



**UNIVERSIDAD NACIONAL AUTÓNOMA
DE MÉXICO**

FACULTAD DE CIENCIAS

**Solución al Problema de Localización P-Mediana Usando
Descomposición de Benders**

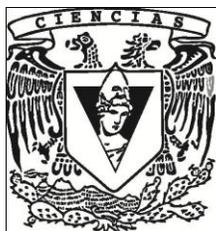
T E S I S

QUE PARA OBTENER EL TÍTULO DE:

M A T E M Á T I C O

P R E S E N T A:

Oswaldo Emmanuel Palma Cabrera



**DIRECTORA DE TESIS:
Dra. Zaida Estefanía Alarcón Bernal
Ciudad Universitaria, Cd. Mx., 2020**



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

1. Datos del alumno

Palma

Cabrera

Oswaldo Emmanuel

5555263973

Universidad Nacional Autónoma de

México

Facultad de Ciencias

Matemáticas

310221852

2. Datos del tutor

Dra.

Zaida Estefanía

Alarcón

Bernal

3. Datos del sinodal 1

Dr.

Ricardo

Aceves

García

4. Datos del sinodal 2

Dra.

Bibiana

Obregón

Quintana

5. Datos del sinodal 3

Dra.

María del Carmen

Fernández

García

6. Datos del sinodal 4

Mat.

Ana Lilia

Anaya

Muñoz

7. Datos del trabajo escrito

Solución al Problema de Localización

P-Mediana Usando Descomposición

de Benders

115p

2020

Agradecimientos

El primer y más grande agradecimiento, es para mis padres José Manuel Cruz Palma Ortega y María Mercedes Cabrera Moreno, así como a mis hermanos el M.A. José Manuel Palma Cabrera y el L.C. Habid Ariel Palma Cabrera. Gracias por todo su cariño y apoyo que siempre tienen conmigo. No encuentro palabras para agradecerles por tantos esfuerzos y sacrificios que hicieron para que yo pudiera llegar a este momento.

Otro enorme agradecimiento es para la Dra. Zaida Estefanía Alarcón Bernal, ya que gracias a su ayuda, paciencia, y guía pude realizar este trabajo. Asimismo agradezco ampliamente al Dr. Ricardo Aceves García por sus consejos y apoyo que tuvo conmigo para poder realizar este trabajo. Además de un gran agradecimiento a mis sinodales, la Dra. Bibiana Obregón Quintana, la Dra. María del Carmen Fernández García, y la Mat. Ana Lilia Anaya Muñoz, por tomarse el tiempo para revisar mi trabajo.

También quiero agradecer a todos los amigos que hice durante la carrera por todos los buenos momentos que compartimos, pero en especial, quiero agradecer a mis buenos amigos Diego Antonio García Mendoza y Dayana Claudia Cruz Delgado, por el buen equipo que formamos durante la carrera. Sin olvidar a mis amigos del laboratorio OPTISERV, pero en especial quiero darle las gracias a la Mat. Nilse Pamela Romero Basurto por la ayuda y consejos que me brindo.

Continuando con los agradecimientos, quiero darle las gracias a la Universidad Nacional autónoma de México y a la Facultad de Ciencias por darme la oportunidad de lograr uno de mis más grandes sueños, el de ser un profesionalista.

Y finalmente un agradecimiento especial al proyecto PAPIIT con número de expediente IN116217 “Optimización de los procesos y operaciones en las empresas de servicios”. Por el apoyo económico para poder realizar este trabajo.

Índice general

1	Introducción	1
2	El Problema de Localización de Servicios	3
2.1	Localización continua	4
2.2	Localización discreta	6
2.3	Localización en redes	8
2.3.1	Conjunto de Cobertura	9
2.3.2	Conjunto de Cobertura Máxima	11
2.3.3	Modelo del P-centro	13
2.3.4	Modelo de la P-mediana	15
2.3.5	Modelo de la P-centdiana	17
3	Modelo de Localización de la P-mediana	20
3.1	Propiedades del modelo de la P-mediana	21
3.2	Complejidad computacional	25
3.3	El problema P-mediana en un árbol: el caso de la 1-mediana	28
3.4	Modelo del problema de localización de la planta capacitada	33
4	Métodos de Solución para Resolver el Problema de Localización P-mediana	36
4.1	Métodos Exactos	36
4.1.1	Método de los planos de corte	37
4.1.2	Ramificación y acotamiento	38
4.2	Métodos Heurísticos	42
4.2.1	Miope	42
4.2.2	Algoritmo de búsqueda en vecindarios	43
4.2.3	Intercambio	44
5	Solución al Problema de Localización P-mediana Usando Descomposición de Benders	45

5.1	Descomposición de Benders	45
5.2	Principio de Descomposición de Benders	46
5.3	Algoritmo de Descomposición de Benders	48
5.4	Solución al Problema de Localización P-mediana Usando Descomposición de Benders	50
6	Caso de Estudio	59
6.1	Localización de moto-ambulancias para la atención de accidentes de tránsito en la Ciudad de México	59
7	Conclusiones	69
	Bibliografía	72
	Apéndice	75
A	Datos del problema del caso de estudio	75
B	Algoritmo escrito en GAMS	87
C	Captura de pantalla de GAMS	90

Índice de tablas

Tabla 5.1	Tabla de las distancias entre los sitios candidatos con los sitios de demanda	52
Tabla 5.2	Costo de cada instalación de los sitios candidatos	52
Tabla 5.3	Capacidad de cada sitio candidato	53
Tabla 5.4	Demanda de los sitios	53
Tabla 6.1	Cobertura de los puntos de accidentes por los sitios seleccionados.	66
Tabla A.1	Tabla de los costos por localizar las instalaciones en los sitios candidatos.	76
Tabla A.2	Tabla de los costos de los vehículos, incluyen el mantenimiento de los vehículos, y el sueldo de los paramédicos.	78
Tabla A.3	Tabla del promedio de los accidentes ocurridos en el año 2018. . .	80
Tabla A.4	Distancia de los nodos de demanda a los sitios candidatos.	81
Tabla A.5	Distancia de los nodos de demanda a los sitios candidatos.	82
Tabla A.6	Distancia de los nodos de demanda a los sitios candidatos.	83
Tabla A.7	Distancia de los nodos de demanda a los sitios candidatos.	84
Tabla A.8	Distancia de los nodos de demanda a los sitios candidatos.	85
Tabla A.9	Distancia de los nodos de demanda a los sitios candidatos.	86

Índice de figuras

Figura 2.1	Principales modelos de localización.	4
Figura 6.1	Sitios candidatos para localizar las bases de emergencias.	61

Figura 6.2	Principales puntos de accidentes viales.	62
Figura 6.3	Mapa con las soluciones obtenidas por GAMS.	67
Figura 6.4	Cobertura de los sitios candidatos.	68
Figura C.1	Captura de pantalla del algoritmo en GAMS	91
Figura C.2	Captura de pantalla del algoritmo en GAMS	92
Figura C.3	Captura de pantalla del algoritmo en GAMS	93
Figura C.4	Captura de pantalla del algoritmo en GAMS	94
Figura C.5	Captura de pantalla del algoritmo en GAMS	95
Figura C.6	Captura de pantalla del algoritmo en GAMS	96
Figura C.7	Captura de pantalla del algoritmo en GAMS	97
Figura C.8	Captura de pantalla del algoritmo en GAMS	98
Figura C.9	Captura de pantalla del algoritmo en GAMS	99
Figura C.10	Captura de pantalla del algoritmo en GAMS	100
Figura C.11	Captura de pantalla del algoritmo en GAMS	101
Figura C.12	Captura de pantalla del algoritmo en GAMS	102
Figura C.13	Captura de pantalla del algoritmo en GAMS	103
Figura C.14	Captura de pantalla del algoritmo en GAMS	104
Figura C.15	Captura de pantalla del algoritmo en GAMS	105
Figura C.16	Captura de pantalla del algoritmo en GAMS	106
Figura C.17	Captura de pantalla del algoritmo en GAMS	107
Figura C.18	Captura de pantalla del algoritmo en GAMS	108
Figura C.19	Captura de pantalla del algoritmo en GAMS	109

Capítulo 1

Introducción

La Investigación de Operaciones es el área de las matemáticas que se enfoca en dar solución a problemas relacionados con la toma de decisiones de diversos conflictos del mundo real, identificando la mejor dirección posible para actuar. Dentro de esta área, se encuentra un tipo bastante singular de ellos: los problemas de localización de instalaciones. En este trabajo se hablará acerca de los problemas de localización de instalaciones, enfocándonos especialmente en el modelo de localización de la P-mediana.

El modelo de localización de la P-mediana tiene una amplia gama de aplicaciones, por ejemplo, localizar oficinas gubernamentales, bancos, escuelas, centros comerciales, tiendas minoristas, bases de defensa militar, entre muchos más. Entre estas aplicaciones se encuentra la localización de servicios médicos de emergencia.

Por lo mencionado anteriormente, la resolución de problemas utilizando el modelo de la P-mediana es de mucha utilidad en la vida real, por eso, para este trabajo se busca resolver el modelo de una manera distinta a las que existen y que habitualmente se usan para resolverlo. Una alternativa para dar solución puede ser a través de la descomposición de Benders. Ya que este método heurístico, podría resultar mejor que los métodos tradicionales, debido a que con la descomposición de Benders, se logra resolver el problema de forma simultánea e iterativa, con lo cual se vuelve más fácil encontrar una solución. Para comprobarlo vamos a ponerlo a prueba en el caso de estudio, el cual tratará acerca de la localización de servicios médicos de emergencia en la Ciudad de México. Debido al elevado número de fallecimientos, lesiones, y daños materiales que resultan cada año a causa de los accidentes viales.

Dicho lo anterior, en el capítulo 2 se hablará sobre algunos de los tipos más importantes de modelos de localización que existen, los cuales son: el modelo de localización continua, el modelo de localización discreta y los modelos de localización en redes. Se va a profundizar más sobre el último conjunto de modelos, para mostrar sus principales clases, las cuales son: el conjunto de cobertura, el conjunto de cobertura máxima, el modelo de localización del P-centro, el modelo de localización de la P-mediana, y el modelo de localización de la P-centdiana, para posteriormente enfocarnos en el modelo de localización de la P-mediana, el cual será el tema central de este trabajo.

En el capítulo 3 se mencionarán algunas de las propiedades más destacables del modelo de la P-mediana, también se hablará de forma breve acerca de la complejidad computacional y del modelo de la P-mediana en un árbol. Posteriormente se va a tratar con una variante del modelo de localización de la P-mediana: el problema de localización de la planta capacitada.

