. Given an edge $e = (v_s, v_t) \in E$, let y_e^1 and y_e^2 be the local uncenter points of each objective function $f_{\min}^{i}(x)$, $i = 1, 2$. From now on we assume that the local uncenter points are measured with respect to the first length l_e^1.

Lemma VII.1. *If* $y_e^1 \neq y_e^2$, *then the set of local efficient points on edge e is* $Y_e = [\min\{y_e^1, y_e^2\}, \max\{y_e^1, y_e^2\}]$ (bold line in Figure VII.1).

Proof. For $1 \leq i \leq 2$, the two uncenter functions are concave, with $f_{\min}^{i}(v_s) = f_{\min}^{i}(v_t) = 0$, and they are increasing in the interval $[v_s, y_e^i]$ and decreasing in $[y_e^i, v_t]$. Hence, $f_{\min}^{1}(y_e^1) > f_{\min}^{1}(y_e^2)$ and $f_{\min}^{2}(y_e^1) < f_{\min}^{1}(y_e^1)$, and the result follows. ∎

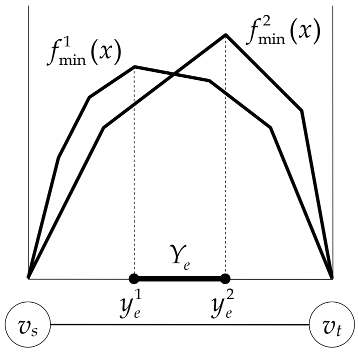


Figure VII.1: Illustration of Lemma VII.1.

From this result we can derive two interesting consequences.

Corollary VII.1. *All the points belonging to the intervals* $[v_s, \min\{y_e^1, y_e^2\})$ *and* $(\max\{y_e^1, y_e^2\}, v_t]$ *are inefficient points.*

Corollary VII.2. *If* $y_e^1 = y_e^2$ *then the unique local efficient point on edge e is the point* $Y_e = y_e^1 = y_e^2$.

With these ideas, we can now establish a rule by which all inefficient edges can be easily removed and, hence, the search of the optimal points will become faster. Let $y_N^1, y_N^2 \in N$ be the network uncenter points for each objective function.

Proposition VII.1. *Edge e contains no efficient points and, thus, it can be discarded if for some point* y_N^i, $1 \le i \le 2$, *satisfies* $f_{\min}^1(y_e^1) \le f_{\min}^1(y_N^i)$ *and* $f_{\min}^2(y_e^2) \le f_{\min}^2(y_N^i)$, *with at least one inequality strict.*

Proof. The maximum values of each objective function $f_{\min}^i(x)$, $1 \le i \le 2$, are attained, respectively, at y_e^1 and y_e^2. Any other point *x* inside interval Y_e holds smaller values (Lemma VII.1). Thus, any point with function values greater than y_e^1 and y_e^2 will dominate all points inside interval Y_e. Therefore, if $f_{\min}^1(y_e^1) \le f_{\min}^1(y_N^1)$ and $f_{\min}^2(y_e^2) \le f_{\min}^2(y_N^1)$, with at least one inequality strict, then $y_N^1 \succ Y_e$. The same analysis can be applied to y_N^2, and hence the result follows. ■

We now extend the previous results for any value of *k*. Given an edge $e = (v_s, v_t) \in E$, let y_e^i be the local uncenter point of each objective function $f_{\min}^i(x)$, $i = 1, \ldots, k$. Lemma VII.1 and the subsequent corollaries can be extended to the multicriteria case as follows. If all points y_e^i are equal, then it is obvious that the efficient point is $Y_e = y_e^1 = \cdots = y_e^k$. Otherwise, the following properties are verified.

Lemma VII.2. *The set of local efficient points on edge e is* $Y_e = [\min_{1\le i\le k} y_e^i, \max_{1\le i\le k} y_e^i]$.

Proof. The proof is straightforward since the objective functions are concave and each objective function $f_{\min}^i(x)$ is increasing in $[v_s, y_e^i]$ and decreasing in $[y_e^i, v_t]$, $i = 1, \ldots, k$. ■

Corollary VII.3. *All the points belonging to the intervals* $[v_s, \min_{1\le i\le k} y_e^i)$ *and* $(\max_{1\le i\le k} y_e^i, v_t]$ *are inefficient points.*

Before analyzing the multicriteria maxian problem, we present a rule by which all inefficient edges are removed from *E*. Let $y_N^1, \ldots, y_N^k$ be the network uncenter points for all the *k* objective functions.

Proposition VII.2. *If some network uncenter point* y_N^i, $1 \le i \le k$, *satisfies*

$$f_{\min}^1(y_e^1) \le f_{\min}^1(y_N^i) \quad \wedge \quad f_{\min}^2(y_e^2) \le f_{\min}^2(y_N^i) \quad \wedge \quad \ldots \quad \wedge \quad f_{\min}^k(y_e^k) \le f_{\min}^k(y_N^i)$$

with at least one inequality strict, then edge e contains no efficient points inside and hence, it can be discarded.

Proof. It follows from the proof of Proposition VII.1, but considering now *k* objective functions. ■

VII.4 The multicriteria maxian problem

Given any point $x \in N$, we define the function $f_{\text{sum}}^{sr}(x) = \sum_{v_i \in V} w_i^s d^r(x, v_i)$ as the sum of weighted distances from point *x* to the set of nodes, with $1 \le s \le p$ and $1 \le r \le q$. The properties of $f_{\text{sum}}^{sr}(x)$ were stated in Chapter VI. Recall that these functions are continuous, concave and piecewise linear over each edge $e \in E$, with at least one *local maxian point* $z_e^{sr} \in B_e^r \cup \{v_s, v_t\}$ such that $f_{\text{sum}}^{sr}(z_e^{sr}) = \max_{x \in e} f_{\text{sum}}^{sr}(x)$, with $1 \le s \le p$ and $1 \le r \le q$. Besides, if $f_{\text{sum}}^{sr}(x)$ reaches its maximum value at two consecutive points $\tilde{z}_e^{sr}, \hat{z}_e^{sr} \in B_e^r \cup \{v_s, v_t\}$, then all points in $[\tilde{z}_e^{sr}, \hat{z}_e^{sr}]$ also maximize $f_{\text{sum}}^{sr}(x)$.

A point $z_N^{sr} \in B_N^r \cup V$ is a *network maxian point* iff $f_{\text{sum}}^{sr}(z_N^{sr}) = \max_{x \in N} f_{\text{sum}}^{sr}(x) = \max_{e \in E} f_{\text{sum}}^{sr}(z_e^{sr})$, for $1 \le s \le p$ and $1 \le r \le q$. Likewise, two consecutive points $\tilde{z}_N^{sr}, \hat{z}_N^{sr} \in B_N^r \cup V$ are the *network maxian points* iff $f_{\text{sum}}^{sr}(z) = \max_{x \in N} f_{\text{sum}}^{sr}(x)$, $\forall z \in [\tilde{z}_N^{sr}, \hat{z}_N^{sr}]$.

Let $F_{\text{sum}}(x) = (f_{\text{sum}}^{11}(x), f_{\text{sum}}^{12}(x), \ldots, f_{\text{sum}}^{pq}(x)) \in \mathbb{R}^{p \times q}$ be the vector of values of the maxian function $f_{\text{sum}}^{sr}(x)$ for all combinations of weights $s = 1, \ldots, p$ and lengths $r = 1, \ldots, q$. For the sake of comprehensibility, let $k = p \times q$, and henceforth we denote the maxian functions as $f_{\text{sum}}^{i}(x)$, with $i = 1, \ldots, k$.

The set of points $Z_N \subset N$ is the efficient set for the multicriteria maxian problem iff $F_{\text{sum}}(Z_N) = \max_{x \in N} F_{\text{sum}}(x)$. Given two points $x, y \in N$, we say that x dominates point y, and is denoted by $x \succ y$, if $f_{\text{sum}}^{i}(x) \ge f_{\text{sum}}^{i}(y)$, $\forall i = 1, \ldots, k$, with at least one of the inequalities strict. Then, a point $x \in N$ is an efficient or Pareto-optimal point for the multicriteria maxian problem if there is no other point $y \in N$ such that $y \succ x$.

Following the analysis of the multicriteria uncenter problem, we can now obtain new properties for the maxian problem when $k = 2$. Given an edge $e = (v_s, v_t) \in E$, if each function $f_{\text{sum}}^{i}(x)$ has a single local maxian point z_e^i, $1 \le i \le 2$, then Lemma VII.1 is fulfilled. Otherwise, let $[\tilde{z}_e^1, \hat{z}_e^1]$ and $[\tilde{z}_e^2, \hat{z}_e^2]$ be, respectively, the *local maxian intervals* where $f_{\text{sum}}^{1}(x)$ and $f_{\text{sum}}^{2}(x)$ attain their maximum values, with $\tilde{z}_e^i, \hat{z}_e^i \in B_e^r \cup \{v_s, v_t\}$, $i = 1, 2$.

Lemma VII.3. *The set of efficient points on edge e is* $Z_e = [\min\{\tilde{z}_e, \hat{z}_e\}, \max\{\tilde{z}_e, \hat{z}_e\}]$, *where* $\tilde{z}_e = \max\{\tilde{z}_e^1, \tilde{z}_e^2\}$ *and* $\hat{z}_e = \min\{\hat{z}_e^1, \hat{z}_e^2\}$.

Proof.

a) Without loss of generality, we assume $\tilde{z}_e^1 < \hat{z}_e^1 < \tilde{z}_e^2 < \hat{z}_e^2$ (see Figure VII.2a). In this case $\tilde{z}_e = \tilde{z}_e^2$ and $\hat{z}_e = \hat{z}_e^1$. Due to the concavity of the objective functions, point $\tilde{z}_e$ dominates all points $x \in (\tilde{z}_e^2, v_t]$. On the other hand, $\hat{z}_e \succ [v_s, \hat{z}_e^1)$. Inside interval $[\hat{z}_e^1, \tilde{z}_e^2]$, function $f_{\text{sum}}^{1}(x)$ is decreasing, whereas $f_{\text{sum}}^{2}(x)$ is increasing. Therefore, $Z_e = [\hat{z}_e, \tilde{z}_e]$.

b) Otherwise, by virtue of the concavity property it follows that point $\tilde{z}_e \succ [v_s, \tilde{z}_e)$ and point $\hat{z}_e \succ (\hat{z}_e, v_t]$ (see for example Figure VII.2b,c). Thus, $Z_e = [\tilde{z}_e, \hat{z}_e]$.

Hence, the efficient set on edge *e* is $Z_e = [\min\{\tilde{z}_e, \hat{z}_e\}, \max\{\tilde{z}_e, \hat{z}_e\}]$. ■

This previous result establishes the set of efficient points when the two objective functions attain their maximum value inside an interval. An interesting consequence is that the set Z_e is valid even if either $f_{\text{sum}}^{1}(x)$ or $f_{\text{sum}}^{2}(x)$ reach their maximum at a single point, as the next result states.

Corollary VII.4. *Even if the local maxian points are attained at a single point, that is* $\tilde{z}_e^1 = \hat{z}_e^1 = z_e^1$ *or* $\tilde{z}_e^2 = \hat{z}_e^2 = z_e^2$, *the set of efficient points on edge e is* $Z_e = [\min\{\tilde{z}_e, \hat{z}_e\}, \max\{\tilde{z}_e, \hat{z}_e\}]$.

Corollary VII.5. *If* $\tilde{z}_e^1 = \hat{z}_e^1 = \tilde{z}_e^2 = \hat{z}_e^2 = z_e$ *then the unique local efficient point on edge e is the point* $Z_e = z_e$.

The next result provides the rule to delete all edges that contain no efficient points. We assume that $[\tilde{z}_N^1, \hat{z}_N^1]$ and $[\tilde{z}_N^2, \hat{z}_N^2]$, with $\tilde{z}_N^i, \hat{z}_N^i \in N$, $i = 1, 2$, are the network maxian intervals for each objective function.

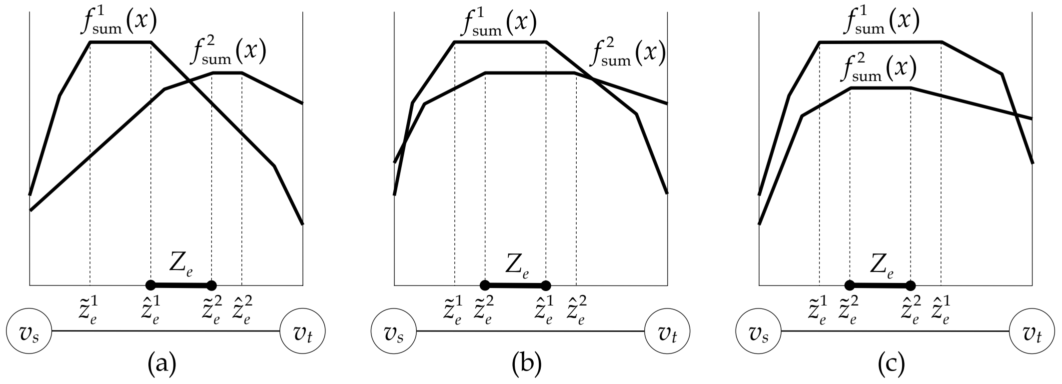


Figure VII.2: Some cases fulfilled in Lemma VII.3.

Proposition VII.3. *Edge e contains no efficient points and hence can be deleted if the points* $\tilde{z}_N^i, \hat{z}_N^i$, $1 \le i \le 2$, *satisfy* $f^1_{\text{sum}}(z_e^1) \le f^1_{\text{sum}}(\tilde{z}_N^i)$ *and* $f^2_{\min}(z_e^2) \le f^2_{\min}(\tilde{z}_N^i)$, *or* $f^1_{\text{sum}}(z_e^1) \le f^1_{\text{sum}}(\hat{z}_N^i)$ *and* $f^2_{\min}(z_e^2) \le f^2_{\min}(\hat{z}_N^i)$, *with at least one of the inequalities strict.*

Proof. Considering the two extreme points of the network maxian intervals, the proof follows from Proposition VII.1. ■

Next, and before analyzing the multicriteria λ-anti-cent-dian problem, we extend these previous results for any value of *k*.

Lemma VII.4. *The set of efficient points on edge e is* $Z_e = [\min\{\tilde{z}_e, \hat{z}_e\}, \max\{\tilde{z}_e, \hat{z}_e\}]$, *where* $\tilde{z}_e = \max_{1\le i\le k} \tilde{z}_e^i$ *and* $\hat{z}_e = \min_{1\le i\le k} \hat{z}_e^i$.

Proof.

a) Since all the objective functions are concave, if $\hat{z}_e < \tilde{z}_e$ then $\hat{z}_e \succ [v_s, \hat{z}_e)$ and $\tilde{z}_e \succ (\tilde{z}_e, v_t]$. Inside interval $[\hat{z}_e, \tilde{z}_e]$, some objective functions are decreasing, some others keep increasing and others may remain flat. Therefore, $Z_e = [\hat{z}_e, \tilde{z}_e]$.
b) Otherwise, $\tilde{z}_e \succ [v_s, \tilde{z}_e)$ and point $\hat{z}_e \succ (\hat{z}_e, v_t]$. Hence $Z_e = [\tilde{z}_e, \hat{z}_e]$.

Accordingly, the efficient set on edge *e* is $Z_e = [\min\{\tilde{z}_e, \hat{z}_e\}, \max\{\tilde{z}_e, \hat{z}_e\}]$. ■

Corollary VII.6. *In case that* $\tilde{z}_e^i = \hat{z}_e^i = z_e^i$ *for some objective functions* $f^i_{\text{sum}}(x)$, *with* $1 \le i \le k$, *then the set of efficient points on edge e is* $Z_e = [\min\{\tilde{z}_e, \hat{z}_e\}, \max\{\tilde{z}_e, \hat{z}_e\}]$.

Finally, we state the rule to delete all inefficient edges. We assume that $[\tilde{z}_N^i, \hat{z}_N^i]$, $\tilde{z}_N^i, \hat{z}_N^i \in N$, $i = 1, \ldots, k$, are the network maxian intervals for each objective function.

Proposition VII.4. *If any point* $\tilde{z}_N^i, \hat{z}_N^i$, $1 \le i \le k$, *satisfies*

$$f^1_{\text{sum}}(z_e^1) \le f^1_{\text{sum}}(\tilde{z}_N^i) \;\wedge\; f^2_{\text{sum}}(z_e^2) \le f^2_{\text{sum}}(\tilde{z}_N^i) \;\wedge\; \ldots \;\wedge\; f^k_{\text{sum}}(z_e^k) \le f^k_{\text{sum}}(\tilde{z}_N^i)$$

or

$$f^1_{\text{sum}}(z_e^1) \le f^1_{\text{sum}}(\hat{z}_N^i) \;\wedge\; f^2_{\text{sum}}(z_e^2) \le f^2_{\text{sum}}(\hat{z}_N^i) \;\wedge\; \ldots \;\wedge\; f^k_{\text{sum}}(z_e^k) \le f^k_{\text{sum}}(\hat{z}_N^i)$$

with at least one of the inequalities strict, then edge e contains no efficient points, and therefore, it can be removed.

Proof. It follows from Proposition VII.3. ■

VII.5 Multicriteria λ-anti-cent-dian problem (MACDP)

Given $\lambda \in [0,1]$ and $x \in N$, the λ-anti-cent-dian function is defined as follows

$$f_{\text{acd}}^{sr}(\lambda,x) = \lambda f_{\min}^{sr}(x) + (1-\lambda) f_{\text{sum}}^{sr}(x)$$

being $f_{\min}^{sr}(x) = \min_{v_i \in V} w_i^s d^r(x,v_i)$ and $f_{\text{sum}}^{sr}(x) = \sum_{v_i \in V} w_i^s d^r(x,v_i)$, with $s=1,\ldots,p$ and $r=1,\ldots,q$. This model was introduced in Chapter VI, though function $f_{\min}(x)$ was unweighted and $f_{\text{sum}}(x)$ was divided by the total sum of weights. Provided that both $f_{\min}^{sr}(x)$ and $f_{\text{sum}}^{sr}(x)$ are continuous, concave and piecewise linear functions on *x*, the λ-anti-cent-dian function $f_{\text{acd}}^{sr}(\lambda,x)$, being a convex combination of the two latter functions, fulfills these characteristics as well. Thus, bringing together the properties of the uncenter function (Chapter V) and the maxian function (Chapter VI), we can now derive new properties for function $f_{\text{acd}}^{sr}(\lambda,x)$.

Property VII.1. *Given any edge $e=(v_s,v_t)\in E$ and a value λ, $0 \le \lambda \le 1$, for any point $x \in e$ the objective function $f_{\text{acd}}^{sr}(\lambda,x)$, $1 \le s \le p$, $1 \le r \le q$, is a continuous, concave and piecewise linear function,*

a) *having a finite number of breakpoints, all belonging to $B_e^r \cup Q_e^{sr}$,*
b) *with a finite number of locally maximum values, all attained at the points belonging to the set $A = \{v_s,v_t\} \cup B_e^r \cup Q_e^{sr}$,*
c) *having value zero at the ends of the edge for $\lambda = 1$, and*
d) $f_{\text{acd}}^{sr}(\lambda,v_s) = (1-\lambda) f_{\text{sum}}^{sr}(v_s)$ *and* $f_{\text{acd}}^{sr}(\lambda,v_t) = (1-\lambda) f_{\text{sum}}^{sr}(v_t)$.

Property VII.2. *Given a value of λ, $0 \le \lambda \le 1$, and $1 \le s \le p$, $1 \le r \le q$, there exists at least one point, called the local anti-cent-dian point $x_e^{sr} \in A = \{v_s,v_t\} \cup B_e^r \cup Q_e^{sr}$ on each edge $e=(v_s,v_t)\in E$ such that $f_{\text{acd}}^{sr}(\lambda,x_e^{sr}) = \max_{x\in e} f_{\text{acd}}^{sr}(\lambda,x)$. If function $f_{\text{acd}}^{sr}(\lambda,x)$ reaches its maximum value at two consecutive points $\tilde{x}_e^{sr},\hat{x}_e^{sr} \in A$, then all points inside $[\tilde{x}_e^{sr},\hat{x}_e^{sr}]$ maximize function $f_{\text{acd}}(\lambda,x)$.*

Likewise, a point $x_N^{sr} \in N$ is called a *network anti-cent-dian point* for a certain value of λ, $0 \le \lambda \le 1$, iff $f_{\text{acd}}^{sr}(\lambda,x_N^{sr}) = \max_{x\in N} f_{\text{acd}}^{sr}(\lambda,x) = \max_{e\in E} f_{\text{acd}}^{sr}(\lambda,x_e^{sr})$, with $1 \le s \le p$ and $1 \le r \le q$. Moreover, since $f_{\text{acd}}^{sr}(\lambda,x) = f_{\min}^{sr}(x)$ when $\lambda = 1$, the local (network) anti-cent-dian point x_e^{sr} (x_N^{sr}) is equal to the local (network) uncenter point y_e^{sr} (y_N^{sr}). On the other hand, for $\lambda = 0$ we get $f_{\text{acd}}^{sr}(\lambda,x) = f_{\text{sum}}^{sr}(x)$, so the value of x_e^{sr} (x_N^{sr}) is equal to the local (network) maxian point z_e^{sr} (z_N^{sr}). If function $f_{\text{acd}}^{sr}(0,x)$ attains its maximum value inside a local (network) interval $[\tilde{x}_e^{sr},\hat{x}_e^{sr}]$ ($[\tilde{x}_N^{sr},\hat{x}_N^{sr}]$), then this interval matches the local (network) maxian interval $[\tilde{z}_e^{sr},\hat{z}_e^{sr}]$ ($[\tilde{z}_N^{sr},\hat{z}_N^{sr}]$). From these results and the earlier properties, we can now derive the following Lemma concerning the set of candidate points inside an edge.

Lemma VII.5. *Given an edge $e \in E$ and a value of λ, $0 \le \lambda \le 1$, the local anti-cent-dian points fall inside the interval $[\min\{y_e^{sr},\tilde{z}_e^{sr}\},\max\{y_e^{sr},\hat{z}_e^{sr}\}]$, with $1 \le s \le p$ and $1 \le r \le q$.*

Proof. When $\lambda = 0$ and $\lambda = 1$, the local anti-cent-dian points are, respectively, $[\tilde{z}_e^{sr},\hat{z}_e^{sr}]$ and y_e^{sr}. Since the anti-cent-dian function is a convex combination of $f_{\min}^{sr}(x)$ and $f_{\max}^{sr}(x)$, with $1 \le s \le p$ and $1 \le r \le q$, for any other value of λ the local anti-cent-dian must be attained at a point between y_e^{sr} and $[\tilde{z}_e^{sr},\hat{z}_e^{sr}]$. Therefore, this point must fall inside $[\min\{y_e^{sr},\tilde{z}_e^{sr}\},\max\{y_e^{sr},\hat{z}_e^{sr}\}]$. ■

In the same way as the previous models, we now define the multicriteria anti-cent-dian problem as follows. Let $F_{\text{acd}}(\lambda,x) = (f_{\text{acd}}^{11}(\lambda,x), f_{\text{acd}}^{12}(\lambda,x),\ldots,f_{\text{acd}}^{pq}(\lambda,x)) \in \mathbb{R}^{p\times q}$ be the vector of values for all combinations of weights $s=1,\dots,p$ and lengths $r=1,\dots,q$. Besides, let $k=p\times q$, so we denote the λ-anti-cent-dian functions as $f^i_{\text{acd}}(\lambda,x)$, with $i=1,\dots,k$. A set $X_N\subset N$ is an efficient set for the λ-anti-cent-dian problem iff $F_{\text{acd}}(\lambda,X_N)=\max\limits_{x\in N}F_{\text{acd}}(\lambda,x)$. In this sense, given two points $x,y\in N$, we say that x dominates y ($x\succ y$) if $f^i_{\text{acd}}(\lambda,x)\geq f^i_{\text{acd}}(\lambda,x)$, $\forall i=1,\dots,k$, $0\leq\lambda\leq 1$, with at least one inequality strict. Therefore, a point $x\in N$ is an efficient or Pareto-optimal point for the multicriteria λ-anti-cent-dian problem if there is no other point $y\in N$ such that $y\succ x$.

Next, we present the properties of the multicriteria λ-anti-cent-dian problem. Given that the λ-anti-cent-dian function can attain its maximum value inside an interval, the properties are rather the same as the multicriteria maxian problem, and thus, the following result is straightforward.

Lemma VII.6. *The interval of efficient points on edge e is* $X_e=[\min\{\tilde{x}_e,\hat{x}_e\},\max\{\tilde{x}_e,\hat{x}_e\}]$, *where* $\tilde{x}_e=\max\limits_{1\leq i\leq k}\tilde{x}^i_e$ *and* $\hat{x}_e=\min\limits_{1\leq i\leq k}\hat{x}^i_e$.

Finally, we set the rule by which any edge containing no efficient points is easily removed.

Lemma VII.7. *Given λ, $0\leq\lambda\leq 1$, for any edge $e\in E$*

$$f^i_{\text{acd}}(\lambda,x^i_e)\leq UB^i_e=\lambda f^i_{\min}(y^i_e)+(1-\lambda)f^i_{\text{sum}}(\tilde{z}^i_e),\quad i=1,\dots,k \qquad \text{(VII.1)}$$

Proof. As $f^i_{\text{acd}}(1,x^i_e)=f^i_{\min}(y^i_e)$ and $f^i_{\text{acd}}(0,x^i_e)=f^i_{\text{sum}}(\tilde{z}^i_e)=f^i_{\text{sum}}(\hat{z}^i_e)$, then for $\lambda=0$ or $\lambda=1$, obviously (VII.1) is verified. Thus, we now assume $0<\lambda<1$. The following cases take place:

a) If $x^i_e\in[y^i_e,\tilde{z}^i_e]$ or $x^i_e\in[\hat{z}^i_e,y^i_e]$, and since $f^i_{\min}(x^i_e)\leq f^i_{\min}(y^i_e)$ and $f^i_{\text{sum}}(x^i_e)\leq f^i_{\text{sum}}(\tilde{z}^i_e)=f^i_{\text{sum}}(\hat{z}^i_e)$, then $f^i_{\text{acd}}(\lambda,x^i_e)=\lambda f^i_{\min}(x^i_e)+(1-\lambda)f^i_{\text{sum}}(x^i_e)\leq\lambda f^i_{\min}(y^i_e)+(1-\lambda)f^i_{\text{sum}}(\tilde{z}^i_e)$ (see Figure VII.3a,c).

b) If $x^i_e=y^i_e$ and $\tilde{z}^i_e\leq y^i_e\leq\hat{z}^i_e$ (Figure VII.3b), then $f^i_{\text{sum}}(y^i_e)=f^i_{\text{sum}}(\tilde{z}^i_e)=f^i_{\text{sum}}(\hat{z}^i_e)$. Consequently, $f^i_{\text{acd}}(\lambda,x^i_e)=\lambda f^i_{\min}(y^i_e)+(1-\lambda)f^i_{\text{sum}}(y^i_e)=\lambda f^i_{\min}(y^i_e)+(1-\lambda)f^i_{\text{sum}}(\tilde{z}^i_e)$.

Even if the λ-anti-cent-dian function attains its maximum value inside an interval $[\tilde{x}^i_e,\hat{x}^i_e]$, this result also applies since $f^i_{\text{acd}}(\lambda,\tilde{x}^i_e)=f^i_{\text{acd}}(\lambda,\hat{x}^i_e)$. ■

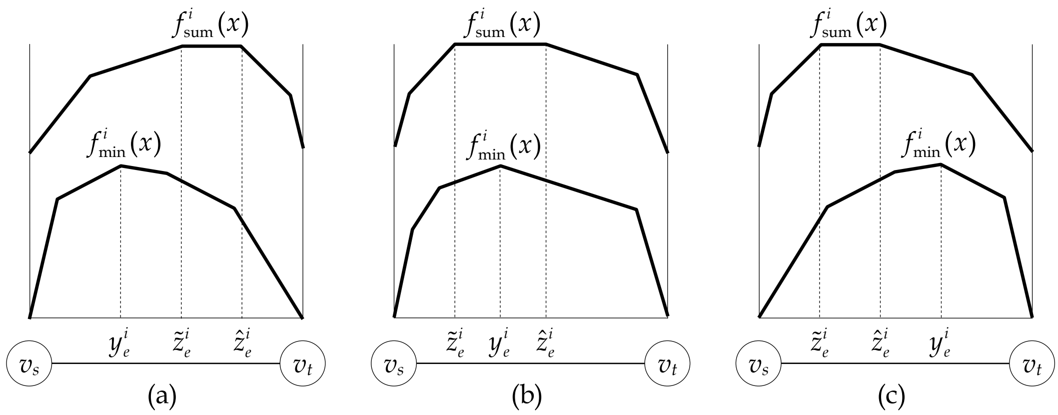


Figure VII.3: Illustration of Lemma VII.7.

Theorem VII.1. *Let* $[\tilde{x}_N^i, \hat{x}_N^i]$, $1 \le i \le k$, *be the network anti-cent-dian intervals for the k criteria. Any edge* $e = (v_s, v_t) \in E$ *fulfilling*

$$UB_e^1 = \lambda f_{\min}^1(y_e^1) + (1-\lambda) f_{\text{sum}}^1(\tilde{z}_e^1) \le f_{\text{acd}}^1(\lambda, \tilde{x}_N^i) \wedge \ldots \wedge UB_e^k = \lambda f_{\min}^k(y_e^k) + (1-\lambda) f_{\text{sum}}^k(\tilde{z}_e^k) \le f_{\text{acd}}^k(\lambda, \tilde{x}_N^i)$$

or

$$UB_e^1 = \lambda f_{\min}^1(y_e^1) + (1-\lambda) f_{\text{sum}}^1(\tilde{z}_e^1) \le f_{\text{acd}}^1(\lambda, \hat{x}_N^i) \wedge \ldots \wedge UB_e^k = \lambda f_{\min}^k(y_e^k) + (1-\lambda) f_{\text{sum}}^k(\tilde{z}_e^k) \le f_{\text{acd}}^k(\lambda, \hat{x}_N^i)$$

with at least one of the inequalities strict, contains no efficient points, and therefore, it can be deleted.

Proof. Following the result obtained for the multicriteria maxian problem in Proposition VII.4, edge *e* can be removed if it satisfies the following expression

$$f_{\text{acd}}^1(\lambda, x_e^1) \le f_{\text{acd}}^1(\lambda, \tilde{x}_N^i) \quad \wedge \quad \ldots \quad \wedge \quad f_{\text{acd}}^k(\lambda, x_e^k) \le f_{\text{acd}}^k(\lambda, \tilde{x}_N^i)$$

$$\text{or} \qquad \text{(VII.2)}$$

$$f_{\text{acd}}^1(\lambda, x_e^1) \le f_{\text{acd}}^1(\lambda, \hat{x}_N^i) \quad \wedge \quad \ldots \quad \wedge \quad f_{\text{acd}}^k(\lambda, x_e^k) \le f_{\text{acd}}^k(\lambda, \hat{x}_N^i)$$

with at least one of the inequalities strict. The right hand side of (VII.1) is an upper bound to $f_{\text{acd}}^i(\lambda, x_e^i)$. Hence, we replace this expression in (VII.2), and the result follows. ■

This previous theorem involves computing the network anti-cent-dian points in advance. However, we can avoid this computation by setting a lower bound to the value of each $f_{\text{acd}}^i(\lambda, x_N^i)$, $i = 1, \ldots, k$, as the next Lemma states.