Después en el capítulo 4 se hablará acerca de algunas de las formas más comunes que se pueden utilizar para resolver el modelo de localización de la P-mediana, las cuales se dividen en métodos exactos (método de los planos de corte, y ramificación y acotamiento) y métodos heurísticos (miope, búsqueda en vecindarios, e intercambio). Luego, en el capítulo 5 se tratará el tema de la descomposición de Benders, método que se utilizará para la resolución del modelo de este trabajo, el cual es un método iterativo que consiste en dividir el problema en una parte complicada “problema maestro” y en un subproblema.

En el capítulo 6 se tratará el caso de estudio, el cual consistirá en la localización de bases de servicios médicos de emergencia, las cuales tienen por función brindar el servicio prehospitalario. Finalmente en el capítulo 7 se darán las conclusiones obtenidas de este trabajo.

Capítulo 2

El Problema de Localización de Servicios

Los problemas de localización son problemas que pueden ser formulados como problemas de programación matemática y que tienen como finalidad seleccionar la opción más adecuada para ubicar instalaciones (las cuales brindan algún tipo de servicio), considerando las condiciones óptimas de su entorno y los requerimientos del problema que se está modelando.

Los principales objetivos en este tipo de problemas son la minimización del costo por construcción o ubicación de la instalación, así como la maximización del beneficio que se puede obtener de ella, ya sea económico o de algún otro tipo, para de esta forma lograr el mejor desempeño de la instalación. Por otra parte, es necesario considerar, entre otras cosas, que la cantidad de instalaciones a utilizar debe ser la adecuada para satisfacer la demanda, el tamaño que deben tener las instalaciones debe ser el correcto. Al resolver el problema se conocerá el lugar en donde debe ser ubicada cada instalación, y la manera en la que se debe asignar la demanda de los servicios de las instalaciones para cada instalación [7].



Figura 2.1: Principales modelos de localización.

2.1. Localización continua

Los problemas de localización continua son los problemas más antiguos, en estos, el problema es representado geoméricamente y se aprovecha la continuidad de la zona de ubicación. En este tipo de problemas la demanda ocurre en cualquier parte del plano y las instalaciones también pueden ser localizadas en cualquier parte de éste [10]. Además de que, las distancias se piensan para ser distancias euclidianas, aunque la mayoría de las funciones de distancia se pueden usar. Para el problema de localización continua se puede suponer que los sitios de demanda pueden ser vistos como puntos en el espacio. Esto puede ser cierto siempre y cuando las dimensiones de los sitios de demanda sean pequeñas en relación con las distancias entre las instalaciones y los sitios de demanda [9].

Nuestra meta, es poder formular el problema de localización continua como un problema de programación lineal. Teniendo en cuenta que lo que se quiere es minimizar la distancia que hay entre sitios de demanda con sus instalaciones más cercanas, además de considerar un peso entre las instalaciones y los sitios de demanda, el cual dependerá de cada pro-

blema. No se contará con restricciones para este tipo de problemas. Por lo anteriormente expuesto, el problema puede ser formulado como:

$$\text{Minimizar} \quad \sum_{i=1}^m \sum_{j=1}^n r_{ij} l_p(X_i, a_j) + \sum_{i=1}^m \sum_{j=1}^n s_{ij} l_p(X_i, X_r) \quad (2.1)$$

Donde definimos la siguiente notación:

Conjuntos

$I = \{1, \dots, i, \dots, m\}$ Número de instalaciones por localizar

$J = \{1, \dots, j, \dots, n\}$ Número de sitios de demanda

Parámetros

r_{ij} = Peso entre los sitios de demanda y las instalaciones

s_{ij} = Peso entre las instalaciones

$l_p(X_i, a_j)$ = Medida de distancia entre los sitios de demanda y las instalaciones

$l_p(X_i, X_r)$ = Medida de distancia entre las instalaciones

Variables

X_i = Variable de decisión que corresponde a las coordenadas de localización de las nuevas instalaciones

X_r = Variable de decisión que corresponde a las coordenadas de localización de las nuevas instalaciones

a_j = Coordenadas de localización de los sitios de demanda

La medida de la distancia para este caso es la norma:

$$l_p(X_i, a_j) = [|x_1 - a_1|^p + |x_2 - a_2|^p]^{1/p}, (\text{con } p \geq 1) \quad (2.2)$$

[10].

Un ejemplo de la aplicación del problema de localización continua en el plano podría ser la colocación de helicópteros para la recolección y traslado de pacientes con algún trauma. Mientras que la ubicación de una ambulancia o una grúa a lo largo de un tramo de carretera

es una aplicación de un problema de localización continua en una red. [21]

2.2. Localización discreta

Los problemas de localización discreta son problemas en los cuales el conjunto de las posibles ubicaciones de los sitios candidatos para las instalaciones es finito, en este tipo de problemas se permite usar distancias arbitrarias entre las instalaciones y los sitios de demanda. Los problemas de localización discreta también son conocidos como problemas de selección de sitios ya que tenemos un conocimiento previo de los sitios candidatos para las instalaciones [7].

Nuestro fin es formular al problema de localización discreta como un problema de programación lineal entero binario. Lo que se busca al plantear este tipo de problema es minimizar los costos de localizar las instalaciones que atienden los sitios de demanda. De esta manera para cada uno de los sitios candidatos para localizar las instalaciones, se va a considerar una variable binaria, la cual tomara el valor 1 si el sitio, es un sitio candidato seleccionado, y tomara el valor de 0 en caso contrario.

Ahora para cada ruta que exista entre un sitio candidato con un sitio de demanda vamos a definir una variable binaria. Dicha variable será igual a 1 cuando un sitio de demanda sea atendido por una instalación, y tomará el valor de 0 en caso contrario. También es importante que las demandas sean atendidas por instalaciones ya establecidas, y asegurándose que todas las demandas sean satisfechas. Otra cosa importante es que en este tipo de problema se quiere evitar conflictos en la capacidad de las instalaciones. Así mismo este problema tiene la posibilidad de incluir límites para cada una de las variables antes mencionadas. Por lo anteriormente expuesto, el problema puede ser formulado como:

$$\text{Minimizar} \quad \sum_{i=1}^m \sum_{j=1}^n C_{ij} x_{ij} + \sum_{i=1}^m f_i y_i \quad (2.3)$$

$$\text{Sujeto a :} \quad \sum_{i=1}^m x_{ij} = 1 \quad \forall j \quad (2.4)$$

$$x_{ij} = y_i \quad \forall i, j \quad (2.5)$$

$$\sum_{j=1}^n d_j x_{ij} \leq s_i y_i \quad \forall i \quad (2.6)$$

$$\sum_{i=1}^m a_{ij} x_{ij} + \sum_{i=1}^m b_i y_i \leq r \quad \forall j \quad (2.7)$$

$$x_{ij} \in \{0, 1\} \quad \forall i, j \quad (2.8)$$

$$y_i \in \{0, 1\} \quad \forall i \quad (2.9)$$

Donde definimos la siguiente notación:

Conjuntos

$I = \{1, \dots, i, \dots, m\}$ Número de instalaciones por localizar

$J = \{1, \dots, j, \dots, n\}$ Número de sitios de demanda

Parámetros

C_{ij} = Costo de la instalación i por atender al nodo de demanda j

f_i = Costo fijo por localizar la instalación en el sitio i

s_i = Capacidad de la instalación i

d_j = Demanda del nodo j

a_{ij} = Constantes adicionales relacionadas con la variable x_{ij}

b_i = Constantes adicionales relacionadas con la variable y_i

r = Constante de la restricción adicional

Variables

x_{ij} = Fracción de la demanda total atendida por la instalación i , del nodo de demanda j

$$y_i = \begin{cases} 1 & \text{si la instalación } i \text{ está cubierta} \\ 0 & \text{en caso contrario} \end{cases}$$

[10].

La función objetivo 2.3 minimiza el costo por atender la demanda. La restricción 2.4 asegura que la demanda de cada usuario se satisfaga. La restricción 2.5 garantiza que los nodos de demanda serán atendidos únicamente por instalaciones abiertas. La restricción 2.6 evita que se viole la capacidad de las instalaciones abiertas. La restricción 2.7 considera la posibilidad de incluir limitantes (lineales) sobre las variables x_{ij} y y_i . Las restricciones 2.8 y 2.9 especifican que todas las variables de decisión son binarias.

Un ejemplo de la aplicación del problema de localización discreta en el plano es el posicionamiento de las estaciones transmisoras que se ubicarán en algunos puntos permitidos dentro de una región, como cimas de montañas, mientras que una aplicación de problemas de localización de red discreta se encuentra en la ubicación de instalaciones minoristas que solo se puede ubicar en lotes que están zonificados para ellos [21].

2.3. Localización en redes

Los problemas de localización en redes son problemas en los cuales las instalaciones se pueden considerar como nodos de una red, además de que las mismas pueden ser localizadas en los nodos o en las aristas de la red. En este tipo de modelos, se puede suponer que la demanda al igual que los viajes entre las instalaciones y sitios de demanda ocurre dentro de la misma red. En los modelos de localización en redes, se pueden distinguir entre dos tipos de problemas, los primeros son los que se presentan como árboles, y los segundos son los que se expresan en una gráfica más general. Un árbol es un tipo especial de red en la que hay una sola ruta entre cualquier par de nodos, es decir es una gráfica sin ciclos. De los dos tipos de problemas antes mencionados, se presta una mayor atención a los árboles, ya que muchos problemas de la vida real pueden ser vistos como árboles, además de que resolverlos se vuelve fácil en comparación de una red más general [7].

Antes de seguir, es importante introducir unos conceptos importantes. El primer concepto, es que para este trabajo únicamente vamos a considerar gráficas (o redes) finitas no dirigidas, las cuales están formadas por un conjunto de vértices, y un conjunto de aristas que tienen asociados una serie de pesos no negativos. Otro concepto importante, es la distancia

que existe entre cualquier par de vértices en la gráfica (o red). Esta distancia es representada por $d(x, y)$ y se define como la longitud del camino más corto entre cualesquiera par de vértices x y y de la gráfica, donde la longitud del camino es la suma de los pesos de las aristas que lo forman.

Los principales modelos de localización en redes son:

1. **Conjunto de Cobertura**
2. **Conjunto de Cobertura Máxima**
3. **Modelo del P-centro**
4. **Modelo de la P-mediana**
5. **Modelo de la P-centdiana**

2.3.1. Conjunto de Cobertura

El modelo del conjunto de cobertura de instalaciones es un modelo que busca que cada nodo de demanda sea cubierto, lo cual es un requisito muy restrictivo. Ya que, en un problema con muchos nodos de demanda dispersos espacialmente, este requisito de cobertura puede dar como resultado una serie de instalaciones que no son realistas desde el punto de vista presupuestario [21].

Buscamos formular el modelo del conjunto de cobertura como un problema de programación lineal entero binario. Para este modelo buscamos cubrir a cada uno de los nodos de demanda con una instalación, considerando un costo por localizar la instalación. Así, para cada nodo de los sitios candidatos para localizar las instalaciones, se va a considerar una variable binaria, la cual tomará el valor 1 si el nodo es un sitio candidato seleccionado, y tomará el valor de 0 en caso contrario.

Es necesario definir una variable binaria, la cual representa la cobertura del nodo candidato seleccionado a los nodos de demanda. Dicha variable será igual a 1 si el nodo seleccionado cubre al nodo de demanda, y tomará el valor de 0 en caso contrario. Por tal motivo, el problema puede ser formulado como:

$$\text{Minimizar} \quad z = \sum_j f_j x_j \quad (2.10)$$

$$\text{Sujeto a :} \quad \sum_j a_{ij} x_j \geq 1 \quad \forall i \quad (2.11)$$

$$x_j \in \{0, 1\} \quad \forall j \quad (2.12)$$

Donde definimos la siguiente notación:

Conjuntos

$I = \{1, \dots, i, \dots, n\}$ Número de nodos de demanda

$J = \{1, \dots, j, \dots, m\}$ Número de nodos candidatos para localizar las instalaciones

Parámetro

f_j = Costo fijo por localizar la instalación en el sitio j

Variables

$$a_{ij} = \begin{cases} 1 & \text{si la demanda del nodo } i \text{ fue cubierta por la instalación ubicada en el nodo } j \\ 0 & \text{en caso contrario} \end{cases}$$

$$x_j = \begin{cases} 1 & \text{si la instalación se localiza en el sitio candidato } j \\ 0 & \text{en caso contrario} \end{cases}$$

[10].

La función objetivo 2.10 minimiza el costo total de las instalaciones seleccionadas. La restricción 2.11 establece que cada nodo de demanda i debe estar cubierto por al menos una instalación. Teniendo en cuenta que el lado izquierdo de la restricción 2.11 proporciona el número de instalaciones ubicadas que pueden cubrir el nodo de demanda i . La restricción 2.12 especifica que todas las variables de decisión son binarias.

Ejemplos de la aplicación del problema del conjunto de cobertura son, la programación del personal de aerolíneas, localización de redes de defensa (en guerras), distribución de productos, ubicación de almacenes, instalación de servicio de emergencia [9].

2.3.2. Conjunto de Cobertura Máxima

El modelo del conjunto de cobertura máxima, a diferencia del modelo anterior, busca localizar un número de instalaciones económicamente factibles digamos “ P ”, de tal manera que el número de nodos de demanda cubiertos por las instalaciones sea el máximo posible [21].

Nuestro fin, es formular al modelo del conjunto de cobertura máxima como un problema de programación lineal entero binario. Una diferencia importante con el modelo anterior, es que debemos considerar cubrir la mayor cantidad de nodos de demanda con una cantidad finita de instalaciones. Por tal motivo, para cada nodo de los sitios candidatos para localizar las instalaciones, vamos a considerar una variable binaria, la cual tomará el valor 1 si el nodo es un sitio candidato seleccionado, y tomará el valor de 0 en caso contrario. Por la condición anterior es necesario limitar la cantidad de instalaciones disponibles.

Vamos a definir una variable binaria, dicha variable será igual a 1 si el nodo seleccionado cubre al nodo de demanda, y tomará el valor de 0 en caso contrario. En este modelo también es importante considerar la demanda que cada nodo tiene, así como si el nodo de demanda está cubierto, y la vamos a representar con una variable binaria la cual será 1 si el nodo de demanda está cubierto y será 0 en caso contrario. Por lo anteriormente expuesto, el problema puede ser formulado como:

$$\text{Maximizar} \quad G = \sum_i h_i Z_i \quad (2.13)$$

$$\text{Sujeto a :} \quad \sum_j a_{ij} x_j \geq Z_i \quad \forall i \quad (2.14)$$

$$\sum_j x_j \leq P \quad (2.15)$$

$$x_j \in \{0, 1\} \quad \forall j \quad (2.16)$$

$$Z_i \in \{0, 1\} \quad \forall i \quad (2.17)$$

Donde definimos la siguiente notación:

Conjuntos

$I = \{1, \dots, i, \dots, n\}$ Número de nodos de demanda

$J = \{1, \dots, j, \dots, m\}$ Número de nodos candidatos para localizar las instalaciones

Parámetros

h_i = Demanda en el nodo i

P = Cantidad de instalaciones por localizar

Variables

$$a_{ij} = \begin{cases} 1 & \text{si la demanda del nodo } i \text{ fue cubierta por la instalación ubicada en el nodo } j \\ 0 & \text{en caso contrario} \end{cases}$$

$$x_j = \begin{cases} 1 & \text{si la instalación se localiza en el sitio candidato } j \\ 0 & \text{en caso contrario} \end{cases}$$

$$Z_i = \begin{cases} 1 & \text{si el nodo } i \text{ está cubierto} \\ 0 & \text{en caso contrario} \end{cases}$$

[10].

La función objetivo 2.13 maximiza el número de nodos de demandas cubiertos. La restricción 2.14 establece que la demanda en el nodo i no se puede cubrir a menos que se seleccione al menos uno de los sitios de la instalación que cubre el nodo i . Cabe recalcar que el lado izquierdo de la restricción 2.14, proporciona el número de instalaciones seleccionadas que pueden cubrir el nodo i . La restricción 2.15 establece que no localizamos más de P instalaciones. Teniendo en cuenta que a menos de que P supere la cantidad de instalaciones necesarias para cubrir todos los nodos de demanda, la restricción 2.15 será vinculante en la solución óptima. Finalmente, las restricciones 2.16 y 2.17 especifican que todas las variables de decisión son binarias.

Ejemplos de la aplicación del problema del conjunto de cobertura máxima son, la localización de instalaciones de servicios y vehículos de respuesta para emergencias, redes de comunicación, tiendas minoristas y sensores de vigilancia de seguridad [24].

2.3.3. Modelo del P-centro

El modelo del P-centro es un modelo que tiene como objetivo ubicar “ P ” instalaciones de manera que se cubran todas las demandas, minimizando la distancia máxima entre los nodos de demanda y las instalaciones. A cada uno de los puntos de demanda se le asocia un peso, el cual puede tener un significado diferente como puede ser el tiempo por unidad de distancia, costo por unidad de distancia o pérdida por unidad de distancia, el cual se busca optimizar (maximizar o minimizar) [9].

Buscamos formular al modelo del P-centro como un problema de programación lineal entero binario. Para el modelo del P-centro queremos minimizar la distancia máxima que existe entre sitios de demanda con sus instalaciones. Así, para cada nodo de los sitios candidatos para localizar las instalaciones, vamos a considerar una variable binaria, la cual tomará el valor 1 si el nodo es un sitio candidato seleccionado, es decir es un centro, y tomará el valor de 0 en caso contrario. Vamos a tener un límite en el número de centros que se pueden instalar digamos “ P ” (de ahí el nombre P-centro).

Después, la distancia que exista entre cada uno de los nodos candidatos con los nodos de demanda se representara con una arista. Definimos una variable binaria, esta variable será igual a 1 si el nodo es el centro más cercano al nodo de demanda, y tomará el valor de 0 en caso contrario. Además de que únicamente será posible asignar una demanda a centros que estén abiertos, y asegurándonos de que se satisfaga toda la demanda. Otra consideración importante es que no se debe exceder la distancia máxima que hay entre un sitio de demanda con su instalación más próxima. Por lo anteriormente expuesto, el problema puede ser formulado como:

$$\text{Minimizar} \quad W \quad (2.18)$$

$$\text{Sujeto a :} \quad \sum_j Y_{ij} = 1 \quad \forall i \quad (2.19)$$

$$\sum_j X_j = P \quad (2.20)$$

$$Y_{ij} \leq X_j \quad \forall i, j \quad (2.21)$$

$$W \geq \sum_j d_{i,j} Y_{i,j} \quad \forall i \quad (2.22)$$

$$X_j \in \{0, 1\} \quad \forall j \quad (2.23)$$

$$Y_{ij} \in \{0, 1\} \quad \forall i, j \quad (2.24)$$

Donde definimos la siguiente notación:

Conjuntos

$I = \{1, \dots, i, \dots, n\}$ Número de nodos de demanda

$J = \{1, \dots, j, \dots, m\}$ Número de nodos candidatos para localizar las instalaciones

Parámetros

W = Distancia máxima entre un nodo de demanda y la instalación más cercana

d_{ij} = Distancia entre el nodo de demanda i y el nodo candidato j

P = Número de instalaciones por ubicar

Variables

$$Y_{ij} = \begin{cases} 1 & \text{si el nodo de demanda } i \text{ se asigna a la instalación en el nodo candidato } j \\ 0 & \text{en caso contrario} \end{cases}$$

$$X_j = \begin{cases} 1 & \text{si la instalación se ubica en el sitio candidato } j \\ 0 & \text{en caso contrario} \end{cases}$$

[7].

La función objetivo 2.18 minimiza la distancia máxima entre un nodo de demanda con su instalación más cercana. La restricción 2.19 indica que toda demanda de los nodos i

debe ser satisfecha. La restricción 2.20 garantiza que las P instalaciones estén ubicadas. La restricción 2.21 asegura que la asignación solamente será para instalaciones abiertas. En las restricciones 2.22 establece que no se exceda la distancia máxima entre un nodo de demanda con su instalación más próxima. Las restricciones 2.23 y 2.24 especifican que todas las variables de decisión son binarias.

El modelo del P-centro es utilizado para localizar estaciones de bomberos, policía o ambulancias, unidades de urgencias, etc. Pensando en los centros de atención médica, por ejemplo, hay ocasiones en las que más que minimizar la distancia del usuario más lejano con su instalación de servicios más cercana, lo que se busca es maximizar el número de usuarios potenciales atendidos en un tiempo aceptable [1].

2.3.4. Modelo de la P-mediana

El modelo de la P-mediana, es un modelo que tiene el objetivo de ubicar en el mejor lugar posible “ P ” instalaciones de tal forma que se minimice la distancia promedio ponderada entre los nodos de demanda con las instalaciones que son designadas. Este modelo pertenece a los modelos de distancia total o distancia promedio [9].

Nuestro objetivo es poder formular al modelo de la P-mediana como un problema de programación lineal entero binario. En el modelo de la p-mediana queremos minimizar la distancia promedio ponderada que existe entre sitios de demanda con su instalación más cercana, donde la ponderación dependerá de cada problema. De este modo para cada nodo que represente a los sitios candidatos para localizar las instalaciones, se va a considerar una variable binaria, la cual tomará el valor 1 si el nodo es un sitio candidato seleccionado, es decir es una mediana, y tomará el valor de 0 en caso contrario. Igualmente se tiene un límite en el número de medianas que se pueden instalar digamos “ P ” (por eso el nombre P-mediana).