Lemma VII.8. *For each edge* $e = (v_s, v_t)$, *a lower bound of* $f_{\text{acd}}^i(\lambda, x_e^i)$, $i = 1, \ldots, k$ *is*

$$LB_e^i = \max\{\lambda f_{\min}^i(y_e^i) + (1-\lambda) f_{\text{sum}}^i(y_e^i), \lambda \max\{f_{\min}^i(\tilde{z}_e^i), f_{\min}^i(\hat{z}_e^i)\} + (1-\lambda) f_{\text{sum}}^i(\tilde{z}_e^i)\} \qquad \text{(VII.3)}$$

Proof. If $\lambda = 0$ or $\lambda = 1$ then we get $f_{\text{acd}}^i(0, x_e^i) = f_{\text{sum}}^i(\tilde{z}_e^i) = f_{\text{sum}}^i(\hat{z}_e^i)$ and $f_{\text{acd}}^i(1, x_e^i) = f_{\min}^i(y_e^i)$, respectively. Therefore, we assume $0 < \lambda < 1$. Due to the concavity of the λ-anti-cent-dian function it follows that $f_{\text{acd}}^i(\lambda, y_e^i) = \lambda f_{\min}^i(y_e^i) + (1-\lambda) f_{\text{sum}}^i(y_e^i) < f_{\text{acd}}^i(\lambda, x_e^i)$, $f_{\text{acd}}^i(\lambda, \tilde{z}_e^i) < f_{\text{acd}}^i(\lambda, x_e^i)$ and $f_{\text{acd}}^i(\lambda, \hat{z}_e^i) < f_{\text{acd}}^i(\lambda, x_e^i)$. Besides, always $f_{\text{sum}}^i(\tilde{z}_e^i) = f_{\text{sum}}^i(\hat{z}_e^i)$, and $f_{\min}^i(\tilde{z}_e^i)$ might be different to $f_{\min}^i(\hat{z}_e^i)$, then $\lambda \max\{f_{\min}^i(\tilde{z}_e^i), f_{\min}^i(\hat{z}_e^i)\} + (1-\lambda) f_{\text{sum}}^i(\tilde{z}_e^i)\}$ is smaller than $f_{\text{acd}}^i(\lambda, x_e^i)$. Thus, a lower bound of $f_{\text{acd}}^i(\lambda, x_e^i)$ is $\max\{\lambda f_{\min}^i(y_e^i) + (1-\lambda) f_{\text{sum}}^i(y_e^i), \lambda \max\{f_{\min}^i(\tilde{z}_e^i), f_{\min}^i(\hat{z}_e^i)\} + (1-\lambda) f_{\text{sum}}^i(\tilde{z}_e^i)\}$.

Since $f_{\text{acd}}^i(\lambda, \tilde{x}_e^i) = f_{\text{acd}}^i(\lambda, \hat{x}_e^i)$, this result also applies even if the λ-anti-cent-dian function attains its maximum value inside the interval $[\tilde{x}_e^i, \hat{x}_e^i]$. ■

For each criterion $1 \le i \le k$, let

$$x_{LB}^i = \arg\max_{e \in E}\{LB_e^i\} \qquad \text{(VII.4)}$$

be the points on *N* where the network lower bound $LB_N^i = \max_{e \in E} LB_e^i$ is achieved. Obviously, $LB_N^i \le f_{\text{acd}}^i(\lambda, \tilde{x}_N^i) = f_{\text{acd}}^i(\lambda, \hat{x}_N^i)$. Now, it suffices relating Lemma VII.8 to Theorem VII.1 to get the following result.

Theorem VII.2. *Any edge* $e = (v_s, v_t) \in E$ *fulfilling for some point* x_{LB}^i, $1 \le i \le k$

$$UB_e^1 \le f_{\text{acd}}^1(\lambda, x_{LB}^i) \quad \wedge \quad \ldots \quad \wedge \quad UB_e^k \le f_{\text{acd}}^k(\lambda, x_{LB}^i)$$

with at least one of the inequalities strict, contains no efficient points, and therefore, it can be deleted.

Proof. The result follows by replacing in (VII.2) the upper bound of each edge (VII.1) and the function values of the points (VII.4) where the network lower bound is attained. ■

Note that when $\lambda = 1$, the previous theorem matches with Proposition VII.2, whereas for $\lambda = 0$ Theorem VII.2 is closely related to Proposition VII.4.

In the next section, all these preceding results, along with some results described in Chapter III, are gathered in an algorithm that we propose to solve the multicriteria λ-anti-cent-dian problem for a given value of parameter λ.

VII.6 The algorithm to solve MACDP

We now introduce the method proposed to solve MACDP, which is outlined in Algorithm VII.1. It has five input data, namely, the network $N(V,G)$, the distance matrix *d*, the number of weights per node *p*, the number of lengths per edge *q*, and the parameter λ. The method follows the guidelines established in Chapter IV for the multicriteria λ-cent-dian problem. Thus, we first define the set of points *P* and the set of segments *S*.

Then, Theorem VII.2 is applied to remove edges containing no efficient points. This is done in $\mathrm{O}(k^2mn)$ time, since the computation of UB_e^i is done in $\mathrm{O}(kmn)$ time and $f_{\mathrm{acd}}^i(\lambda, x_{LB}^j)$, $1 \le i, j \le k$, requires $\mathrm{O}(k^2mn)$ steps.

For each remaining edge *e*, and for each weight *s* and length *r* we compute functions $f_{\min}^{sr}(x)$ and $f_{\mathrm{sum}}^{sr}(x)$. Function $f_{\min}(x)$ corresponds to the lower envelope of all the *n* distance functions (see Chapter V). This lower envelope is calculated in $\mathrm{O}(n \log n)$ time (Hershberger, 1989). Being $k = p \times q$, the time to get all the $f_{\min}^{sr}(x)$ functions is $\mathrm{O}(kn \log n)$. On the other hand, all the $f_{\mathrm{sum}}^{sr}(x)$ functions can be computed in $\mathrm{O}(kn \log n)$ (see Chapter II).

From these two latter functions, the λ-anti-cent-dian function is build up in at most $\mathrm{O}(kn)$ time. Next, Lemma VII.6 is applied to get the interval of local efficient points X_e for the current edge *e*. Within this set X_e, the λ-anti-cent-dian function values of the breakpoints are used to generate the set of points *P* and the set of segments *S* in at most $\mathrm{O}(kn)$ time. Thus, the overall complexity of the outer loop for all the edges is at most $\mathrm{O}(kmn \log n)$, with $|P| \in \mathrm{O}(km)$ and $|S| \in \mathrm{O}(kmn)$.

Finally, it remains only to compare all the points in set *P* and all the segments in *S*. Comparing pairwise all the elements in *P* is performed in $\mathrm{O}(k^2m^2)$ steps. Each comparison takes $\mathrm{O}(k)$ time, and hence, Algorithm VII.2 runs in $\mathrm{O}(k^3m^2)$ time. The same analysis can be applied to Algorithm VII.3 and Algorithm VII.4 since $|S| \in \mathrm{O}(kmn)$. Then, the complexity of these two procedures is $\mathrm{O}(k^3m^2n^2)$. We remark that the *Dominate* function used in Algorithm VII.3 is described thoroughly in Chapter III.

Once all points have been compared against all segments, we obtained the set of non-dominated points P_{ND} and the set of non-dominated segments S_{ND}. Therefore, the overall complexity of Algorithm VII.1 is $\mathrm{O}(k^3m^2n^2)$. Note that this complexity is the same we obtained in Chapter IV for the multicriteria λ-cent-dian problem.

```
function MACD(Network N(V,E), DistanceMatrix d, Parameters p, q, λ)
{   Let P := ∅ be the set of candidate points to be non-dominated
    Let S := ∅ be the set of possible non-dominated segments
    Apply Theorem VII.2 to remove all edges containing no efficient points
    for all remaining edges e := (v_s, v_t) ∈ E do
        {   for s := 1 to p do
                for r := 1 to q do
                    {   if λ ≠ 0 then    Compute f^sr_min(x)
                        if λ ≠ 1 then    Compute f^sr_sum(x)
                    }
            for s := 1 to p do
                for r := 1 to q do
                    Compute f^sr_acd(λ,x) = λ f^sr_min(x) + (1−λ) f^sr_sum(x)
            Apply Lemma VII.6 to get the set of efficient points X_e
            Let x_1,...,x_j be the sorted sequence of j breakpoints for the k = p×q
                λ-anti-cent-dian functions inside X_e
            if j = 1 then P := P ∪ {x_1}
            else
                for i := 1 to j−1 do
                    {   Let [x_i, x_{i+1}] be a segment of edge e within X_e
                        S := S ∪ {[x_i, x_{i+1}]}
                    }
        }
    // Let P_ND the set of non-dominated points and S_ND the set of non-dominated segments.
    P_ND := PointComparison(P)
    S_ND := SegmentComparison(S)
    (P_ND, S_ND) := PointAgainstSegmentComparison(P_ND, S_ND)
    return P_ND and S_ND
}
```

Algorithm VII.1: The multicriteria λ-anti-cent-dian function.

```
function PointComparison(PointSet P)
{   Let {x_1, x_2, ..., x_p} be the points belonging to P
    Let P_ND := ∅ be the set of non-dominated points.
    for i := 1 to p do
        {   Let x_i ∈ P be a point
            if ∄x_j ∈ P_ND : x_j ≻ x_i then
                {   P_ND := P_ND ∪ {x_i}
                    if ∃x_k ∈ P_ND : x_i ≻ x_k then
                        P_ND := P_ND / {x_k}
                }
        }
    return P_ND
}
```

Algorithm VII.2: The point comparison function.

```
function SegmentComparison(SegmentSet S)
{   S_ND := S
    for all segments X := [x_0, x_1] ∈ S_ND do
        for all segments Y := [y_0, y_1] ∈ S_ND successors in S_ND to X do
            {   for i := 1 to k do
                    {   Create inequality y(x)
                        T := T ∪ y(x)
                    }
                Dominate(T, X, Y)
                if X ≻ Y then   Y := Y / [y_min, y_max]
                Change inequalities y(x) to x(y)
                Dominate(T, Y, X)
                if Y ≻ X then   X := X / [x_min, x_max]
        }
    return S_ND
}
```

Algorithm VII.3: The segment comparison function.

```
function PointAgainstSegmentComparison(PointSet P_ND, SegmentSet S_ND)
{   for all points z ∈ P_ND do
        for all segments X := [x_0, x_1] ∈ S_ND do
            {   if z ≻ X then
                    {   Let [x_min, x_max] ∈ X be the interval dominated by point z
                        X := X / [x_min, x_max]
                    }
                if X ≻ z then P_ND := P_ND / {z}
            }
    return P_ND and S_ND
}
```

Algorithm VII.4: Comparing points against segments.

VII.7 An example

Figure VII.4 shows a random planar network with $n = 7$ nodes, $m = 15$ edges, $p = 2$ weights per node and $q = 2$ lengths per edge. Thus, we have $k = 4$ criteria. Beside each node $v_i \in V$ we placed (in bold) two integer weights (w_i^1, w_i^2) randomly generated in the interval $[1,5]$. Likewise, each edge $e = (v_s, v_t) \in E$ is labeled (in italics) with two integer lengths (l_e^1, l_e^2) randomly ranging in the interval $[1,25]$. We set the parameter λ to 0.5.

The algorithm begins by removing all edges that contain no efficient points. In this sense, we need to compute, for each criterion $i = 1, \dots, k$, the upper bounds UB_e^i for each edge *e* as well as the network lower bounds LB_N^i. Table VII.1 shows the points x_{LB}^i where the network lower bounds are achieved for each criterion characterized by the weight index *s* and the length index *r*, along with their function values.

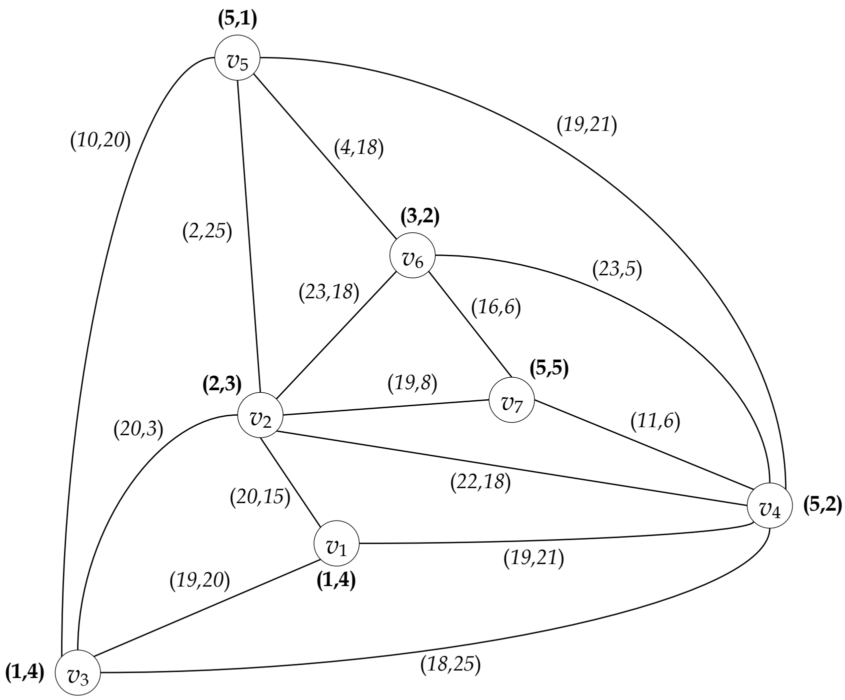


Figure VII.4: A network with two lengths per edge and two weights per node.

s	r	x^i_{LB}	*Edge*	$f_{\text{acd}}(\lambda = 0.5, x^i_{LB})$
1	1	$x^1_{LB} = 8.5$	(v_1, v_3)	($\boldsymbol{LB^1_N}$ **= 294.25**, 273.5, 244, 203.368)
1	2	$x^2_{LB} = 4$	(v_1, v_3)	(288.5, $\boldsymbol{LB^2_N}$ **= 296**, 233.9, 217)
2	1	$x^3_{LB} = 12.5$	(v_2, v_6)	(255.25, 181.413, $\boldsymbol{LB^3_N}$ **= 257.5**, 168.109)
2	2	$x^4_{LB} = 21$	(v_2, v_5)	(124.86, 236.5, 141.28, $\boldsymbol{LB^4_N}$ **= 272.5**)

Table VII.1: The points where the network lower bounds LB^i_N are achieved for each criterion $i = 1, \ldots, k$.

Once these values are computed, Theorem VII.2 is applied on each edge *e*. Table VII.2 summarizes the removal process. Only 8 out of the 15 initial edges remain after the deletion, namely: (v_1, v_3), (v_1, v_4), (v_2, v_5), (v_2, v_6), (v_3, v_4), (v_3, v_5), (v_4, v_5) and (v_5, v_6). On this set of remaining edges we now proceed to compute, for each combination of weights and lengths, the functions $f^{sr}_{\min}(x)$ and $f^{sr}_{\text{sum}}(x)$. Subsequently, given the parameter $\lambda = 0.5$ we calculate the λ-anti-cent-dian functions $f^{sr}_{\text{acd}}(\lambda, x)$.

Then, we apply Lemma VII.6 to get the intervals which contain the local efficient points. Table VII.3 shows, for each remaining edge *e*, both the efficient local interval X_e and the breakpoints of the *k* λ-anti-cent-dian functions within such interval. These breakpoints are joined in pairs to form the intervals $[x_i, x_{i+1}]$ that are added to the set of segments *S*.

Finally, it suffices to compare pairwise all the points in set *P* and all the segments in set *S*. Since set *P* is empty, we only have to compare all the segments in *S*, and thus, the solution is the set of non-dominated segments S_{ND}, which are located on 5 edges only. The set of efficient points is shown in Table VII.4 and is also drawn in bold on the partial network of Figure VII.5.

In the next section we present the computational experience performed to test the goodness of both the removal edge rule and the proposed algorithm.