Después, la distancia que exista entre cada uno de los nodos candidatos con los nodos de demanda se representara con una arista. Definimos una variable binaria, dicha variable será igual a 1 si el nodo seleccionado es la mediana más cercana al nodo de demanda, y tomará el valor de 0 en caso contrario. Además de que únicamente será posible asignar una demanda a medianas que están instaladas, fijándonos en que la demanda total se satisfaga. Por lo anteriormente expuesto, el problema puede ser formulado como:

$$\text{Minimizar} \quad \sum_i \sum_j h_i d_{ij} Y_{ij} \quad (2.25)$$

$$\text{Sujeto a :} \quad \sum_j Y_{ij} = 1 \quad \forall i \quad (2.26)$$

$$\sum_j X_j = P \quad (2.27)$$

$$Y_{ij} - X_j \leq 0 \quad \forall i, j \quad (2.28)$$

$$X_j \in \{0, 1\} \quad \forall j \quad (2.29)$$

$$Y_{ij} \in \{0, 1\} \quad \forall i, j \quad (2.30)$$

Donde definimos la siguiente notación:

Conjuntos

$I = \{1, \dots, i, \dots, n\}$ Número de nodos de demanda

$J = \{1, \dots, j, \dots, m\}$ Número de nodos candidatos para localizar las instalaciones

Parámetros

h_i = Demanda en el nodo i

d_{ij} = Distancia entre el nodo de demanda i y el nodo candidato j

P = Número de instalaciones por ubicar

Variables

$$Y_{ij} = \begin{cases} 1 & \text{si la demanda del nodo } i \text{ fue atendida por la instalación ubicada en el nodo } j \\ 0 & \text{en caso contrario} \end{cases}$$

$$X_j = \begin{cases} 1 & \text{si la instalación está ubicada en el nodo } j \\ 0 & \text{en caso contrario} \end{cases}$$

[7].

La función objetivo 2.25 minimiza la suma ponderada entre las demandas y las distancias. La restricción 2.26 indica que la demanda de cada centro debe ser satisfecha en su

totalidad. La restricción 2.27 garantiza que el número total de instalaciones sea P . La restricción 2.28 garantiza que, la demanda sólo puede ser atendida por la instalación j si esta está abierta. Las restricciones 2.29 y 2.30 especifican que todas las variables de decisión son binarias.

El modelo de la P-mediana es utilizado para localizar tiendas minoristas, oficinas de correos, cajeros automáticos, estaciones de bomberos, policía o ambulancias, unidades de urgencias, etc [9]. Pero a diferencia del modelo P-centro, en este modelo se busca minimizar la distancia total (o media) ponderada recorrida para atender la demanda.

2.3.5. Modelo de la P-centdiana

El modelo de la P-centdiana es un modelo en el cual se busca encontrar “ P ” puntos que minimicen una combinación convexa de los dos modelos anteriores; el modelo de la P-mediana y el modelo del P-centro [9].

Para este problema, se busca minimizar la función objetivo para una determinada λ .

Donde:
$$\lambda \in \{0, 1\}$$

Queremos formular al modelo de la P-centdiana como un problema de programación lineal entero binario. Lo que buscamos en el modelo de la P-centdiana es minimizar la distancia máxima que existe entre nodos de demanda con las instalaciones a la vez que buscamos minimizar la distancia promedio ponderada que hay entre nodos de demanda con las instalaciones. De esta manera para cada nodo de los sitios candidatos para localizar las instalaciones, se va a considerar una variable binaria, la cual tomará el valor 1 si el nodo es un sitio candidato seleccionado, es decir es una centdiana, y tomará el valor de 0 en caso contrario. Igualmente que en los modelos anteriores, se tendrá un límite en el número de centdianas que se pueden instalar digamos “ P ”.

Después, la distancia que exista entre cada uno de los nodos candidatos con los nodos de demanda se representara con una arista. Definimos una variable binaria, dicha variable será igual a 1 si el nodo seleccionado de los sitios candidatos es la centdiana más cercana al nodo de demanda, y tomará el valor de 0 en caso contrario. Además de que únicamente será posible asignar una demanda a centdianas que están abiertas, fijándonos en que la demanda total se satisfaga. Otra consideración importante es que no se debe exceder la

distancia máxima que hay entre un nodo de demanda con su instalación más próxima. Dicho lo anterior, el problema lo podemos formular como:

$$\text{Minimizar} \quad \lambda \sum_i \sum_j h_i d_{ij} Y_{ij} + (1 - \lambda) W \quad (2.31)$$

$$\text{Sujeto a :} \quad \sum_j Y_{ij} = 1 \quad \forall i \quad (2.32)$$

$$\sum_j X_j = P \quad (2.33)$$

$$Y_{ij} \leq X_j \quad \forall i, j \quad (2.34)$$

$$W \geq \sum_j d_{i,j} Y_{i,j} \quad \forall i \quad (2.35)$$

$$X_j \in \{0, 1\} \quad \forall j \quad (2.36)$$

$$Y_{ij} \in \{0, 1\} \quad \forall i, j \quad (2.37)$$

Donde definimos la siguiente notación:

Conjuntos

$I = \{1, \dots, i, \dots, n\}$ Número de nodos de demanda

$J = \{1, \dots, j, \dots, m\}$ Número de nodos candidatos para localizar las instalaciones

Parámetros

W = Distancia máxima entre un nodo de demanda y la instalación más cercana

h_i = Demanda en el nodo i

d_{ij} = Distancia entre el nodo de demanda i y el nodo candidato j

P = Número de instalaciones por ubicar

Variables

$$Y_{ij} = \begin{cases} 1 & \text{si la demanda del nodo } i \text{ fue atendida por la instalación ubicada en el nodo } j \\ 0 & \text{en caso contrario} \end{cases}$$

$$X_j = \begin{cases} 1 & \text{si la instalación está ubicada en el nodo } j \\ 0 & \text{en caso contrario} \end{cases}$$

La función objetivo 2.31 minimiza la combinación convexa de los objetivos del modelo P-centro y P-mediana. La restricción 2.32 asegura que la demanda se satisfice. La restricción 2.33 garantiza que se seleccionen exactamente P localizaciones. La restricción 2.34 asegura que los nodos de demanda sean asignados a una centdiana sólo si ésta ha sido seleccionada. La restricción 2.35 garantiza que no se exceda la distancia máxima entre un nodo de demanda con su instalación más próxima. Por ultimo las restricciones 2.36 y 2.37 especifican que todas las variables de decisión son binarias.

El modelo de la P-centdiana puede ser utilizado para localizar instalaciones de servicios tales como centros administrativos, centros de salud, centros comerciales, estaciones de autobuses etc. El objetivo que se busca con este modelo es el de reducir la distancia total por recorrer entre los nodos de demanda y las instalaciones (objetivo del modelo de la P-mediana), pero tratando que la distancia por recorrer entre cualquier nodo de demanda con las instalaciones no sea exageradamente grande (objetivo del modelo del P-centro) [6].

Capítulo 3

Modelo de Localización de la P-mediana

Los problemas de localización de instalaciones son algunos de los problemas más antiguos que han acompañado a la humanidad. Una de las primeras personas en plantearse un tipo de problema de localización fue Cavalier, él considero un problema en el cual buscaba determinar un punto en el plano, cuya suma de sus distancias a tres puntos dados sea mínima. Luego Fagnano, demostró que la suma de las distancias a los vértices de un cuadrilátero es mínima en la intersección de las diagonales. Posteriormente Alfred Weber y Walter Isard, cimentaron las bases de la teoría de localización actual, ellos aplicaron la teoría anterior en la localización industrial.

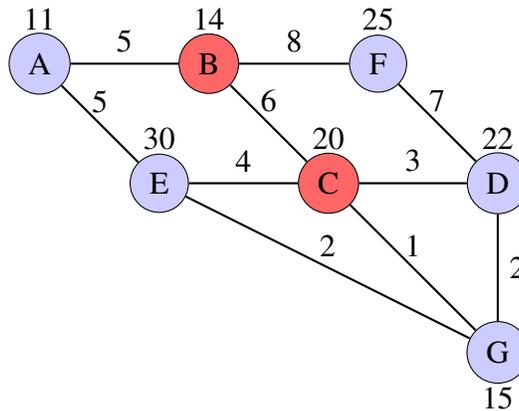
Hasta finalmente llegar al modelo de localización de la P-mediana, el cual remonta sus orígenes al trabajo de Hakimi titulado “Optimum locations of switching centers and the absolute centers and medians of a graph”, el cual fue publicado en el año de 1964 [12].

De los modelos en redes que anteriormente mencionamos, vamos a seleccionar y usar el modelo de la P-mediana (descrito en el capítulo 2) debido a las características con las que cuenta este modelo, ya que para poder resolver el caso de estudio que se va a tratar más adelante, es de suma importancia que la distancia promedio ponderada entre las instalaciones (bases de emergencia) y los puntos de demanda (accidentes viales) sea la mínima posible para así poder responder de la mejor manera. Con el objetivo de minimizar las muertes y los diversos daños ocasionados por los accidentes.

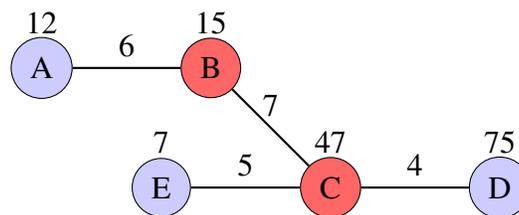
3.1. Propiedades del modelo de la P-mediana

La primera de las propiedades es que se puede representar en una gráfica. Además de que puede ser formulado como un modelo de programación lineal entera 0 – 1 (binario).

El primer ejemplo, es la representación del de la modelo P-mediana en una gráfica general en donde los nodos de color rojo representan los sitios candidatos para localizar las medianas, y el resto representa el conjunto de los sitios de demanda. Aclarando que la demanda es la que aparece sobre cada uno de los nodos y la distancia es la que aparece sobre las aristas.



El segundo ejemplo, es la representación del modelo P-mediana en un árbol, donde los nodos de color rojo representan el conjunto de los sitios candidatos para localizar las medianas, y el resto representa el conjunto de los sitios de demanda. Una observación importante es que en algunas ocasiones se puede tener un conjunto de nodos con los sitios candidatos para localizar las instalaciones y otro conjunto de nodos con los sitios de demanda. En otras ocasiones el conjunto de nodos candidatos para localizar las instalaciones y el conjunto de nodos con los sitios de demanda coinciden, todo dependerá del problema que se modele.