Edge	UB_e	*Removal process*
(v_1, v_2)	$(261.667, 272.5, 229.3, 194.857)$	Dominated by x_{LB}^1 and x_{LB}^2 : Removed
(v_1, v_3)	$(294.75, 299, 245.35, 225)$	**Not removed**
(v_1, v_4)	$(267.833, 282.75, 246.083, 213)$	**Not removed**
(v_2, v_3)	$(248.667, 158.5, 225.5, 114.571)$	Dominated by x_{LB}^1, x_{LB}^2 and x_{LB}^3 : Removed
(v_2, v_4)	$(191.5, 198.75, 222, 173.3)$	Dominated by x_{LB}^1 and x_{LB}^2 : Removed
(v_2, v_5)	$(131.429, 250.667, 142.5, 279.875)$	**Not removed**
(v_2, v_6)	$(255.25, 198.375, 257.5, 173.05)$	**Not removed**
(v_2, v_7)	$(179.917, 147.083, 208.75, 111.5)$	Dominated by x_{LB}^1, x_{LB}^2 and x_{LB}^3 : Removed
(v_3, v_4)	$(202, 237.417, 208.083, 218)$	**Not removed**
(v_3, v_5)	$(189.5, 233.333, 171.75, 257.5)$	**Not removed**
(v_4, v_5)	$(166.75, 215.5, 201.083, 271)$	**Not removed**
(v_4, v_6)	$(200.75, 125.188, 237.5, 149.75)$	Dominated by x_{LB}^1 and x_{LB}^3 : Removed
(v_4, v_7)	$(182.583, 125.583, 195.857, 131.786)$	Dominated by x_{LB}^1, x_{LB}^2 and x_{LB}^3 : Removed
(v_5, v_6)	$(137.1, 208.25, 155.333, 262)$	**Not removed**
(v_6, v_7)	$(166.5, 122.125, 203.083, 140.286)$	Dominated by x_{LB}^1, x_{LB}^2 and x_{LB}^3 : Removed

Table VII.2: Removal of inefficient edges for the network shown in Figure VII.4.

Edge	X_e	*Breakpoints within* X_e
(v_1, v_3)	$[3.8, 8.5]$	$x_1 = 3.8, 5.225, 5.5, 5.8, 7.6, 8.5 = x_6$
(v_1, v_4)	$[1.90476, 8.5]$	$x_1 = 1.90476, 6.66667, 8.5 = x_3$
(v_2, v_5)	$[0, 1.64]$	$x_1 = 0, 0.08, 0.24, 0.5, 1.16, 1.28, 1.42857, 1.5, 1.62667, 1.64 = x_{10}$
(v_2, v_6)	$[8.30556, 12.5]$	$x_1 = 8.30556, 8.5, 9.2, 10, 10.2222, 12, 12.5 = x_7$
(v_3, v_4)	$[7.2, 13.5]$	$x_1 = 7.2, 8.66667, 9.36, 9.5, 10.08, 12.96, 13.5 = x_7$
(v_3, v_5)	$[0.5, 8.25]$	$x_1 = 0.5, 2, 5.25, 5.5, 6, 6.5, 8, 8.25 = x_8$
(v_4, v_5)	$[5.5, 15.381]$	$x_1 = 5.5, 6.33333, 10.5, 10.8571, 13, 13.5714, 14, 14.4762, 15.381 = x_9$
(v_5, v_6)	$[0.666667, 4]$	$x_1 = 0.666667, 1, 1.11111, 1.33333, 1.6, 1.66667, 2.11111, 2.66667, 4 = x_9$

Table VII.3: For each edge not removed, we show the local set of efficient points X_e and the breakpoints of all the k λ-anti-cent-dian functions with respect to the first length.

Edge	*Efficient points*
(v_1, v_3)	$[3.8, 8.5]$
(v_2, v_5)	$[1.00923, 1.64]$
(v_2, v_6)	$[9.61111, 12.5]$
(v_3, v_5)	$[5.28571, 8.25]$
(v_4, v_5)	$[7.9418, 15.381]$

Table VII.4: Set of efficient points of the network shown in Figure VII.4.

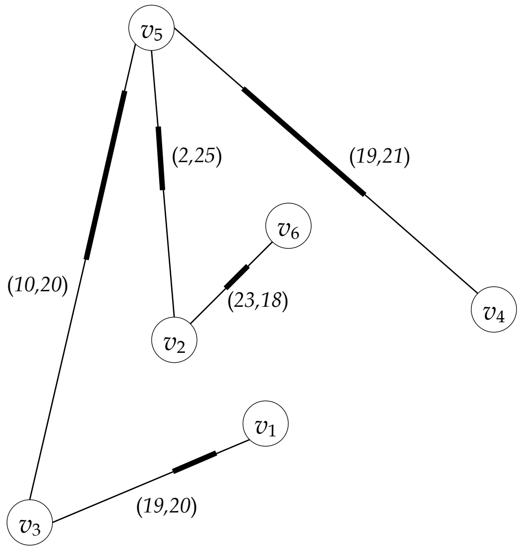


Figure VII.5: Efficient points are drawn in bold on the partial network.

VII.8 Computational results

We have programmed Algorithm VII.1 in C++ programming language (GNU g++ 2.95.2) using the class library LEDA 4.2.1, on a two 1.2 Ghz processor Pentium III with 1 Gb of RAM under Red Hat Linux 7.3 (Valhalla).

Two kinds of experiments were performed. In both of them, random planar networks were generated with $m = 3n - 6$ edges using the generators developed by LEDA. Likewise, parameter λ varies from $\lambda = 0$ (maxian problem) to $\lambda = 1$ (uncenter problem) with a step of 0.25. Both the number of weights per node p and the number of lengths per edge q range between 1 and 3. Ten instances were generated for each combination of the latter parameters. The weight values are random integers uniformly distributed in the interval $[1,10]$, whereas the edge lengths are random integers in the range $[1,50]$. We remark that calculation of the distance matrix was not included in the total computing time.

In the first experiment, random planar networks were generated with $n = 10$ up to 100 in steps of 10 nodes. Table VII.5 shows the average times. Regardless of the number of nodes n, note that the computing time grows as both p and q increase. The average percentage of edges deleted by Theorem VII.2 is shown in Table VII.6. In most cases the number of removed edges is very high, achieving in some instances 99% of deletion. This issue becomes quite remarkable when $p = q = 1$ (single criterion). In this particular case, the bounds seem to be very tight, and thus, the removal rule becomes very effective since over 95% of the edges are deleted, leaving only those edges that contain the final optimal points.

On the other hand, we also wanted to test the performance of the new algorithm on bigger networks. Accordingly, the second experiment involved generating random planar networks with $n = 50$ to 500, with a step of 50 nodes. Table VII.7 shows the average computing times, whereas Table VII.8 presents the average percentage of removed edges. Note that for $p = q = 1$, the percentage of deletion in all cases is over 99%. However, when $p = q = 3$, the average edge removal percentage is greater for $\lambda = 0$ than for $\lambda = 1$, and hence, the average times in the latter are higher. Anyhow, the average computing time never exceeds one minute, not even for the largest networks.

Finally, Figure VII.6 graphically summarizes the last experiment. Observe that, obviously, the computing times (left) polynomially increase with *n*, *p* and *q*. We also remark that, when $p = q = 1$, the number of processed edges (right) is rather low (diamond line near to zero). As we previously commented, in the case of $p = q = 3$, solving the multicriteria uncenter problem ($\lambda = 1$) requires significantly much more time than the multicriteria maxian problem ($\lambda = 0$).

VII.9 Conclusions and discussion

In the first part of this chapter we have analyzed the uncenter and maxian problems on multicriteria networks, namely, networks holding several weights on the nodes and several lengths on the edges. New properties were established together with the rules to remove edges containing no efficient points.

Through a parameter λ, the convex combination of these two latter problems was addressed as the multicriteria λ-anti-cent-dian problem. We propose a rule to delete inefficient edges and a polynomial algorithm in $\mathrm{O}(k^3 m^2 n^2)$ time to solve this problem. Besides, for $\lambda = 0$ we can solve the multicriteria maxian problem, whereas for $\lambda = 1$ we can obtain the solution for the multicriteria uncenter problem. Furthermore, when $p = q = 1$ this procedure can even solve the single criterion uncenter, maxian or anti-cent-dian problem. The computational experience strengthens the polynomial complexity of the algorithm as well as the effectiveness of the rule to eliminate the inefficient edges.

This model could be slightly modified to generalize other models studied in the literature. For instance, if we define a set of *k* parameters $\Lambda = \{\lambda^1, \ldots, \lambda^k\}$ then we could deal with each function $f^i_{\mathrm{acd}}(\lambda^i, x)$ independently. Thus, the problem proposed by Zhang and Melachrinoudis (2001) might be denoted as $\max_{x \in N}(f^1_{\mathrm{acd}}(\lambda^1, x), f^2_{\mathrm{acd}}(\lambda^2, x))$, with $p = 2$, $q = 1$, $k = p \times q = 2$ and $\Lambda = \{\lambda^1 = 1, \lambda^2 = 0\}$. On the other hand, the multicriteria semi-obnoxious median problem presented by Hamacher, Labbé, Nickel, and Skriver (2002) can be formulated as $\max_{x \in N}(f^i_{\mathrm{acd}}(\lambda^i, x), -f^j_{\mathrm{acd}}(\lambda^j, x))$, with $p > 1$, $q = 1$, $\lambda^i = \lambda^j = 0$ and $i \in Q_1$, $j \in Q_2$, $|Q_1 \cup Q_2| = p$, $Q_1 \cap Q_2 = \varnothing$, being Q_1 the set of obnoxious objective functions, and Q_2 the set of desirable objective functions. Obviously, if $Q_2 = \varnothing$ then we get the multicriteria maxian problem discussed in this chapter.

Finally, we remark that if $p > 1$ and $q = 1$ then the number of criteria matches the number of weights per node, i.e., $k = p$. Besides, if $\lambda = 0$ then the number of breakpoints for all the *k* objective functions of a given edge is $\mathrm{O}(n)$, since all the $f^{s1}_{\mathrm{sum}}(x)$ functions share the same breakpoints. Hence, the total number of segments to compare is $\mathrm{O}(mn)$. Therefore, the overall complexity of the algorithm is reduced to $\mathrm{O}(km^2 n^2)$, which is the same complexity achieved by Hamacher, Labbé, Nickel, and Skriver (2002) for the location of a semi-obnoxious facility on networks with sum objectives.

		λ = 0								
		$p=1$			$p=2$			$p=3$		
n	m	$q=1$	$q=2$	$q=3$	$q=1$	$q=2$	$q=3$	$q=1$	$q=2$	$q=3$
10	24	0.01	0.01	0.04	0.00	0.01	0.09	0.00	0.02	0.14
20	54	0.01	0.02	0.09	0.01	0.03	0.09	0.01	0.03	0.17
30	84	0.02	0.04	0.10	0.02	0.06	0.17	0.03	0.05	0.34
40	114	0.02	0.04	0.16	0.03	0.08	0.30	0.05	0.06	0.35
50	144	0.04	0.06	0.24	0.05	0.11	0.29	0.06	0.14	0.21
60	174	0.04	0.07	0.34	0.06	0.13	0.45	0.07	0.14	0.40
70	204	0.05	0.08	0.26	0.06	0.13	0.42	0.09	0.18	0.62
80	234	0.05	0.10	0.33	0.08	0.16	0.64	0.10	0.21	0.99
90	264	0.09	0.14	0.43	0.12	0.23	0.64	0.15	0.29	0.65
100	294	0.09	0.17	0.58	0.13	0.25	0.73	0.17	0.34	1.08

		λ = 0.25								
		$p=1$			$p=2$			$p=3$		
n	m	$q=1$	$q=2$	$q=3$	$q=1$	$q=2$	$q=3$	$q=1$	$q=2$	$q=3$
10	24	0.01	0.01	0.06	0.01	0.02	0.06	0.01	0.05	0.30
20	54	0.01	0.02	0.09	0.01	0.04	0.30	0.01	0.07	0.33
30	84	0.02	0.04	0.19	0.03	0.08	0.29	0.03	0.10	0.43
40	114	0.03	0.06	0.15	0.03	0.09	0.32	0.04	0.10	0.53
50	144	0.04	0.07	0.13	0.06	0.15	0.57	0.07	0.24	0.79
60	174	0.06	0.12	0.25	0.06	0.25	0.60	0.10	0.27	0.96
70	204	0.06	0.16	0.50	0.08	0.26	0.73	0.11	0.28	0.78
80	234	0.06	0.18	0.52	0.11	0.23	0.90	0.13	0.33	1.21
90	264	0.10	0.23	0.71	0.15	0.33	1.10	0.20	0.53	1.57
100	294	0.11	0.25	0.71	0.16	0.35	1.38	0.21	0.69	1.64

		λ = 0.5								
		$p=1$			$p=2$			$p=3$		
n	m	$q=1$	$q=2$	$q=3$	$q=1$	$q=2$	$q=3$	$q=1$	$q=2$	$q=3$
10	24	0.01	0.02	0.03	0.01	0.01	0.07	0.01	0.03	0.05
20	54	0.01	0.03	0.08	0.02	0.03	0.26	0.02	0.08	0.27
30	84	0.02	0.03	0.22	0.03	0.05	0.47	0.03	0.12	0.17
40	114	0.03	0.08	0.31	0.04	0.12	0.44	0.05	0.09	0.65
50	144	0.05	0.08	0.39	0.06	0.12	0.50	0.07	0.22	0.55
60	174	0.06	0.12	0.54	0.07	0.17	0.78	0.08	0.27	1.14
70	204	0.05	0.18	0.50	0.08	0.26	0.91	0.11	0.27	1.30
80	234	0.07	0.16	0.65	0.10	0.23	0.83	0.13	0.45	1.44
90	264	0.10	0.19	0.76	0.14	0.35	1.47	0.18	0.54	1.81
100	294	0.12	0.23	0.62	0.16	0.39	1.14	0.20	0.55	1.73

		λ = 0.75								
		$p=1$			$p=2$			$p=3$		
n	m	$q=1$	$q=2$	$q=3$	$q=1$	$q=2$	$q=3$	$q=1$	$q=2$	$q=3$
10	24	0.01	0.01	0.04	0.01	0.01	0.11	0.01	0.03	0.12
20	54	0.02	0.02	0.08	0.01	0.03	0.16	0.02	0.03	0.19
30	84	0.02	0.03	0.18	0.03	0.10	0.25	0.03	0.08	0.59
40	114	0.03	0.06	0.18	0.04	0.09	0.45	0.04	0.14	0.34
50	144	0.05	0.09	0.29	0.05	0.15	0.55	0.08	0.13	0.84
60	174	0.05	0.15	0.39	0.07	0.19	0.75	0.08	0.22	1.17
70	204	0.06	0.19	0.50	0.08	0.22	0.93	0.11	0.30	0.95
80	234	0.07	0.22	0.56	0.10	0.34	1.35	0.14	0.34	1.40
90	264	0.11	0.22	0.66	0.14	0.36	1.29	0.18	0.46	1.59
100	294	0.12	0.22	0.65	0.16	0.47	0.99	0.22	0.64	2.12

		λ = 1								
		$p=1$			$p=2$			$p=3$		
n	m	$q=1$	$q=2$	$q=3$	$q=1$	$q=2$	$q=3$	$q=1$	$q=2$	$q=3$
10	24	0.00	0.01	0.01	0.00	0.01	0.04	0.01	0.05	0.12
20	54	0.01	0.01	0.03	0.02	0.06	0.18	0.02	0.12	0.41
30	84	0.02	0.02	0.04	0.03	0.07	0.29	0.03	0.21	0.56
40	114	0.01	0.03	0.09	0.03	0.06	0.42	0.04	0.20	1.33
50	144	0.04	0.06	0.10	0.07	0.14	0.45	0.07	0.35	2.09
60	174	0.04	0.06	0.11	0.05	0.10	0.67	0.07	0.47	1.91
70	204	0.05	0.09	0.16	0.05	0.23	0.53	0.07	0.54	2.88
80	234	0.04	0.07	0.15	0.07	0.26	0.74	0.09	0.48	3.28
90	264	0.08	0.13	0.25	0.12	0.26	1.15	0.16	0.69	3.86
100	294	0.08	0.15	0.28	0.11	0.42	1.23	0.15	1.05	4.21

Table VII.5: Average computing time results for planar networks with $n = 10$ up to 100 nodes.

		$\lambda = 0$								
		$p = 1$			$p = 2$			$p = 3$		
n	m	$q = 1$	$q = 2$	$q = 3$	$q = 1$	$q = 2$	$q = 3$	$q = 1$	$q = 2$	$q = 3$
10	24	95.83	75.83	43.33	94.17	70.83	25.00	92.50	63.33	30.00
20	54	98.15	65.93	52.78	98.15	70.74	63.89	98.15	83.70	46.30
30	84	98.81	70.71	57.38	97.86	55.12	69.88	98.81	88.10	43.81
40	114	99.12	89.47	63.77	98.25	72.54	60.61	99.12	98.25	60.00
50	144	99.31	75.35	66.04	99.31	79.72	81.67	99.31	80.69	89.58
60	174	99.43	90.40	58.51	99.43	68.51	58.62	99.43	82.76	78.74
70	204	99.51	87.06	74.22	99.51	83.38	72.60	99.22	85.69	66.37
80	234	99.57	90.00	74.87	99.44	84.02	68.80	99.32	85.81	59.79
90	264	99.62	85.49	73.52	99.55	81.97	73.67	99.62	86.06	79.24
100	294	99.66	85.14	65.88	99.66	85.54	75.03	99.63	81.05	66.84

		$\lambda = 0.25$								
		$p = 1$			$p = 2$			$p = 3$		
n	m	$q = 1$	$q = 2$	$q = 3$	$q = 1$	$q = 2$	$q = 3$	$q = 1$	$q = 2$	$q = 3$
10	24	95.83	89.17	9.17	95.83	84.17	50.00	95.83	24.17	21.67
20	54	98.15	77.78	71.48	96.67	82.22	56.67	98.15	69.81	51.30
30	84	98.81	82.14	61.90	98.81	83.33	67.74	98.10	77.50	64.40
40	114	99.12	82.81	74.82	99.12	86.67	73.07	99.12	90.96	71.49
50	144	99.31	86.74	88.68	99.31	78.54	72.01	99.31	75.21	63.40
60	174	99.43	81.90	84.31	99.43	71.55	71.32	99.20	82.07	72.99
70	204	99.51	76.03	66.81	99.51	80.29	76.67	99.36	89.80	82.79
80	234	99.57	76.88	74.66	99.49	90.30	74.40	99.49	90.47	75.00
90	264	99.62	84.36	68.37	99.62	86.97	71.74	99.47	82.77	71.33
100	294	99.66	85.65	74.39	99.56	90.92	71.97	99.63	78.37	74.93

		$\lambda = 0.5$								
		$p = 1$			$p = 2$			$p = 3$		
n	m	$q = 1$	$q = 2$	$q = 3$	$q = 1$	$q = 2$	$q = 3$	$q = 1$	$q = 2$	$q = 3$
10	24	95.83	45.00	59.17	95.83	75.00	36.67	89.17	70.00	69.17
20	54	98.15	67.22	62.96	98.15	87.04	46.85	95.93	65.74	56.67
30	84	98.81	92.26	52.86	98.81	91.19	57.62	98.81	75.95	88.93
40	114	99.12	67.54	60.70	99.12	76.14	68.68	98.86	95.44	66.58
50	144	99.31	87.78	59.93	98.75	88.26	70.49	98.06	82.64	78.54
60	174	99.43	78.39	56.26	99.25	87.18	66.26	99.25	80.06	65.00
70	204	99.51	73.24	66.13	99.31	77.94	61.52	99.36	90.20	68.58
80	234	99.57	85.98	70.38	99.44	91.03	68.63	99.49	78.08	71.07
90	264	99.62	86.70	60.68	99.51	83.86	60.49	99.62	80.98	68.94
100	294	99.66	86.46	76.53	99.66	87.07	73.91	99.56	87.45	74.01

		$\lambda = 0.75$								
		$p = 1$			$p = 2$			$p = 3$		
n	m	$q = 1$	$q = 2$	$q = 3$	$q = 1$	$q = 2$	$q = 3$	$q = 1$	$q = 2$	$q = 3$
10	24	95.83	71.25	65.00	90.42	95.42	51.67	92.08	59.17	42.92
20	54	98.15	87.04	67.22	98.15	84.07	56.48	96.85	92.22	67.04
30	84	98.81	89.40	63.93	98.81	62.02	64.64	98.33	88.10	55.24
40	114	99.12	85.09	74.30	99.12	85.96	67.28	98.68	80.61	80.96
50	144	99.31	86.94	66.60	99.03	84.10	75.14	99.03	96.53	67.01
60	174	99.43	68.16	68.85	99.25	84.48	70.29	99.25	87.87	64.14
70	204	99.36	68.87	71.47	99.22	85.69	67.84	99.31	86.47	73.97
80	234	99.57	70.34	72.26	99.57	77.18	59.87	99.32	90.64	71.84
90	264	99.62	83.94	71.33	99.62	84.66	70.19	99.51	88.33	69.77
100	294	99.66	87.41	76.43	99.59	78.74	81.05	99.56	81.70	67.86

		$\lambda = 1$								
		$p = 1$			$p = 2$			$p = 3$		
n	m	$q = 1$	$q = 2$	$q = 3$	$q = 1$	$q = 2$	$q = 3$	$q = 1$	$q = 2$	$q = 3$
10	24	95.83	45.42	64.17	95.83	56.25	22.92	87.08	33.75	32.08
20	54	98.15	79.26	54.07	85.19	40.93	23.15	81.85	50.00	38.89
30	84	98.57	92.86	72.02	67.74	65.48	31.90	94.05	47.86	38.45
40	114	99.12	84.91	45.00	85.44	78.51	32.54	89.30	58.77	31.32
50	144	99.31	77.50	67.64	72.57	68.89	48.54	89.86	56.11	25.35
60	174	99.43	83.74	75.46	94.77	88.28	41.90	89.25	45.63	38.10
70	204	99.51	75.34	66.76	96.67	67.25	57.60	94.85	53.92	37.65
80	234	99.57	94.36	79.96	94.06	66.97	57.01	94.10	71.97	36.50
90	264	99.62	85.34	70.23	92.61	79.09	45.23	88.41	60.83	35.11
100	294	99.66	82.62	72.55	97.55	65.10	49.93	93.88	47.21	39.86

Table VII.6: Average percentage of edges removed by Theorem VII.2 for planar networks with n = 10 up to 100 nodes.

		$\lambda = 0$								
		$p = 1$			$p = 2$			$p = 3$		
n	m	$q = 1$	$q = 2$	$q = 3$	$q = 1$	$q = 2$	$q = 3$	$q = 1$	$q = 2$	$q = 3$
50	144	0.04	0.06	0.24	0.05	0.11	0.29	0.06	0.14	0.21
100	294	0.09	0.14	0.66	0.13	0.27	0.91	0.17	0.34	0.90
150	444	0.13	0.29	0.81	0.23	0.41	1.35	0.30	0.59	1.41
200	594	0.26	0.47	1.28	0.38	0.72	1.80	0.52	0.98	2.24
250	744	0.33	0.59	1.25	0.52	1.04	2.42	0.71	1.42	3.41
300	894	0.42	0.82	2.19	0.71	1.63	3.52	1.02	2.17	4.69
350	1044	0.65	1.15	3.26	1.05	2.00	4.52	1.43	3.11	6.19
400	1194	0.75	1.40	3.04	1.27	2.46	4.82	1.79	3.45	7.15
450	1344	0.90	1.75	4.02	1.53	3.05	5.43	2.19	4.94	8.69
500	1494	1.04	2.14	4.04	1.80	3.80	8.19	2.59	5.46	10.17

		$\lambda = 0.25$								
		$p = 1$			$p = 2$			$p = 3$		
n	m	$q = 1$	$q = 2$	$q = 3$	$q = 1$	$q = 2$	$q = 3$	$q = 1$	$q = 2$	$q = 3$
50	144	0.04	0.07	0.13	0.06	0.15	0.57	0.07	0.24	0.79
100	294	0.12	0.24	0.87	0.17	0.35	1.30	0.21	0.50	2.09
150	444	0.18	0.44	1.16	0.30	0.83	2.30	0.40	1.21	2.53
200	594	0.31	0.77	2.62	0.52	1.38	3.62	0.71	1.78	4.25
250	744	0.42	0.97	3.00	0.74	1.80	5.64	1.03	2.75	8.61
300	894	0.56	1.50	3.76	1.00	2.82	5.37	1.46	4.03	8.77
350	1044	0.84	1.98	4.61	1.42	3.36	6.74	2.02	5.28	10.93
400	1194	1.01	2.36	5.50	1.77	4.40	8.40	2.54	7.79	11.83
450	1344	1.19	2.79	6.89	2.17	5.69	14.53	3.12	7.25	13.81
500	1494	1.41	3.13	8.56	2.57	6.80	15.44	3.80	9.71	23.80

		$\lambda = 0.5$								
		$p = 1$			$p = 2$			$p = 3$		
n	m	$q = 1$	$q = 2$	$q = 3$	$q = 1$	$q = 2$	$q = 3$	$q = 1$	$q = 2$	$q = 3$
50	144	0.05	0.08	0.39	0.06	0.12	0.50	0.07	0.22	0.55
100	294	0.11	0.27	0.73	0.16	0.42	1.12	0.21	0.57	2.06
150	444	0.18	0.37	0.96	0.29	0.95	2.11	0.39	1.02	3.27
200	594	0.30	0.73	1.69	0.50	1.25	3.51	0.70	1.86	5.19
250	744	0.41	0.89	2.15	0.72	1.81	4.68	1.01	3.17	6.23
300	894	0.56	1.47	3.35	1.02	2.36	6.91	1.46	4.20	8.60
350	1044	0.84	1.95	4.41	1.42	3.44	9.59	1.99	4.62	9.44
400	1194	1.03	2.22	5.69	1.76	5.07	9.62	2.54	5.48	12.31
450	1344	1.21	3.35	6.64	2.16	6.25	11.33	3.11	9.91	18.06
500	1494	1.44	4.56	8.02	2.60	7.24	14.71	3.76	9.81	20.31

		$\lambda = 0.75$								
		$p = 1$			$p = 2$			$p = 3$		
n	m	$q = 1$	$q = 2$	$q = 3$	$q = 1$	$q = 2$	$q = 3$	$q = 1$	$q = 2$	$q = 3$
50	144	0.05	0.09	0.29	0.05	0.15	0.55	0.08	0.13	0.84
100	294	0.12	0.25	0.72	0.15	0.46	1.18	0.20	0.56	2.38
150	444	0.18	0.46	1.40	0.28	1.00	2.46	0.40	0.88	3.18
200	594	0.31	0.79	2.53	0.52	1.60	3.21	0.72	1.97	5.02
250	744	0.40	1.11	2.80	0.71	1.87	4.77	1.03	2.60	7.19
300	894	0.56	1.88	4.12	1.02	2.66	6.46	1.45	4.42	8.95
350	1044	0.82	1.90	4.14	1.44	3.82	7.47	2.03	5.15	12.24
400	1194	1.04	2.81	4.24	1.79	4.85	10.23	2.52	6.38	15.09
450	1344	1.20	2.90	7.14	2.16	6.38	10.76	3.07	8.33	17.95
500	1494	1.42	4.20	7.84	2.58	7.04	11.95	3.76	10.68	23.95

		$\lambda = 1$								
		$p = 1$			$p = 2$			$p = 3$		
n	m	$q = 1$	$q = 2$	$q = 3$	$q = 1$	$q = 2$	$q = 3$	$q = 1$	$q = 2$	$q = 3$
50	144	0.04	0.06	0.10	0.07	0.14	0.45	0.07	0.35	2.09
100	294	0.08	0.14	0.38	0.15	0.48	1.39	0.16	0.83	4.67
150	444	0.09	0.23	0.59	0.24	0.72	2.17	0.35	1.70	8.32
200	594	0.19	0.38	0.87	0.38	1.37	3.38	0.59	2.48	12.90
250	744	0.22	0.49	1.32	0.52	1.77	4.98	0.51	3.22	17.52
300	894	0.26	0.60	2.10	0.67	2.71	7.50	0.98	5.09	20.68
350	1044	0.42	1.51	1.82	0.95	2.82	8.78	1.44	6.41	28.28
400	1194	0.49	1.56	2.90	0.97	4.15	11.52	1.63	7.98	29.08
450	1344	0.56	1.55	3.28	1.70	3.95	10.28	1.83	8.87	35.32
500	1494	0.65	2.10	4.35	1.21	4.97	15.61	3.60	15.48	45.72

Table VII.7: Average computing time results for planar networks with $n = 50$ up to 500 nodes.

		$\lambda = 0$								
		$p = 1$			$p = 2$			$p = 3$		
n	m	$q = 1$	$q = 2$	$q = 3$	$q = 1$	$q = 2$	$q = 3$	$q = 1$	$q = 2$	$q = 3$
50	144	99.31	75.35	66.04	99.31	79.72	81.67	99.31	80.69	89.58
100	294	99.66	92.89	66.60	99.39	76.77	67.28	99.66	81.29	73.33
150	444	99.77	78.11	76.35	99.71	91.67	70.92	99.73	89.84	79.68
200	594	99.83	84.85	66.90	99.81	91.03	76.43	99.80	94.16	79.33
250	744	99.87	89.10	86.53	99.87	88.52	78.91	99.87	94.15	78.58
300	894	99.89	89.27	74.73	99.87	79.27	69.50	99.88	89.36	73.18
350	1044	99.90	91.94	60.99	99.90	91.02	74.71	99.90	84.03	73.53
400	1194	99.92	91.38	80.56	99.90	92.13	81.43	99.88	96.24	81.03
450	1344	99.93	88.86	71.85	99.93	92.45	86.73	99.91	81.21	80.92
500	1494	99.93	84.48	82.06	99.93	89.05	71.33	99.93	90.19	80.39

		$\lambda = 0.25$								
		$p = 1$			$p = 2$			$p = 3$		
n	m	$q = 1$	$q = 2$	$q = 3$	$q = 1$	$q = 2$	$q = 3$	$q = 1$	$q = 2$	$q = 3$
50	144	99.31	86.74	88.68	99.31	78.54	72.01	99.31	75.21	63.40
100	294	99.66	84.39	66.90	99.66	92.07	73.57	99.66	91.19	70.92
150	444	99.77	85.29	78.72	99.71	84.93	70.50	99.75	83.99	83.47
200	594	99.83	85.94	63.64	99.83	85.20	71.62	99.83	90.37	80.56
250	744	99.87	90.69	68.97	99.85	90.22	67.88	99.87	88.80	69.15
300	894	99.89	86.35	72.68	99.89	86.33	83.83	99.87	87.90	78.17
350	1044	99.90	90.71	79.43	99.90	93.67	83.34	99.90	89.24	84.22
400	1194	99.92	90.42	79.64	99.91	90.85	84.51	99.92	83.32	85.85
450	1344	99.93	91.50	76.37	99.92	89.01	71.60	99.93	94.59	88.07
500	1494	99.93	94.16	76.44	99.93	89.77	77.90	99.93	90.98	74.62

		$\lambda = 0.5$								
		$p = 1$			$p = 2$			$p = 3$		
n	m	$q = 1$	$q = 2$	$q = 3$	$q = 1$	$q = 2$	$q = 3$	$q = 1$	$q = 2$	$q = 3$
50	144	99.31	87.78	59.93	98.75	88.26	70.49	98.06	82.64	78.54
100	294	99.66	80.95	70.20	99.59	84.90	77.69	99.59	86.33	72.86
150	444	99.77	93.94	82.43	99.75	79.30	74.77	99.68	91.35	73.54
200	594	99.83	87.71	77.66	99.83	89.09	73.08	99.83	88.23	72.41
250	744	99.87	93.41	84.73	99.87	89.87	74.53	99.85	83.43	80.34
300	894	99.89	87.73	79.75	99.89	93.29	75.29	99.87	86.51	80.16
350	1044	99.90	90.68	80.99	99.90	80.19	72.55	99.90	94.13	87.93
400	1194	99.91	92.50	75.44	99.92	92.90	80.13	99.91	96.41	84.83
450	1344	99.93	84.51	77.57	99.92	95.94	81.51	99.90	82.62	79.80
500	1494	99.93	78.97	77.95	99.93	85.48	78.84	99.93	90.69	81.99