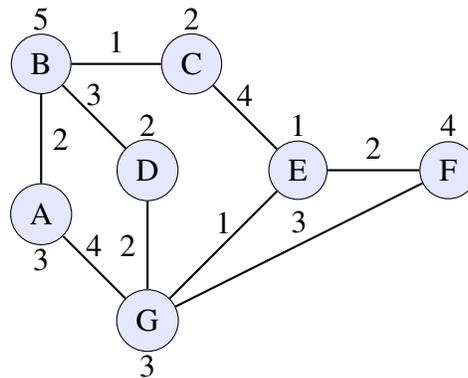


La segunda propiedad, se les debe a Kariv y Hakimi: demostraron que el problema P-mediana es NP-Duro en las gráficas en general, incluso cuando las gráficas fueran una estructura simple, así como las que se pueden dibujar en el plano sin que sus aristas se crucen, es decir en graficas planas y cuyo grado máximo sea tres. La demostración de esta propiedad se puede encontrar en [17].

La tercera propiedad se la debemos a Hakimi. Él propuso que una solución óptima para el problema P-mediana consiste en localizar las instalaciones únicamente en los nodos [13]. Para demostrar que esta propiedad es cierta, vamos a suponer que existe una solución en la cual una instalación se encuentra localizada en la arista (A, B) a una distancia de “ r ” unidades del nodo A , en donde $0 < r < d(A, B)$, siendo $d(A, B)$ la distancia que existe entre el nodo A y B . Sea D_A la demanda total que atiende la instalación que se encuentra en la arista (A, B) , y que ingresa a través del nodo A . De manera análoga definimos a D_B como la demanda total que atiende la instalación que se encuentra en la arista (A, B) , y que ingresa a través del nodo B . Ahora sin pérdida de generalidad supongamos que $D_A > D_B$. Así, al mover la instalación que se encuentra en la arista (A, B) , “ r ” unidades al nodo A , sin cambiar ninguna de las demandas ya asignadas, vamos a cambiar el valor en la función objetivo por $(D_B - D_A)r$. Como $D_A > D_B$, el valor que se va a obtener no es positivo. Por lo tanto, al realizar este cambio en la localización de la instalación, se reducirá el valor de la solución. Con lo cual podemos concluir que al menos una solución óptima para el problema de la P-mediana consiste en localizar las instalaciones en los nodos.

Y una última propiedad es que la distancia total ponderada por los nodos de demanda disminuye con la adición de cada instalación subsecuente. Esto es siempre cierto, ya que al existir una solución óptima para el problema con las “ P ” instalaciones, al agregar una instalación en cualquiera de los nodos candidatos que no tengan una instalación, disminuirá la distancia ponderada de los nodos de demanda y, por lo tanto, disminuirá el valor de la función objetivo [18].

Para ver cómo funciona esta propiedad, consideremos el siguiente ejemplo que consiste en una gráfica ponderada R , para la cual nos gustaría encontrar una 2-mediana. Cabe señalar que en la gráfica R , la demanda es la que se encuentra sobre los nodos, y la distancia es la que aparece sobre las aristas.



Entonces, el primer paso consiste en calcular la matriz de distancia de R . Sea D la matriz de distancia donde $d(i, j)$, es la distancia que hay entre el nodo i y el nodo j .

Así la matriz de distancia queda de la siguiente manera:

$$D = \begin{pmatrix} A & B & C & D & E & F & G \\ 0 & 2 & 3 & 5 & 5 & 7 & 4 \\ 2 & 0 & 1 & 3 & 5 & 7 & 5 \\ 3 & 1 & 0 & 4 & 4 & 6 & 5 \\ 5 & 3 & 4 & 0 & 3 & 5 & 2 \\ 5 & 5 & 4 & 3 & 0 & 2 & 1 \\ 7 & 7 & 6 & 5 & 2 & 0 & 3 \\ 4 & 5 & 5 & 2 & 1 & 3 & 0 \end{pmatrix}$$

El siguiente paso es calcular la matriz D^* , la cual se obtiene al multiplicar cada una de las filas de D por el peso del vértice (demanda). De esta manera, la matriz D^* queda de la siguiente manera:

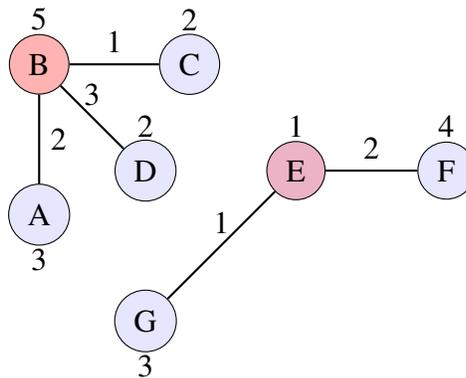
$$D^* = \begin{pmatrix} A & B & C & D & E & F & G \\ 0 & 6 & 9 & 15 & 15 & 21 & 12 \\ 10 & 0 & 5 & 10 & 25 & 35 & 25 \\ 6 & 2 & 0 & 8 & 8 & 12 & 10 \\ 10 & 6 & 8 & 0 & 6 & 10 & 4 \\ 5 & 5 & 4 & 3 & 0 & 2 & 1 \\ 28 & 28 & 24 & 20 & 8 & 0 & 12 \\ 12 & 15 & 15 & 6 & 3 & 9 & 0 \end{pmatrix}$$

Para encontrar una 2-mediana de R , debemos examinar la siguiente suma para todos los nodos i, j .

$$\sum_{r=1}^{r=n} \min [d^*(i, r)d^*(j, r)] \text{ donde } (1 \leq i, j \leq n)$$

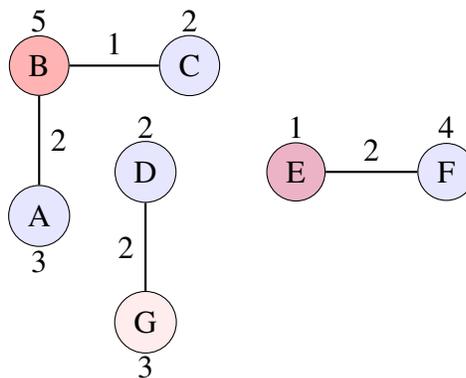
La suma anterior es mínima cuando $i = B$, y $j = E$. Y el valor de la suma es de 25 unidades, por lo tanto una 2-mediana para la gráfica R es el conjunto de vértices $\{B, E\}$.

Y la cobertura quedaría de la siguiente manera:



Ahora, si añadimos otra instalación en alguno de los nodos candidatos que no tiene una, por ejemplo el nodo G . El valor de la suma es 20 unidades, que es menor a la suma anterior.

La nueva cobertura quedaría de la siguiente manera:



3.2. Complejidad computacional

Al tratar con problemas de localización del mundo real, eventualmente se vuelve ineficaz resolver los problemas de forma manual, por eso es inevitable ignorar las ventajas computacionales sobre los algoritmos que los pueden resolver. Pero antes de poder hablar sobre la dificultad computacional de un problema, es necesario entender las instancias que los componen. Dicho esto, una instancia de un problema está dado por un conjunto finito de valores de un campo $E = \{1, \dots, n\}$, un conjunto de soluciones factibles $F \subseteq 2^E$, y una función objetivo $f : 2^E \rightarrow R$. Para el caso de un problema de minimización, buscamos una solución óptima global $S' \in F$ tal que $f(S') \leq f(S), \forall S \in F$. Cada una de estas partes de E , la función objetivo f y el conjunto de soluciones factibles F se definen para cada problema en específico. De manera análoga se define para un problema de maximización con la diferencia de que se invierte la desigualdad [20].

Un problema tiene tres versiones

Antes de continuar con la complejidad computacional, es importante mencionar que un mismo problema puede expresarse de tres formas distintas: como un problema de optimización, como un problema de evaluación, o como un problema decisión.

Problema de optimización: En esta forma del problema, se busca que un algoritmo A , encuentre una solución óptima factible del problema.

Problema de evaluación: En esta forma del problema, se busca que un algoritmo A , encuentre el costo de una solución óptima factible del problema. Si suponemos que A es un algoritmo de tiempo polinomial, este supuesto significaría que el costo de cualquier otra solución se puede calcular de manera eficiente. Por otra parte, el problema de evaluación no puede ser más difícil que la versión de optimización, ya que una vez resuelta la versión de optimización del problema se conoce una solución óptima, con lo cual el algoritmo A puede calcular fácilmente su costo en tiempo polinomial.

Problema de decisión: En esta forma del problema, se busca que un algoritmo A , y un número entero B que representa un límite, respondan a la pregunta: ¿existe una solución factible $S \in F$ tal que $f(S) = B$? La versión de decisión de un problema de optimización, no puede ser más difícil que su versión de evaluación. Debido a que una vez que se obtiene

el costo de una solución óptima como la solución de la versión de evaluación, se puede comparar con el valor de B para dar la respuesta “sí” o “no” de la versión de decisión [20].

Gracias a las observaciones previas, podemos establecer un orden en la dificultad de los problemas, siendo la versión de optimización la de mayor dificultad, seguida de la versión de evaluación, y finalmente la versión de decisión. De esta manera, las tres versiones para el modelo de la P-mediana quedarían de la siguiente manera:

1. Versión de optimización: Dada una gráfica (o red) $G = (V, E)$, donde V es el conjunto de los nodos de la gráfica (o red), y E es el conjunto de las aristas de esta. Se quiere encontrar $V_p \subseteq V$ tal que $|V_p| = p$, y que la suma de las distancias más cortas desde los vértices en $\{V \setminus V_p\}$ hasta su vértice más próximo en V_p se minimice.

2. Versión de evaluación: Dada una gráfica (o red) $G = (V, E)$, donde V es el conjunto de los nodos de la gráfica (o red), y E es el conjunto de las aristas de esta. Se quiere calcular el valor de $V_p \subseteq V$ donde $|V_p| = p$, y la suma de las distancias más cortas desde los vértices en $\{V \setminus V_p\}$ hasta su vértice más próximo en V_p son mínimas.

3. Versión de decisión: Dada una gráfica (o red) $G = (V, E)$, donde V es el conjunto de los nodos de la gráfica (o red), E es el conjunto de las aristas de esta, y B es un número entero no negativo. ¿Existe $V_{p^*} \subseteq V$, tal que $|V_{p^*}| = B$, y que la suma de las distancias más cortas desde los vértices en $\{V \setminus V_{p^*}\}$ hasta su vértice más próximo en V_{p^*} se minimice?

Algoritmos de Tiempo Polinomial

Por lo mencionado anteriormente, un problema de decisión básicamente equivale a responder una pregunta con un “sí” o un “no”. Por esta razón, los problemas de decisión así como las versiones de decisión de un problema de optimización se parecen a los problemas computacionales (relaciones entre conjuntos de instancias y conjuntos de respuestas) estudiados por la teoría de la computación. De acuerdo con lo anterior, se sabe que la versión de decisión de un problema no puede ser más difícil que su versión de optimización, por tal motivo si un problema de decisión no se puede resolver en tiempo polinomial, su versión de optimización tampoco se podrá resolver en tiempo polinomial.

En consecuencia, vamos a decir que un problema de decisión R pertenecerá a la clase P

si existe un algoritmo A que resuelva cualquiera de sus instancias en tiempo polinomial. En pocas palabras, la clase P está formada por aquellos problemas de decisión “fáciles” que se pueden resolver mediante algoritmos de tiempo polinomial.

Una cosa importante por mencionar, es que dado un problema de decisión R , y una instancia J del problema de valor “sí”, un certificado $c(J)$ es una cadena que codifica una solución que permite llegar a la respuesta “sí”. Por otra parte, un certificado es conciso si la longitud de su codificación está limitada por arriba por un polinomio en la cantidad de memoria que se utiliza para codificar la instancia J [20].

Algoritmos No Deterministas de Tiempo Polinomial

Otro tipo de problemas son los que usan algoritmos no deterministas de tiempo polinomial (NP). Diremos que un problema de decisión R , va a pertenecer a la clase NP si existe un algoritmo A que verifica los certificados tal que, para cualquier instancia J de R de valor “sí”, existe un certificado conciso $c(J)$ con la propiedad que el algoritmo A aplicado a la instancia J y el certificado $c(J)$ puede llegar a la respuesta “sí” en tiempo polinomial. Para que un problema pertenezca a la clase NP , basta con que exista un certificado conciso para cualquier instancia de valor “sí” que se pueda verificar su validez en tiempo polinomial [20].

Transformaciones de Tiempo Polinomial

Para comprender esta parte es necesario suponer que tenemos dos problemas de decisión, sean dichos problemas $R1$, y $R2$. Podemos decir que existe una transformación de tiempo polinomial del problema $R1$ al problema $R2$ si cualquier instancia $J2$ de $R2$ se puede construir en tiempo polinomial de cualquier instancia $J1$ de $R1$, de tal modo que $J1$ es una instancia de valor “sí” en el problema de decisión $R1$ si y solo si $J2$ es una instancia de valor “sí” en el problema de decisión $R2$ [20].

Problemas NP-Completo

Diremos que un problema de decisión $R \in NP$ es un problema $NP - Completo$, si cualquier otro problema en NP se puede transformar en tiempo polinomial a él. Este tipo de problemas tienen una propiedad muy importante, ya que si existe un algoritmo de tiempo polinomial para resolver cualquiera de ellos, ese mismo algoritmo servirá para resolver todos los demás problemas en NP . Para probar que un problema de decisión pertenece a la clase $NP - Completo$, basta con probar dos cosas:

1. Probar que está en NP y
2. Demostrar que todos los demás problemas en NP se pueden transformar en tiempo polinomial a él.

Generalmente la segunda parte de la prueba es la más difícil, y por lo regular se demuestra mediante otro problema que ya se demostró que es $NP - Completo$, al cual sea polinomialmente transformable [20].

Problemas NP-Duros

Un último tipo son los problemas $NP - Duros$. Nombraremos a un problema de decisión R como $NP - Duro$ si todos los problemas en NP son polinomialmente transformables a R , pero su pertenencia a NP no se puede establecer. Aunque R es tan difícil como cualquier problema en NP , no clasifica para ser llamado $NP - Completo$. El termino $NP - Duro$ también se utiliza para referirse a problemas de optimización (que no están en NP , porque no son problemas de decisión) cuyas versiones de decisión son problemas $NP - Completos$ [20].

3.3. El problema P-mediana en un árbol: el caso de la 1-mediana

De las propiedades del modelo de la P-mediana, sabemos que es un problema $NP - Duro$ en una gráfica general, sin embargo, este modelo puede ser resuelto en tiempo polinomial

en un árbol [18].

Como ya se había mencionado, en el problema de la P-mediana esencialmente se busca localizar “P” instalaciones de manera que se reduzca la distancia promedio ponderada entre sitios de demanda con las instalaciones. Para ver el funcionamiento de un algoritmo de tiempo lineal en un árbol, trataremos con el caso en donde se quiere localizar una sola instalación. Este tipo de problemas se conocen como el problema de la 1-mediana. Con esto en mente, vamos a usar el algoritmo de tiempo lineal que fue propuesto por Goldman [11] para encontrar la 1-mediana en un árbol. El algoritmo ayuda a explicar el nombre “mediana” del modelo. Ya que, si algún nodo del árbol tiene la mitad o más de la demanda total de todos los nodos en el árbol, entonces es óptimo ubicar la instalación en ese nodo. Al alejarse del nodo, la instalación se alejará de la mitad o más de la demanda y tendrá menos de la mitad de la demanda, lo que aumentará el valor de la función objetivo [18]. Para explicar el algoritmo consideremos los siguientes conjuntos:

$I = \{1, \dots, i, \dots, s\}$ el conjunto de los nodos de demanda

$J = \{1, \dots, j, \dots, r\}$ el conjunto de los sitios candidatos

Ahora supongamos que ningún nodo tiene la mitad o más de la demanda total. Llamaremos nodo extremo a cualquier nodo que solo se conecte con otro nodo en el árbol. Sea h_i la demanda original del nodo $i \in I$, y h'_i la demanda modificada del nodo $i \in I$. Sea $D_{total} = \sum_{i \in I} h_i$

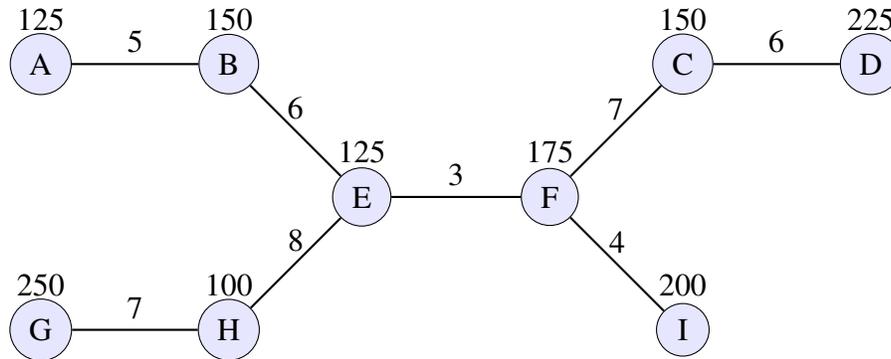
Entonces el algoritmo queda de la siguiente manera:

Paso 1: Sea $h'_i = h_i \quad \forall i \in I$

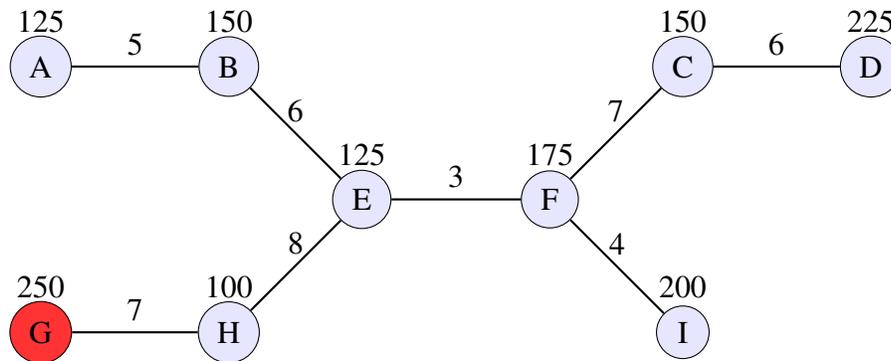
Paso 2: Seleccionar cualquier nodo extremo (preferentemente el que tenga la mayor demanda). Sean X el nodo extremo y Y el nodo al que está conectado. Eliminar el nodo X y la arista (X, Y) . Agregar la demanda modificada del nodo X a la demanda modificada en el nodo Y . Si la nueva demanda modificada del nodo Y es igual o superior a $D_{total}/2$, parar; el nodo Y es la 1-mediana del árbol. De lo contrario, repetir el Paso 2.

Este es un algoritmo de tiempo lineal ya que el Paso 2 se puede realizar en un tiempo constante y cada nodo se examina una vez. La complejidad del Paso 1 también es lineal [18].

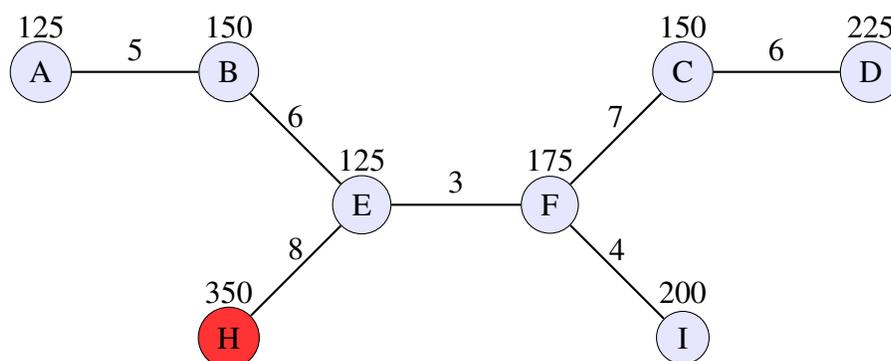
Vamos a ejemplificar el algoritmo con el árbol que se muestra a continuación.



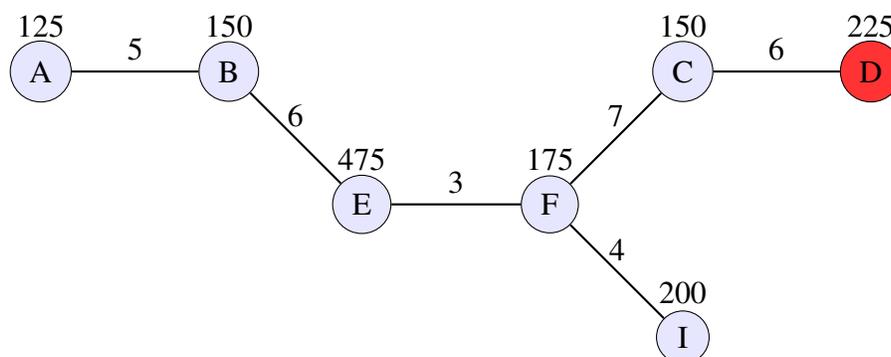
La demanda de cada nodo es la que aparece sobre cada uno de ellos, y las distancias son las que aparecen sobre las aristas de la red. La demanda total para el árbol de nuestro ejemplo es de 1500, así $D_{Total} = 1500$, entonces $D_{Total}/2 = 750$. Los nodos extremos del árbol son los nodos A , D , G , e I . Ya que ningún nodo del árbol tiene la mitad o más de la demanda total, podemos aplicar el algoritmo.