		$\lambda = 0.75$								
		$p = 1$			$p = 2$			$p = 3$		
n	m	$q = 1$	$q = 2$	$q = 3$	$q = 1$	$q = 2$	$q = 3$	$q = 1$	$q = 2$	$q = 3$
50	144	99.31	86.94	66.60	99.03	84.10	75.14	99.03	96.53	67.01
100	294	99.59	84.18	75.75	99.66	81.50	76.60	99.56	87.93	64.83
150	444	99.77	83.90	72.09	99.75	75.95	72.25	99.73	95.63	74.19
200	594	99.83	84.90	61.67	99.81	78.59	75.27	99.80	86.16	74.33
250	744	99.87	85.22	73.35	99.87	88.76	77.42	99.84	90.73	74.41
300	894	99.89	76.87	69.59	99.89	88.59	76.14	99.87	84.19	78.90
350	1044	99.90	91.94	83.33	99.90	89.23	81.48	99.89	90.16	80.21
400	1194	99.92	83.28	87.15	99.91	86.67	78.22	99.92	91.26	76.73
450	1344	99.93	90.97	76.65	99.93	84.06	84.55	99.92	89.69	79.14
500	1494	99.93	82.58	79.30	99.93	88.35	87.03	99.93	87.42	75.55

		$\lambda = 1$								
		$p = 1$			$p = 2$			$p = 3$		
n	m	$q = 1$	$q = 2$	$q = 3$	$q = 1$	$q = 2$	$q = 3$	$q = 1$	$q = 2$	$q = 3$
50	144	99.31	77.50	67.64	72.57	68.89	48.54	89.86	56.11	25.35
100	294	99.66	86.02	50.54	85.75	58.74	43.20	92.14	58.06	38.16
150	444	99.73	88.24	64.62	85.50	68.13	50.34	85.65	59.82	31.96
200	594	99.81	87.93	72.86	88.69	63.65	49.63	86.23	62.69	40.61
250	744	99.85	91.80	69.84	90.51	70.26	50.99	96.92	68.23	37.69
300	894	99.85	92.89	62.35	90.86	64.63	43.80	90.58	61.25	44.22
350	1044	99.89	75.35	82.14	91.38	77.03	52.64	89.74	63.60	29.79
400	1194	99.92	83.42	74.36	95.39	70.63	48.56	91.58	64.73	48.05
450	1344	99.93	88.76	76.97	87.12	80.63	64.22	93.56	70.44	41.84
500	1494	99.93	85.54	73.21	97.51	79.90	52.30	81.27	49.63	38.25

Table VII.8: Average percentage of edges removed by Theorem VII.2 for planar networks with n = 50 up to 500 nodes.

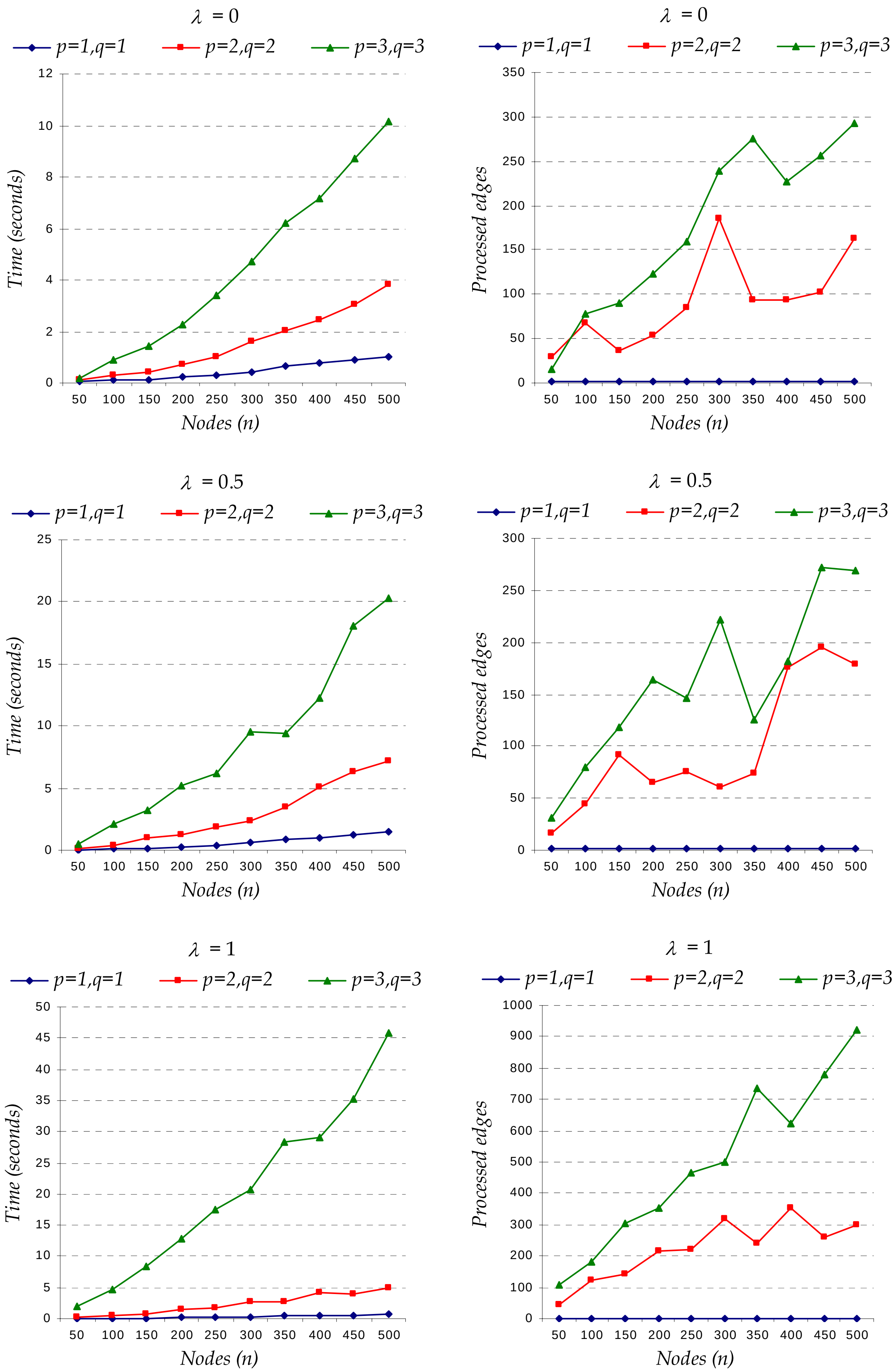


Figure VII.6: Average time results and average processed edges for networks with $n = 50$ to 500 nodes and λ equal to 0, 0.5 and 1.

Conclusions

"Things really make sense when they are over"
ANONYMOUS

Several models on desirable and undesirable single facility location on networks with multiple criteria have been analyzed and developed in this thesis. Likewise, we have also proposed some improvements on undesirable facility location models on single criterion networks.

Accordingly, regarding the location of desirable facilities on networks with n nodes and m edges, we have proposed an $O(mn \log n)$ algorithm to solve the biobjective λ-cent-dian problem. We proved that the set of efficient points to locate the λ-cent-dian could be infinite, as opposed to the uniobjective case where the λ-cent-dian is located on the set of nodes or on the set of local minima of the center function.

We have also studied the location of a facility on a network with multiple median-type objectives. In this case, the set of efficient points is not restricted to the nodes or to the shortest paths linking the median vertices of each objective, but rather to any place on the network. Being q the number of lengths per edge, we have proposed an $O(m^2q^3)$ algorithm to solve this problem. Besides, we have also presented a new procedure in $O(q)$ time that solves a two-variable linear programming problem to determine the set of efficient points.

Likewise, we have developed a polynomial algorithm in $O(m^2n^2k^3)$ time to solve the multicriteria network λ-cent-dian problem on networks with p weights per node and q lengths per edge, with $k = p \times q$. This model generalizes the one presented in Chapter II by using the multicriteria algorithm devised in Chapter III. Moreover, due to the convex combination through a parameter λ, this model allows obtaining the solution to both the multicriteria center problem and the multicriteria median problem.

Regarding undesirable facility location problems, we first addressed the undesirable 1-center (uncenter) location problem on networks. We showed that the upper bounds proposed in earlier papers can be tightened. By means of a more suitable problem formulation we have developed a new $O(mn)$ algorithm, which is more straightforward and computationally faster than the ones already reported in the literature.

Besides, we have analyzed the problem of locating an undesirable median (maxian) on a network, obtaining a new and better upper bound. We have presented a new algorithm in $O(mn)$ time to solve this problem. The new upper bound is dynamically updated within the algorithm, and thus, it accelerates the search of the optimal points. On the other hand, following the resolution of the maxian problem, we have also proposed a new $O(mn)$ algorithm to solve the network λ-anti-cent-dian problem, which improves the former method in $O(mn \log n)$ presented in the literature.

Finally, we have studied the uncenter and maxian problems on multicriteria networks, establishing new properties and rules to remove inefficient edges. We have also presented the multicriteria λ-anti-cent-dian model as a convex combination of the two latter problems through a parameter λ. We propose an effective rule to remove edges containing inefficient points, as well as a polynomial algorithm in $O(m^2 n^2 k^3)$ time, being k the number of criteria. Besides, this model can solve both the multicriteria uncenter problem and the multicriteria maxian problem. Moreover, when the network holds a single weight per node and a single length per edge, this algorithm can efficiently solve the single criterion uncenter, maxian and λ-anti-cent-dian problems. Lastly, this model might be slightly modified to generalize other models presented in the literature.

Appendix

function *UnCenter*(Network *N*, Distance Matrix *d*)

{ *// Current best value on network N.*

  $F_N := 0$

  // Solution set.

  $S := \varnothing$

  for all edges $e = (v_s, v_t) \in E$ **do**

    { *// Compute UB1.*

      $x_{UB1} := X(v_s, v_t)$

      $F_{UB1} := F_s^L(x_{UB1})$

      if $F_N > F_{UB1}$ **then** continue to next edge

      // Compute UB2.

      $F_g := \infty$, $F_h := \infty$

      for all nodes $v_i \in V$ **do**

        { **if** $v_i \neq v_s$ **and** $(F_i^L(0) < F_g$ **or** $(F_i^L(0) = F_g$ **and** $w_i < w_g))$ **then**

            { $F_g := F_i^L(0)$

              $v_g := v_i$

            }

          if $v_i \neq v_t$ **and** $(F_i^R(l_e) < F_h$ **or** $(F_i^R(l_e) = F_h$ **and** $w_i < w_h))$ **then**

            { $F_h := F_i^R(l_e)$

              $v_h := v_i$

            }

        }

      $x_{UB2} := X(v_g, v_h)$

      $F_{UB2} := F_g^L(x_{UB2})$

      // Try to tighten F_{UB2}.

      if $F_s^L(x_{UB2}) \le F_{UB2}$ **then**

        { $x_{UB2} := X(v_s, v_h)$

          $F_{UB2} := F_s^L(x_{UB2})$

          $v_g := v_s$

        }

      else if $F_s^L(x_{UB2}) \le F_{UB2}$ **then**

        { $x_{UB2} := X(v_g, v_t)$

          $F_{UB2} := F_t^R(x_{UB2})$

          $v_h := v_t$

        }

// F_{UB2} must be at least as good as F_{UB1}
if $F_{UB2} \geq F_{UB1}$ **then**
{ $(x_{UB2}, F_{UB2}) := (x_{UB1}, F_{UB1})$
$v_g := v_s$
$v_h := v_t$
}
if $F_N > F_{UB2}$ **then** continue to next edge
// Compute UB3.
$F_p := \infty$, $F_q := \infty$
for all nodes $v_i \in V$ **do**
{ **if** $v_i \neq v_s$ **and** $(F_i^L(l_e) < F_p$ **or** $(F_i^L(l_e) = F_p$ **and** $w_i < w_p))$ **then**
{ $F_p := F_i^L(l_e)$
$v_p := v_i$
}

if $v_i \neq v_t$ **and** $(F_i^R(0) < F_q$ **or** $(F_i^R(0) = F_q$ **and** $w_i < w_q))$ **then**
{ $F_q := F_i^R(0)$
$v_q := v_i$
}
}
$x_{UB3} := X(v_p, v_q)$
$F_{UB3} := F_p^L(x_{UB3})$
// Try to tighten F_{UB3}.
if $F_s^L(x_{UB3}) \leq F_{UB3}$ **then**
{ $x_{UB3} := X(v_s, v_q)$
$F_{UB3} := F_s^L(x_{UB3})$
$v_p := v_s$
}
else if $F_s^L(x_{UB3}) \leq F_{UB3}$ **then**
{ $x_{UB3} := X(v_p, v_t)$
$F_{UB3} := F_t^R(x_{UB3})$
$v_q := v_t$
}

// F_{UB3} must be at least as good as F_{UB1}
if $F_{UB2} \geq F_{UB1}$ **then**
{ $(x_{UB3}, F_{UB3}) := (x_{UB1}, F_{UB1})$
$v_p := v_s$
$v_q := v_t$
}
if $F_N > F_{UB3}$ **then** continue to next edge

```
// Set (x_e, F_e) to the best value found.
if F_UB2 ≤ F_UB3 then
   {  (x_e, F_e) := (x_UB2, F_UB2)
      v_a := v_g
      v_b := v_h
   }
else
   {  (x_e, F_e) := (x_UB3, F_UB3)
      v_a := v_p
      v_b := v_q
   }
// Create set L and R.
if v_a ≠ v_s then   L := L ∪ {v_a}
if v_b ≠ v_t then   R := R ∪ {v_b}
for all nodes v_i ∈ V do
   {  d := d(v_s, v_i) − d(v_t, v_i)
      if d < l_e and F_i^L(x_UB2) < F_UB2 then    L := L ∪ {v_i}
      if −d < l_e and F_i^R(x_UB2) < F_UB2 then   R := R ∪ {v_i}
   }
// Continue till the new value F_e cannot improve the current F_N,
// or until one of the node sets becomes empty.
while F_e ≥ F_N and ( L ≠ ∅ or R ≠ ∅ ) do
   {  // Pair all nodes in L against R, using a max{|L|,|R|} matching
      for all the pair of nodes (v_i ∈ L, v_j ∈ R) in the matching do
         {  x := X(v_i, v_j)
            if F_s^L(x) ≤ F_i^L(x) then  // F_i^L is over F_j^R
               {  L := L − {v_i}
                  x := X(v_s, v_j)
                  v_i := v_s
               }
            else if F_s^L(x) ≤ F_i^L(x) then    // F_j^R is over F_i^L
               {  R := R − {v_j}
                  x := X(v_i, v_t)
                  v_j := v_t
               }
            // Update (x_e, F_e)
            if F_i^L(x) < F_e then
               {  x_e := x
                  F_e := F_i^L(x_e)
                  v_a := v_i
                  v_b := v_j
               }
         }
```

```
                // Project the value x_e on the lower envelope.
                // Find the lowest left line.
                F_a := ∞
                for all nodes v_i ∈ L do
                    if F_i^L(x_e) < F_a or (F_i^L(x_e) = F_a and w_i < w_a) then
                        {   F_a := F_i^L(x_e)
                            v_a := v_i
                        }
                // Find the lowest right line.
                F_b := ∞
                for all nodes v_i ∈ R do
                    if F_i^R(x_e) < F_b or (F_i^R(x_e) = F_b and w_i < w_b) then
                        {   F_b := F_i^R(x_e)
                            v_b := v_i
                        }
                x_e := X(v_a, v_b)
                F_e := F_a^L(x_e)
                // Delete lines above the new value F_e
                for all nodes v_i ∈ L do
                    if F_i^L(x_e) ≥ F_e then L := L − {v_i}
                for all nodes v_i ∈ R do
                    if F_i^R(x_e) ≥ F_e then R := R − {v_i}
            }
        if F_e ≥ F_N then
            {   if F_e > F_N then
                    {   S := ∅
                        F_N := F_e
                    }

                S := S ∪ {(x_e, e)}
            }
    }

    return (F_N, S)
}
```

Bibliography

[1] Arnott, R. (1986), *Location theory*, Harwood Academic Publishers, London.