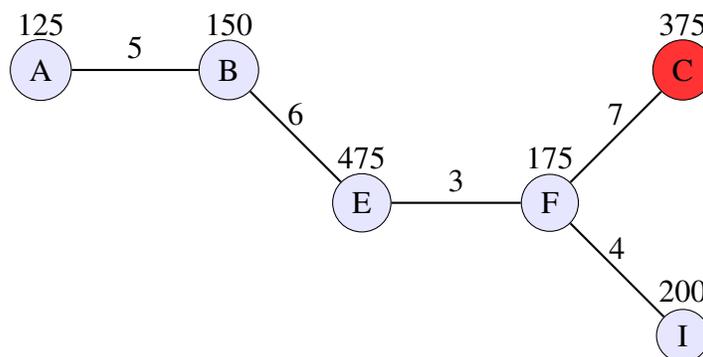
Empezamos con el nodo G (por ser el nodo extremo que tiene la mayor demanda) que es un nodo extremo, así que eliminamos el nodo G y la arista (G, H) , agregamos la demanda de 250 a la demanda del nodo H . Ahora la demanda modificada del nodo H es de 350. Como ningún nodo tiene la mitad o más de la demanda proseguimos con el algoritmo.



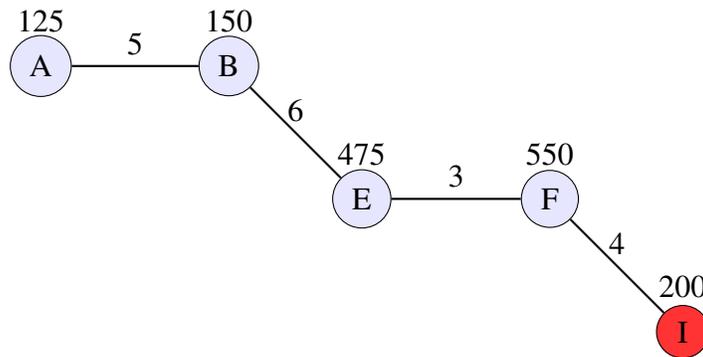
Después continuamos con el nodo extremo H (que es el siguiente nodo extremo de mayor demanda), eliminamos el nodo H y la arista (H, E) , agregamos la demanda de 350 a la demanda del nodo E . Ahora la demanda modificada del nodo E es de 475. Como ningún nodo tiene la mitad o más de la demanda proseguimos con el algoritmo.



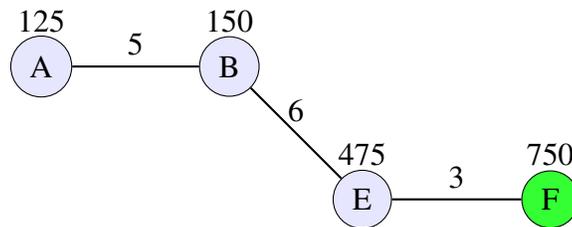
Seguimos con el nodo extremo D , eliminamos el nodo D y la arista (D, C) , agregamos la demanda de 225 al nodo C . Ahora la demanda modificada del nodo C es de 375. Como ningún nodo tiene la mitad o más de la demanda proseguimos con el algoritmo.



Luego seguimos con el nodo extremo C , eliminamos el nodo C y la arista (C, F) , agregamos la demanda de 375 al nodo F . Ahora la demanda modificada del nodo F es de 550. Como ningún nodo tiene la mitad o más de la demanda proseguimos con el algoritmo.



Seguimos con el nuevo nodo extremo I , eliminamos el nodo I y la arista (I, F) , agregamos la demanda de 200 a la demanda del nodo F . Ahora la demanda del nodo F es de 750, la cual es la mitad de la demanda del árbol, entonces paramos. Por lo tanto, el nodo F es la 1-mediana del árbol.



Por lo tanto es óptimo localizar la instalación en el nodo F . Y el valor de la función objetivo de la 1-mediana en el nodo F es de 13850.

Observemos que para la localización de la 1-mediana del árbol, no se usan las distancias entre los nodos [18].

3.4. Modelo del problema de localización de la planta capacitada

Una variante del modelo tradicional de la P-mediana es el modelo del problema de localización de la planta capacitada. Esta variante es de mucha importancia para una gran cantidad de organizaciones tanto públicas como privadas, las cuales tienen por objetivo determinar un conjunto de instalaciones capacitadas (como almacenes, centros de votación etc.) de tal forma que se minimicen los costos por la construcción de las instalaciones así como los costos de transporte. Una diferencia importante con respecto a otros modelos es que en este modelo si se permiten múltiples instalaciones ubicadas en el mismo sitio [15]. Esta variante del modelo de la P-mediana es la que se va a adaptar y usar para poder realizar el caso de estudio de este trabajo. Debido a que para poder utilizar la descomposición de Benders, es indispensable contar con una variable continua en el subproblema, por tal motivo, se consideró a la variable $Y_{i,j}$ como la proporción de la demanda del nodo i atendida por la instalación j . Logrando así hacerla continua.

Buscamos formular al modelo del problema de localización de la planta capacitada como un problema de programación lineal entero mixto. Como se mencionó anteriormente, para este modelo queremos minimizar la distancia que existe entre los nodos de demanda con su instalación más cercana, además de minimizar los costos por localizar las instalaciones. Así, para cada nodo de los sitios candidatos para localizar las instalaciones, se va a considerar una variable binaria, la cual tomará el valor 1 si el nodo es un sitio candidato seleccionado, y tomará el valor de 0 en caso contrario. Igualmente que en el modelo de la P-mediana, se tendrá un límite en el número de instalaciones que se pueden instalar digamos " P ".

Después, la distancia que exista entre cada uno de los nodos candidatos con los nodos de demanda se representara con una arista. Asociada a cada una de las aristas, se considerará una variable. Esta es la variable que se relajó, haciendo que su valor pueda caer en el intervalo $[0, 1]$, y se relaciona con la parte de la demanda que una instalación va a satisfacer. Además de que únicamente será posible asignar una demanda a instalaciones que están establecidas, siempre asegurándonos de que la demanda total sea satisfecha. Y la demanda tiene que ser menor que la capacidad de las instalaciones para evitar problemas. Este modelo lo podemos formular de la siguiente manera:

$$\text{Minimizar} \quad \sum_{j \in J} f_j X_j + \sum_{i \in I} \sum_{j \in J} c_{ij} Y_{ij} \quad (3.1)$$

$$\text{Sujeto a :} \quad \sum_{j \in J} Y_{ij} = 1 \quad i \in I \quad (3.2)$$

$$\sum_{j \in J} X_j = P \quad (3.3)$$

$$\sum_{i \in I} d_i Y_{ij} \leq q_j X_j \quad j \in J \quad (3.4)$$

$$X_j \in \{0, 1\} \quad j \in J \quad (3.5)$$

$$Y_{ij} \in [0, 1] \quad i \in I, \quad j \in J \quad (3.6)$$

Donde definimos la siguiente notación:

Conjuntos

$I = \{1, \dots, i, \dots, s\}$ Conjunto de los nodos de demanda

$J = \{1, \dots, j, \dots, r\}$ Conjunto de los sitios candidatos

Parámetros

d_i = Demanda del nodo i

q_j = Capacidad de la instalación j

c_{ij} = Costo de atender parte de la demanda del nodo i por la instalación j

f_j = Costo fijo asociado con la instalación abierta j

P = Número deseado de instalaciones por abrir

Variables

Y_{ij} = Variable continua de decisión, la cual corresponde al porcentaje de la demanda del nodo i atendida por la instalación j

$$X_j = \begin{cases} 1 & \text{si la instalación ubicada en el nodo } j \text{ está abierta} \\ 0 & \text{de lo contrario} \end{cases}$$

[15].

La función objetivo 3.1 expresa la minimización del costo total de la instalación. La restricción 3.2 asegura que la demanda de cada nodo de demanda sea satisfecha. La restricción 3.3 establece que el número de instalaciones abiertas sean P . La restricción 3.4 establece la conexión entre las variables Y_{ij} y X_j : estas variables establecen que ningún nodo de demanda puede ser abastecido desde una instalación cerrada y la demanda total suministrada desde cada instalación abierta no excede la capacidad de la instalación. La restricción 3.6 proporciona la relajación de la variable binaria Y_{ij} . Finalmente la restricción 3.5 especifica que la variable de decisión es binaria.

Capítulo 4

Métodos de Solución para Resolver el Problema de Localización P-mediana

Una vez que hemos conocido y definido el modelo de la P-mediana como un problema de programación entero binario ($0 - 1$), lo vamos a querer resolver, para ello podemos hacer uso de métodos exactos y heurísticos.

Los métodos exactos para resolver problemas son aquellos que garantizan producir, en un tiempo finito, un óptimo global para el problema. O en caso contrario demostrar que no existe una solución factible. Mientras que los métodos heurísticos para resolver problemas son aquellos que proporcionan soluciones factibles, que no necesariamente son óptimas. Los métodos heurísticos se ejecutan más rápidos que los exactos. Por tanto, los métodos heurísticos son capaces de manejar instancias de problemas más grandes que los métodos exactos [20].

4.1. Métodos Exactos

Los métodos exactos más notorios que pueden usarse para resolver problemas de programación entera (como el problema P-mediana) son: el método de planos de corte, y el método de ramificación y acotamiento.

4.1.1. Método de los planos de corte

El método de los planos de corte consiste en ir agregando restricciones las cuales reducen la región factible del problema relajado de programación lineal, sin eliminar las soluciones factibles del problema original. Este método inicia a partir de una solución factible para posteriormente ir agregando los cortes a las restricciones del problema.

La forma para encontrar los planos de corte en problemas de programación entera binaria es la siguiente:

1. Considerar cualquier restricción de la forma “ \leq ” donde los coeficientes sean no negativos.
2. Encontrar un grupo de variables tales que:
 - a) la restricción se viole si todas las variables del grupo son iguales a 1 y todas las demás variables sean 0
 - b) pero la restricción se satisface si el valor de cualquiera de las variables cambia de 1 a 0.
3. Si N es el número de las variables del grupo, entonces el plano de corte tiene la forma: Suma de variables del grupo $\leq N - 1$ y agregar la nueva restricción (corte) al problema original [14].

Para ver cómo funciona el método vamos a considerar el siguiente ejemplo.

$$\text{Maximizar} \quad z = 9x_1 + 5x_2 + 6x_3 + 4x_4 \quad (4.1)$$

$$\text{Sujeto a :} \quad 6x_1 + 3x_2 + 5x_3 + 2x_4 \leq 10 \quad (4.2)$$

$$x_3 + x_4 \leq 1 \quad (4.3)$$

$$-x_1 + x_3 \leq 0 \quad (4.4)$$

$$-x_2 + x_4 \leq 0 \quad (4.5)$$

$$x_i \in \{0, 1\} \quad \text{con } i = \{1, 2, 3, 4\} \quad (4.6)$$

Comenzamos con la solución factible de la relajación del problema $(\frac{5}{6}, 1, 0, 1)$ (la relajación de un problema consiste en eliminar restricciones que dificultan resolver el problema. Para este caso se elimina la restricción de las variables binarias, haciéndolas continuas). Después aplicamos el procedimiento a la primera restricción, y notamos que si las variables toman el valor de $(1, 1, 0, 1)$. La restricción se viola, por lo tanto seleccionamos el

siguiente grupo de variables $\{x_1, x_2, x_4\}$, ya que si cambiamos el valor de alguna de estas variables la restricción no se viola. Como la restricción que seleccionamos tenía cuatro variables, el plano de corte resultante tendrá tres variables.