[2] Averbakh, I. and Berman, O. (1999), "Algorithms for path medi-centers of a tree", *Computers and Operations Research* 26(14), 1395-1409.

[3] Badri, M.A., Mortagy, A.K. and Alsayed, A. (1998), "A multi-objective model for locating fire stations", *European Journal of Operational Research* 110(2), 243-260.

[4] Barda, O.H., Dupuis, J. and Lencioni, P. (1990), "Multicriteria location of thermal power plants", *European Journal of Operational Research* 45(2-3), 332-346.

[5] Batta, R. and Palekar, U.S. (1988), "Mixed planar/network facility location problems", *Computers and Operations Research* 15(1), 61-67.

[6] Bentley, J.L. and Ottmann, T.A. (1979), "Algorithms for reporting and counting geometric intersections", *IEEE Transactions on Computers* 28(9), 643-647.

[7] Berman, O. and Drezner, Z. (2000), "A note on the location of an obnoxious facility on a network", *European Journal of Operational Research* 120(1), 215-217.

[8] Berman, O. and Yang, E.K. (1991), "Medi-centre location problems", *Journal of the Operational Research Society* 42(4), 313-322.

[9] Berman, O., Drezner, Z. and Wesolowsky, G.O. (1996), "Minimum covering criterion for obnoxious facility location on a network", *Networks* 28(1), 1-5.

[10] Bitran, G.R. and Rivera, J.M. (1982), "A combined approach to solve binary multicriteria problems", *Naval Research Logistics* 29(2), 181-201.

[11] Boffey, B. and Karkazis, J. (eds.) (1991), "Locational analysis", *RAIRO-Recherche Operationnelle* 25(1).

[12] Brandeau, M.L. and Chiu, S.S. (1989), "An overview of representative problems in location research", *Management Science* 35(6), 645-674.

[13] Buhl, H.U. (1988), "Axiomatic considerations in multi-objective location theory", *European Journal of Operational Research* 37(3), 363-367.

[14] Burkard, R.E. and Dollani, H. (2003), "Center problems with pos/neg weights on tree", *European Journal of Operational Research* 145(3), 483-495.

[15] Burkard, R.E., Çela, E. and Woeginger, G.J. (1995), "A minimax assignment problem in treelike communication networks", *European Journal of Operational Research* 87(3), 670-684.

[16] Burkard, R.E., Dollani, H., Lin, Y. and Rote, G. (2001), "The obnoxious center problem on a tree", *SIAM Journal on Discrete Mathematics* 14(4), 498-509.

[17] Cappanera, P. (1999), "A survey on obnoxious facility location problems", *Technical Report* 11, Dipartamento di Informatica, Università di Pisa.

[18] Carrizosa, E. (1992), *Problemas de localización multiobjetivo*, Ph.D. Thesis, Universidad de Sevilla.

[19] Carrizosa, E. and Conde, E. (2002), "A fractional model for locating semi-desirable facilities on networks", *European Journal of Operational Research* 136(1), 67-80.

[20] Carrizosa, E. and Plastria, F. (1999), "Location of semi-obnoxious facilities", *Studies in Locational Analysis* 12, 1-27.

[21] Carrizosa, E., Conde, E., Fernández, F.R. and Puerto, J. (1994), "An axiomatic approach to the cent-dian criterion", *Location Science* 2(3), 165-171.

[22] Carrizosa, E., Conde, E., Muñoz, M. and Puerto, J. (1995), "The generalized weber problem with expected distances", *RAIRO* 29, 35-57.

[23] Chhajed, D., Francis, R.L. and Lowe, T.J. (1993), "Contributions of Operations Research to location analysis", *Location Science* 1(4), 263-287.

[24] Chiu, S.S. (1987), "The minisum location problem on an undirected network with continous link demands", *Computers and Operations Research* 14(5), 369-383.

[25] Church, R.L. and Garfinkel, R.S. (1978), "Locating an obnoxious facility on a network", *Transportation Science* 12(2), 107-118.

[26] Cohon, J.L., ReVelle, C.S., Current, J.R., Eagles, T., Eberhart, R. and Church, R.L. (1980), "Application of a multiobjective facility location model to power plant siting in a six-state region of the U.S.", *Computers and Operations Research* 7(1-2), 107-123.

[27] Colebrook, M., Gutiérrez, J. and Sicilia, J. (2002), "A new bound and an O(*mn*) algorithm for the undesirable 1-median problem (maxian) on networks", *Working Paper* 02(1), Departamento de Estadística, I.O. y Computación, Universidad de La Laguna (accepted for publication in *Computers and Operations Research*).

[28] Colebrook, M., Gutiérrez, J., Alonso, S. and Sicilia, J. (2002), "A new algorithm for the undesirable 1-center problem on networks", *Journal of the Operational Research Society* 53(12), 1357-1366.

[29] Colebrook, M., Ramos, R.M., Ramos, M.T. and Sicilia, J. (2000), "Efficient points in the biobjective cent-dian problem", *Studies in Locational Analysis* 15, 1-15.

[30] Cuninghame-Green, R.A. (1984), "The absolute center of a graph", *Discrete Applied Mathematics* 7(3), 275-283.

[31] Current, J. and Schilling, D. (eds.) (1990), "Special issue on location analysis and modeling", *Geographical Analysis* 22(1).

[32] Current, J. and Schilling, D. (eds.) (1991), "Special issue on location analysis", *INFOR* 29(2), 65-67.

[33] Current, J., Daskin, M. and Schilling, D. (2002), "Discrete network location models", in *Facility location theory: Applications and methods*, Drezner, Z. and Hamacher, H.W. (eds.), Springer Verlag, 85-118.

[34] Current, J., Min, H. and Schilling, D. (1990), "Multiobjective analysis of facility location decisions", *European Journal of Operational Research* 49(3), 295-307.

[35] Current, J.R. (ed.) (1988), "Special issue on location analysis", *Environment and Planning B* 15, 127-236.

[36] Current, J.R. and Ratick, S. (eds.) (1992), "Special issue: Facility location modeling", *Papers in Reginal Science* 71(3).

[37] Current, J.R. and Storbeck, J.E. (1994), "A multiobjective approach to design franchise outlet network", *Journal of the Operational Research Society* 45(1), 71-81.

[38] Daskin, M.S. (1995), *Network and discrete location: Models, algorithms and applications*, Wiley Interscience, New York.

[39] Domschke, W. and Drexl, A. (1985), *Location and layout planning: An international bibliography*, Springer, Berlin.

[40] Drezner, Z. (2002), "Location", in *Hanbook of Applied Optimization*, Pardalos, P. and Resende, M.G.C. (eds.), Oxford University Press, New York, 632-639.

[41] Drezner, Z. (ed.) (1992), "Locational decisions", *Annals of Operations Research* 40.

[42] Drezner, Z. (ed.) (1995), *Facility location: A survey of applications and methods*, Springer, New York.

[43] Drezner, Z. and Hamacher, H.W (eds.) (2002), *Facility location: applications and theory*, Springer.

[44] Drezner, Z. and Wesolowsky, G.O. (1995), "Obnoxious facility location in the interior of a planar network", *Journal of Regional Science* 35(4), 675-688.

[45] Drezner, Z., Thisse, J.-F. and Wesolowsky, G.O. (1986), "The minimax-min location problem", *Journal of Regional Science* 26(1), 87-101.

[46] Dyer, M.E. (1984), "Linear time algorithms for two- and three-variable linear programs", *SIAM Journal on Computing* 13(1), 31-45.

[47] Ehrgott, M. and Gandibleux, X. (2000), "A survey and annotated bibliography of multiobjective combinatorial optimization", *OR Spektrum* 22(4), 425-460.

[48] Eiselt, H.A. (1992), "Location modeling in practice", *American Journal of Mathematical and Management Sciences* 12(1), 3-18.

[49] Eiselt, H.A. and Laporte, G. (1995), "Objectives in location", in *Facility locations: A survey of applications and methods*, Drezner, Z. (ed.), Springer Verlag, 151-180.

[50] Eiselt, H.A., Laporte, G. and Thisse, J-F. (1993), "Competitive location models: A framework and bibliography", *Transportation Science* 27(1), 44-54.

[51] Erkut, E. and Neuman, S. (1989), "Analytical models for locating undesirable facilities", *European Journal of Operational Research* 40(3), 275-291.

[52] Erkut, E. and Neuman, S. (1992), "A multiobjective model for locating undesirable facilities", *Annals of Operations Research* 40, 209-227.

[53] Erkut, E. and Verter, V. (1995), "Hazardous materials logistics", in *Facility locations: A survey of applications and methods*, Drezner, Z. (ed.), Springer Verlag, 467-506.

[54] Fortenberry, J.C., Mitra, A. and Willis, R.D.M. (1989), "A multicriteria approach to optimal emergency vehicle location analysis", *Computers and Industrial Engineering* 16(2), 339-347.

[55] Francis, R.L., McGinnis, L.F. and White, J.A. (1983), "Locational analysis", *European Journal of Operational Research* 12, 220-252.

[56] Francis, R.L., McGinnis, L.F. Jr. and White, J.A. (1992), *Facility layout and location: An analytical approach*, Prentice Hall, Englewood Cliffs.

[57] Fredman, M. and Tarjan, R. (1987), "Fibonacci heaps and their uses in improved network optimization algorithms", *Journal of the ACM* 34(3), 596-615.

[58] Friedrich, F. (1929), *Alfred Weber's theory of the location of industries*, Chicago University Press, Chicago, Illinois.

[59] Gabszewicz, J. and Thisse, J.-F. (1992), "Location", in *Handbook of Game Theory with Economic Applications*, Aumann, R. and Hart, S. (eds.), Elsevier Science Publisher, Amsterdam, 281-304.

[60] Garey, M.R. and Johnson, D.S. (1979), *Computers and intractability: A guide to the theory of $\mathcal{NP}$-Completeness*, Freeman, New York.

[61] Giannikos, I. (1998), "A multiobjective programming model for locating treatment sites and routing hazardous wastes", *European Journal of Operational Research* 104(2), 333-342.

[62] Goldman, A.J. (1969), "Optimal locations for centers in a network", *Transportation Science* 3, 352-360.

[63] Goldman, A.J. (1971), "Optimal center location in simple networks", *Transportation Science* 5, 212-221.

[64] Goldman, A.J. (1972), "Minimax location of a facility in a network", *Transportation Science* 6, 407-418.

[65] Goldman, A.J. and Mayers, P.R. (1965), "A domination theorem for optimal location", *Operations Research* 13B, 147-147.

[66] Gosh, J.K. (1985), "Facility location with multi-objectives", *Industrial Engineering Journal (India)* 14, 21-24.

[67] Hakimi, S.L. (1964), "Optimum locations of switching centers and the absolute centers and medians of a graph", *Operations Research* 12(3), 450-459.

[68] Hakimi, S.L. (1965), "Optimum distribution of switching centers in a communication network and some related graph theoretic problems", *Operations Research* 13(3), 462-475.

[69] Hakimi, S.L. and Maheshwari, S.N. (1972), "Optimum locations of centers in networks", *Operations Research* 20(5), 967-973.

[70] Hakimi, S.L., Schmeichel, E.F. and Pierce, S.G. (1978), "On p-centers in networks", *Transportation Science* 12(1), 1-15.

[71] Hale, T. (1998), "Location science references", *www.ent.ohiou.edu/~thale/thlocation.html*.

[72] Hale, T. (1999), "Facility location: A synopsis and taxonomy", in *Industrial Engineering Applications and Practice: User's Encyclopedia*, Mital, A. (ed.), University of Cincinnati Press, Cincinnati, Ohio.

[73] Halfin, S. (1974), "On finding the absolute and vertex centers of a tree with distances", *Transportation Science* 8, 75-77.

[74] Halpern, J. (1976), "The location of a center-median convex combination on an undirected tree", *Journal of Regional Science* 16, 237-245.

[75] Halpern, J. (1978), "Finding minimal center-median convex combination (cent-dian) of a graph", *Management Science* 24(5), 535-544.

[76] Halpern, J. (1980), "Duality in the cent-dian of a graph", *Operations Research* 28(3), 722-735.

[77] Halpern, J. and Maimon, O. (1983), "Accord and conflict among several objectives in locational decisions on tree networks", in *Locational Analysis of Public Facilities*, Thisse, J.F. and Zoller, H.G. (eds.), North Holland, Amsterdam, 391-304.

[78] Hamacher, H.W. and Nickel, S. (1998), "Classification of location models", *Location Science* 6(1-4), 229-242.

[79] Hamacher, H.W., Labbé, M. and Nickel, S. (1999), "Multicriteria network location problems with sum objectives", *Networks* 33(2), 79-92.

[80] Hamacher, H.W., Labbé, M., Nickel, S. and Skriver, A.J.V. (2002), "Multicriteria semi-obnoxious network location problems (MSNLP) with sum and center objectives", *Annals of Operations Research* 110(1-4), 33-53.

[81] Handler, G.Y. (1973), "Minimax location of a facility in an undirected tree graph", *Transportation Science* 7, 287-293.

[82] Handler, G.Y. (1974), "Minimax network location: Theory and algorithms", *Technical Report* 107, Operations Research Center, MIT.

[83] Handler, G.Y. (1976), "Medi-centers of a tree", *Working Paper* 278, The Recanati Graduate School of Business Administration, Tel Aviv University, Israel.

[84] Handler, G.Y. (1985), "Medi-centers of a tree", *Transportation Science* 19(3), 246-260.

[85] Handler, G.Y. and Mirchandani, P.B. (1979), *Location in networks: Theory and algorithms*, MIT Press, Cambridge, Massachusetts.

[86] Hansen, P., Labbé, M. and Nicolas, B. (1991), "The continuous center set of a network", *Discrete Applied Mathematics* 30(2-3), 181-195.

[87] Hansen, P., Labbé, M. and Thisse, J.-F. (1991), "From the median to the generalized center", *Recherche Operationnelle/Operations Research* 25(1), 73-86.

[88] Hansen, P., Labbé, M., Peeters, D. and Thisse, J.-F. (1987a), "Facility location analysis", *Fundamental Pure Applied Economics* 22, 1-70.

[89] Hansen, P., Labbé, M., Peeters, D. and Thisse, J.-F. (1987b), "Single facility location on networks", *Annals of Discrete Mathematics* 31, 113-146.

[90] Hansen, P., Labbé, M., Peeters, D., Thisse, J.-F. and Henderson, J.V. (eds.) (1987), *Systems of cities and facility location*, Harwood Academic.

[91] Hansen, P., Peeters, D. and Thisse, J.-F. (1983), "Public facility location models: A selective survey", in *Locational Analysis of Public Facilities*, Thisse, J.F. and Zoller, H.G. (eds.), North Holland, New York, 223-262.

[92] Hansen, P., Thisse, J.-F. and Wendell, R.E. (1986a), "Efficient points on a network", *Networks* 16(4), 357-368.

[93] Hansen, P., Thisse, J.-F. and Wendell, R.E. (1986b), "Equivalence of solutions to network location problems", *Mathematics of Operations Research* 11(4), 672-678.

[94] Hershberger, J. (1989), "Finding the upper envelope of n line segments in O(n log n) time", *Information Processing Letters* 33(4), 169-174.

[95] Hoare, C.A.R. (1961), "Algorithm 63: Partition & Algorithm 65: Find", *Communications of the ACM* 4(7), 321-322.

[96] Hokkanen, J. and Salminen, P. (1997), "Locating a waste treatment facility by multicriteria analysis", *Journal of Multi-Criteria Decision Analysis* 6(3), 175-184.

[97] Hooker, J.N., Garfinkel, R.S. and Chen, S.K. (1991), "Finite dominating sets for network location problems", *Operations Research* 39(1), 100-118.

[98] Hultz, J.W., Klingman, D.D., Ross, G.T. and Soland, R.M. (1981), "An interactive computer system for multicriteria facility location", *Computers and Operations Research* 8(4), 249-261.

[99] Hurter, A.P. and Martinich, J.S. (1989), *Facility location and the theory of production*, Kluwer Academic Publishers, London.

[100] Jarvinen, P., Sinervo, H. and Rajala, J. (1972), "A branch-and-bound algorithm for seeking the p-median", *Operations Research* 20(1), 173-178.

[101] Jordan, C. (1869), "Sur les assamblages de lignes", *Zeitschrift für die Reine und Angewandte Mathematik* 70, 185-190.

[102] Kalcsics, J., Nickel, S., Puerto, J. and Tamir, A. (2002), "Algorithmic results for ordered median problems", *Operations Research Letters* 30(3), 149-158.

[103] Kariv, O. and Hakimi, S.L. (1979a), "An algorithmic approach to network location problems. I: The p-centers", *SIAM Journal on Applied Mathematics* 37(3), 513-538.

[104] Kariv, O. and Hakimi, S.L. (1979b), "An algorithmic approach to network location problems. II: The p-medians", *SIAM Journal on Applied Mathematics* 37(3), 539-560.

[105] Kincaid, R.K. and Berger, R.T. (1994), "The maxminsum problem on trees", *Location Science* 2(1), 1-9.

[106] Krarup, J. and Pruzan, P.M. (1990), "Ingredients of Locational Analysis", in *Discrete Location Theory*, Mirchandani, P.B. and Francis, R.L. (eds.), John Wiley & Sons, 1-54.

[107] Krumke, S.O., Noltemeier, H., Ravi, S.S. and Marathe, M.V. (1996), "Bicriteria compact location problems", *Studies in Locational Analysis* 10, 37-51.

[108] Kuby, M.J. (1987), "Programming models for facility dispersion: The p-Dispersion and maxisum dispersion problems", *Geographical Analysis* 19(4), 315-329.

[109] Labbé, M. (1990), "Location of an obnoxious facility on a network - A voting approach", *Networks* 20(2), 197-207.

[110] Labbé, M. (1998), "Facility location: models, methods and applications", in *Operations Research and Decision Aid in Traffic and Transportation Management*, NATO ASI Series F,

Labbé, M., Laporte, G., Tanczos, K. and Toint, Ph. (eds.), Springer Verlag, Heidelberg 166, 264-285.

[111] Labbé, M. and Louveaux, F.V. (1997), "Location problems", in *Annotated bibliography in Combinatorial Optimization*, Dell'Amico, M., Maffioli, F. and Martello, S. (eds.), 261-281.

[112] Labbé, M., Peeters, D. and Thisse, J.-F. (1995), "Location on networks", in *Handbooks in Operations Research and Management Science*, Ball, M.O., Magnanti, T.L., Monma, C.L. and Nemhauser, G.L. (eds.), North Holland, Amsterdam 8, 551-624.

[113] List, G.F. and Mirchandani, P.B. (1991), "An integrated network/planar multiobjective model for routing and siting for hazardous materials and wastes", *Transportation Science* 25(2), 146-156.

[114] López-de-los-Mozos, M.C. and Mesa, J.A. (2001), "The maximum absolute deviation measure in location problems on networks", *European Journal of Operational Research* 135(1), 184-194.

[115] Louveaux, F.V., Labbé, M. and Thisse, J.-F. (eds.) (1989), "Facility location analysis: Theory and applications", *Annals of Operations Research* 18.

[116] Love, R.F, Morris, J.G. and Wesolowsky, G.O. (1988), *Facilities location: Models & methods*, North Holland, New York.

[117] Lowe, T.J. (1978), "Efficient solutions in multiobjective tree network location problems", *Transportation Science* 12(4), 298-316.

[118] Mahmoud, M.R., Fahmy, H. and Labadie J.W. (2002), "Multicriteria siting and sizing of desalination facilities with geographic information system", *Journal of Water Resources Planning and Management-Asce* 128(2), 113-120.

[119] Malczewski, J. and Ogryczak, W. (1995), "The multiple criteria location problem: 1. A generalized network model and the set of efficient solutions", *Environment and Planning A* 27(12), 1931-1960.

[120] Malczewski, J. and Ogryczak, W. (1996), "The multiple criteria location problem: 2. Preference-based techniques and interactive decision support", *Environment and Planning A* 28(1), 69-98.

[121] Manber, U. (1989), *Introduction to algorithms*, Addison-Wesley.

[122] Marsh, M. and Schilling, D.A. (1994), "Equity measurement in facility location analysis: Review and framework", *European Journal of Operational Research* 74(1), 1-17.

[123] Megiddo, N. (1982), "Linear-time algorithms for linear programming in R^3 and related problems", *SIAM Journal on Computing* 12(4), 759-776.

[124] Melachrinoudis, E. and Zhang, F.G. (1999), "An O(*mn*) algorithm for the 1-maxmin problem on a network", *Computers and Operations Research* 26(9), 849-869.

[125] Melachrinoudis, E., Min, H. and Wu, X. (1995), "A multiobjective model for the dynamic location of landfills", *Location Science* 3(3), 143-166.

[126] Melhorn, K. and Näher, S. (1999), *LEDA: A platform for combinatorial and geometric computing*, Cambridge University Press.

[127] Miller, T.C., Friesz, T.C. and Tobin, R. (1996), *Equilibrium facility location on networks*, Springer, Berlin.

[128] Min, H. (1987), "A multiobjective retail service location model for fast-food restaurants", *Omega* 15(5), 429-441.

[129] Min, H. (1988), "The dynamic expansion and relocation of capacitated public facilities: A multi-objective approach", *Computers and Operations Research* 15(3), 243-252.

[130] Minieka, E. (1977), "The centers and medians of a graph", *Operations Research* 25(4), 641-650.

[131] Minieka, E. (1980), "Conditional centers and medians of a graph", *Networks* 10(3), 265-272.

[132] Minieka, E. (1981), "A polynomial time algorithm for finding the absolute center of a network", *Networks* 11(4), 351-355.

[133] Minieka, E. (1983), "Anticenters and antimedians of a network", *Networks* 13(3), 359-364.

[134] Mirchandani, P.B. (1990), "The p-median problem and generalizations", in *Discrete Location Theory*, Mirchandani, P.B. and Francis, R.L. (eds.), John Wiley & Sons, 55-117.

[135] Mirchandani, P.B. and Francis, R.L. (eds.) (1990), *Discrete location theory*, Wiley Interscience, New York.

[136] Mladineo, N., Margeta, J., Barns, J.P. and Mareschal, B. (1987), "Multicriteria ranking of alternative locations for small scale hydro plants", *European Journal of Operational Research* 31, 215-222.

[137] Moon, I.D. (1989), "Maximin center of pendant vertices in a tree network", *Transportation Science* 23(3), 213-216.

[138] Moon, I.D. and Chaudhry, S.S. (1984), "An analysis of network location problems with distance constraints", *Management Science* 30(3), 290-307.

[139] Moreno-Pérez, J.A. and Rodríguez-Martín, I. (1999), "Anti-cent-dian on networks", *Studies in Locational Analysis* 12, 29-39.

[140] Mosler, K.C. (1987), *Continuous location of transportation networks*, Springer Verlag, Berlin.

[141] Murray, A.T., Church, R.L., Gerrard, R.A. and Tsui, W.S. (1998), "Impact models for siting undesirable facilities", *Papers in Regional Science* 77(1), 19-36.

[142] Narula, S.C., Ogbu, U.I. and Samuelsson, H.M. (1977), "An algorithm for the p-median problem", *Operations Research* 25(4), 709-713.

[143] Nickel, S. and Puerto, J. (1999), "A unified approach to network location problems", *Networks* 34(4), 283-290.

[144] Nijkamp, P. and Spronk, J. (1981), "Interactive multidimensional programming models for locational decisions", *European Journal of Operational Research* 6, 220-223.

[145] Ogryczak, W. (1997a), "On cent-dians of general networks", *Location Science* 5(1), 15-28.

[146] Ogryczak, W. (1997b), "On the lexicographic minimax approach to location problems", *European Journal of Operational Research* 100(3), 566-585.

[147] Ogryczak, W. (1999), "On the distribution approach to location problems", *Computers and Industrial Engineering* 37(3), 595-612.

[148] Osleeb, J.P. and Ratick, S.J. (eds.) (1986), "Locational decisions: Methodology and applications", *Annals of Operations Research* 6.

[149] Oudjit, A. (1981), *Median locations on deterministic and probabilistic multidimensional networks*, Ph.D. Thesis, Rensselaer Polytechnic Institute, Troy, New York.

[150] Pareto, V. (1896), *Cours d'economie politique*, F. Rouge, Lausanne 1.

[151] Plastria, F. (1996), "Optimal location of undesirable facilities: A selective overview", *Belgian Journal of Operations Research, Statistics and Computer Science* 36(2-3), 109-127.

[152] Preparata, F.P. and Shamos, M.I. (1985), *Computational geometry: An introduction*, Springer-Verlag, New York.

[153] Puerto, J. (ed.) (1996), *Lecturas en teoría de localización*, Universidad de Sevilla, Secretariado de Publicaciones.

[154] Puerto, J. and Fernández, F.R. (1994), "Multicriteria decisions in location", *Studies in Locational Analysis* 7, 185-199.

[155] Rahman, M. and Kuby, M. (1995), "A multiobjective model for locating solid waste transfer facilities using an empirical opposition function", *INFOR* 33(1), 34-49.

[156] Ramos, R.M., Ramos, M.T., Colebrook, M. and Sicilia, J. (1999), "Locating a facility on a network with multiple median-type objectives", *Annals of Operations Research* 86, 221-235.

[157] Ramos, R.M., Sicilia, J. and Ramos, T. (1992), "Non-dominated location points for double-weighted networks", in *Proceedings of the VI Meeting of the Working Group on Locational Analysis*, 167-174.

[158] Ramos, R.M., Sicilia, J. and Ramos, T. (1997), "The biobjective absolute center problem", *TOP* 5(2), 187-199.

[159] Ratick, S.J. and White, A.L. (1988), "A risk-sharing model for locating noxious facilities", *Environment and Planning B* 15(2), 165-179.

[160] ReVelle, C.S., Cohon, J.L. and Shobrys, D. (1981a), "Multiple objective facility location", *Sistemi Urbani* 3, 319-343.

[161] ReVelle, C.S., Cohon, J.L. and Shobrys, D. (1981b), "Multiple objectives in facility location: A review", in *Lecture Notes in Economic and Mathematical Systems, Organisations: Multiple agents with multiple criteria*, Springer Verlag 190, 321-337.

[162] Ross, G.T. and Soland, R.M. (1980), "A multicriteria approach to the location of public facilities", *European Journal of Operational Research* 4(5), 307-321.

[163] Saameño, J.J. (1992), *Localización multicriterio de centros peligrosos*, Ph.D. Thesis, Universidad de Sevilla.

[164] Salhi, S., Welch, S.B. and Cunninghame-Green, R.A. (2000), "An enhancement of an analytical approach: The case of the weighted maximin network location problem", *Mathematical Algorithms* 1(4), 315-329.

[165] Salvaneschi, L. (1996), *Location, location, location: How to select the best site for your business*, PSI Research - Oasis Press.

[166] Schilling, D.A. (1980), "Dynamic location modelling for public sector facilities: A multi-criteria approach", *Decision Sciences* 11, 714-725.

[167] Sforza, A. (1990), "An algorithm for finding the absolute center of a network", *European Journal of Operational Research* 48(3), 376-390.

[168] Singer, S. (1968), "Multi-centers and multi-medians of a graph with an application to optimal warehouse location", *Operations Research* 16B, 87-88.

[169] Skriver, A.J.V. (2001), *Multicriteria analysis on network and location problems*, Ph.D. Thesis, University of Aarhus.

[170] Skriver, A.J.V. and Andersen, K.A. (2001), "The bicriterion semi-obnoxious location (BSL) problem solved by an ε-approximation", *Technical Report*, Department of Operations Research, University of Aarhus.

[171] Slater, P. (1975), "Maximin facility location", *Journal of Research National Bureau of Standards* 79B(3-4), 107-115.

[172] Steuer, R.E. (1986), *Multiple criteria optimization: theory, computation and application*, Wiley, New York.

[173] Stowers, C.L. and Palekar, U.S. (1993), "Location models with routing considerations for a single obnoxious facility", *Transportation Science* 27(4), 350-362.

[174] Tamir, A. (1987), "Totally balanced and totally unimodular matrices defined by center location problems", *Discrete Applied Mathematics* 16(3), 245-263.

[175] Tamir, A. (1988), "Improved complexity bounds for center location problems on networks by using dynamic data structures", *SIAM Journal on Discrete Mathematics* 1(3), 377-396.

[176] Tamir, A. (1991), "Obnoxious facility location on graphs", *SIAM Journal on Discrete Mathematics* 4(4), 550-567.

[177] Tamir, A. (1992), "On the complexity of some classes of location problems", *Transportation Science* 26(4), 352-354.

[178] Tamir, A. (2001), "Comment on E. Melachrinoudis and F.G.-S. Zhang, *An O(mn) algorithm for the 1-maximin problenm on a network*, Computers & Operations Research 26 (1999) 849-869", *Computers and Operations Research* 28(2), 189.

[179] Tansel, B.C., Francis, R.L. and Lowe, T.J. (1980), "Binding inequalities for tree networks location problems with distance constraints", *Transportation Science* 14(2), 107-124.

[180] Tansel, B.C., Francis, R.L. and Lowe, T.J. (1982), "A biobjective multifacility minimax location problem on a tree network", *Transportation Science* 16(4), 407-429.

[181] Tansel, B.C., Francis, R.L. and Lowe, T.J. (1983a), "Location on networks: A survey. Part I: The p-center an p-median problems", *Management Science* 29(4), 482-497.

[182] Tansel, B.C., Francis, R.L. and Lowe, T.J. (1983b), "Location on networks: A survey. Part II: Exploiting tree network structure", *Management Science* 29(4), 498-511.

[183] Thisse, J.-F. and Zoller, H.G. (eds.) (1983), *Locational analysis of public facilities*, North Holland, Amsterdam.

[184] Ting, S.S. (1984), “A linear-time algorithm for maxisum facility location on tree networks”, *Transportation Science* 18(1), 76-84.

[185] Verter, V. and Erkut, E. (1995), “Hazardous materials logistics: An annotated bibliography”, in *Operations Research and Environmental Management*, Haurie, A. and Carraro, C. (eds.), Kluwer Publishing Company, 221-267.

[186] Warszawski, A. (1973), “Multi-dimensional location problems”, *Operational Research Quaterly* 24(2), 165-179.

[187] Weber, A. (1909), *Über den Standort der Industrien*, Verlag J.C.B. Mohr, Tübingen, Germany.

[188] Wendell, R.E. and Lee, D.N. (1977), “Efficiency in multiple objective optimization problems”, *Mathematical Programming* 12(3), 406-414.

[189] Wesolowsky, G.O. (1993), “The Weber problem: History and perspectives”, *Location Science* 1, 5-23.

[190] Zemel, E. (1984), “An O(n) algorithm for the linear multiple choice knapsack and related problems”, *Information Processing Letters* 18(3), 123-128.

[191] Zhang, F.G. (1996), *Location on networks with multiple criteria*, Ph.D. Thesis, Northeastern University, Boston, Massachusetts.

[192] Zhang, F.G. and Melachrinoudis, E. (2001), “The maximin-maxisum network location problem”, *Computational Optimization and Applications* 19(2), 209-234.