Entonces el plano de corte queda de la siguiente manera: $6x_1 + 3x_2 + 2x_4 \leq 10$. Ahora por las restricciones binarias, podemos simplificar el plano de corte como: $x_1 + x_2 + x_4 \leq 2$. Y lo agregamos al problema.

$$\text{Maximizar} \quad z = 9x_1 + 5x_2 + 6x_3 + 4x_4 \quad (4.7)$$

$$\text{Sujeto a :} \quad 6x_1 + 3x_2 + 5x_3 + 2x_4 \leq 10 \quad (4.8)$$

$$x_3 + x_4 \leq 1 \quad (4.9)$$

$$-x_1 + x_3 \leq 0 \quad (4.10)$$

$$-x_2 + x_4 \leq 0 \quad (4.11)$$

$$x_1 + x_2 + x_4 \leq 2 \quad (4.12)$$

$$x_i \in \{0, 1\} \quad \text{con } i = \{1, 2, 3, 4\} \quad (4.13)$$

Con este plano de corte, se elimina parte de la región factible del relajamiento, pero no se eliminan soluciones enteras factibles. Resolviendo el problema con el nuevo plano de corte, se obtiene la siguiente solución: $x_1 = 1$, $x_2 = 1$, $x_3 = 0$, y $x_4 = 0$ con un valor en la función objetivo de $z = 14$.

4.1.2. Ramificación y acotamiento

Este método consiste en ir ramificando (dividiendo) el problema en subproblemas más pequeños que son más fáciles de resolver. La ramificación se hace al partir el conjunto de las soluciones factibles en subconjuntos más pequeños. Para después llevar a cabo el sondeo mediante el acotamiento de la mejor solución del subconjunto, y así, ir descartando a los subconjuntos cuyas cotas no tengan una solución óptima para el problema original [14]. Vamos a mostrar la versión que se utiliza para resolver problemas de programación entera binaria.

Para este caso, la manera más fácil de “ramificar” el problema consiste en fijar el valor de una de las variables del problema. Por el tipo de problema que se busca resolver, se van a

fijar los valores de las variables en 0 y 1. Así por ejemplo, para la variable x_1 , al fijar su valor tendríamos: $x_1 = 0$ para un subproblema y $x_1 = 1$ para otro subproblema. De esta manera el problema original se empieza a ramificar para ir construyendo un árbol en el que sus ramas irán creciendo en cada nueva iteración. Dicho árbol se conoce como “árbol de soluciones” y la variable a la que fijamos su valor se llama “variable de ramificación”. Afín de evitar problemas al momento de fijar el valor de las variables, se considerará el orden natural, es decir primero se fijará el valor de la variable x_1 , luego el valor de x_2 , después el valor de x_3, x_4 , etc.

Por otra lado, el “acotamiento”, es la obtención de una cota para cada subproblema con el fin de mostrar la precisión de la mejor solución factible. La forma de realizarlo es mediante la relajación de la solución del subproblema. La relajación de un problema la obtenemos al eliminar un conjunto de restricciones que dificulten resolver los problemas. En los problemas de programación entera binaria, las restricciones que generalmente se eliminan son las que requieren que las variables sean enteras, es decir que su valor sea 0 o 1. Al hacer la relajación, el valor de las variables cae dentro del intervalo $[0, 1]$.

Para cada nuevo subproblema que se vaya generando, se debe resolver su relajación para obtener una solución óptima que contenga el valor de Z . Si el valor de la solución no es entera, se debe redondear a un entero. (Los valores se pueden redondear debido a que todos los coeficientes de la función objetivo son enteros, y por tal motivo deben de dar un valor entero de Z). Este valor Z es la cota del subproblema.

Ahora el “sondeo” de un subproblema se realiza cuando se obtiene la solución del subproblema relajado y el valor de Z . Entonces el subproblema se sondea (elimina) para ya no tomarse en cuenta cuando su solución cumpla una de las tres siguientes condiciones:

1. Su cota es $\leq Z^*$.
2. Su relajamiento no tiene soluciones factibles.
3. La solución óptima de su relajamiento es entera. Si la nueva solución es entera, y es mejor que la solución de “apoyo” actual, entonces debe guardarse como la nueva solución de “apoyo” junto con su valor de Z . Donde denotaremos como Z^* al valor de Z de la solución de apoyo. Posteriormente, se deben aplicar las tres condición anteriores a todos los problemas que aún no hayan sido sondeados, pero con la nueva solución de apoyo Z^* [14].

Algunos autores como: [23], [16], [19], y [22], plantean diferentes estrategias del método de ramificación y acotamiento para resolver el problema de la P-mediana.

Para dar una idea de cómo funciona el método, vamos a considerar el siguiente ejemplo:

$$\text{Maximizar} \quad z = x_1 + x_2 + x_3 + x_4 \quad (4.14)$$

$$\text{Sujeto a :} \quad x_1 + x_2 + x_3 \leq 1 \quad (4.15)$$

$$x_1 + x_2 + x_4 \geq 1 \quad (4.16)$$

$$x_2 + x_3 + x_4 \leq 1 \quad (4.17)$$

$$x_i \in \{0, 1\} \quad \text{con } i = \{1, 2, 3, 4\} \quad (4.18)$$

Entonces, resolvemos el problema relajado, es decir omitimos las restricciones binarias y las volvemos continuas haciendo que el valor de las variables caigan en el intervalo $[0, 1]$, con lo cual el resultado queda de la siguiente manera: $z = 1,333333$, $x_1 = 0,333333$, $x_2 = 0,333333$, $x_3 = 0,333333$, y $x_4 = 0,333333$.

Con esta solución inicial comenzamos a ramificar el problema. Así para el primer subproblema vamos a fijar el valor de la variable x_1 en 0, entonces el subproblema queda como:

$$\text{Maximizar} \quad z = x_2 + x_3 + x_4 \quad (4.19)$$

$$\text{Sujeto a :} \quad x_2 + x_3 \leq 1 \quad (4.20)$$

$$x_2 + x_4 \geq 1 \quad (4.21)$$

$$x_2 + x_3 + x_4 \leq 1 \quad (4.22)$$

$$x_i \in \{0, 1\} \quad \text{con } i = \{1, 2, 3, 4\} \quad (4.23)$$

Igual que en el problema anterior resolvemos el subproblema relajado, con lo cual tenemos la siguiente solución: $z = 1$, $x_1 = 0$, $x_2 = 0,5$, $x_3 = 0$, y $x_4 = 0,5$. Al aplicar las condiciones de sondeo anteriores, notamos que el valor obtenido de Z (la cota) es entero pero la solución no es entera. Por lo tanto aún no se ha encontrado una solución de apoyo. Al no cumplir con alguna de las condiciones de sondeo, el subproblema no puede ser descartado, así que queda pendiente.

Continuamos con el proceso, ahora fijamos el valor de la variable x_1 en 1, para generar el

segundo subproblema, este queda como:

$$\text{Maximizar} \quad z = 1 + x_2 + x_3 + x_4 \quad (4.24)$$

$$\text{Sujeto a :} \quad 1 + x_2 + x_3 \leq 1 \quad (4.25)$$

$$1 + x_2 + x_4 \geq 1 \quad (4.26)$$

$$x_2 + x_3 + x_4 \leq 1 \quad (4.27)$$

$$x_i \in \{0, 1\} \quad \text{con } i = \{1, 2, 3, 4\} \quad (4.28)$$

Al resolver el subproblema relajado nos da una solución de: $z = 2$, $x_1 = 1$, $x_2 = 0$, $x_3 = 0$, y $x_4 = 1$. Al aplicar las condiciones de sondeo antes mencionadas, podemos determinar que es una solución entera binaria, es decir cumple con la tercera condición. Por lo tanto tenemos la primera solución de apoyo $Z^* = 2$. De esta manera el segundo subproblema quedaría sondeado.

Debido a que el primer subproblema no fue sondeado, se tiene que continuar en él el proceso de ramificación.

Entonces para generar el tercer subproblema vamos a fijar el valor de la variable x_2 en 0, así el subproblema queda como:

$$\text{Maximizar} \quad z = x_3 + x_4 \quad (4.29)$$

$$\text{Sujeto a :} \quad x_3 \leq 1 \quad (4.30)$$

$$x_4 \geq 1 \quad (4.31)$$

$$x_3 + x_4 \leq 1 \quad (4.32)$$

$$x_i \in \{0, 1\} \quad \text{con } i = \{1, 2, 3, 4\} \quad (4.33)$$

Al igual que en los problemas anteriores resolvemos el subproblema relajado, con lo cual tenemos la siguiente solución: $z = 1$, $x_1 = 0$, $x_2 = 0$, $x_3 = 0$, y $x_4 = 1$. Aplicando las condiciones de sondeo, notamos que el valor obtenido de Z (la cota) es entero. Pero el valor de la cota es menor que el valor de la solución de apoyo actual, en otras palabras se cumple la primera condición. Por lo tanto el tercer subproblema ha sido sondeado.

Para el cuarto subproblema vamos a fijar el valor de la variable x_2 en 1, entonces el sub-

problema queda como:

$$\text{Maximizar} \quad z = 1 + x_3 + x_4 \quad (4.34)$$

$$\text{Sujeto a :} \quad 1 + x_3 \leq 1 \quad (4.35)$$

$$1 + x_4 \geq 1 \quad (4.36)$$

$$1 + x_3 + x_4 \leq 1 \quad (4.37)$$

$$x_i \in \{0, 1\} \quad \text{con } i = \{1, 2, 3, 4\} \quad (4.38)$$

Como con los subproblemas previos resolvemos el subproblema relajado, lo que nos da la siguiente solución: $z = 1$, $x_1 = 0$, $x_2 = 1$, $x_3 = 0$, y $x_4 = 0$. Utilizando las condiciones de sondeo, se puede observar que el valor obtenido de Z (la cota) es menor que el valor de la solución de apoyo actual, es decir se satisface la primera condición, por lo cual el cuarto subproblema queda sondeado.

Al no quedar más subproblemas por ramificar y tener todas las variables valor 0 o 1, podemos decir que hemos terminado de resolver el problema y encontrado la solución: $x_1 = 1$, $x_2 = 0$, $x_3 = 0$, y $x_4 = 1$ con un valor en la función objetivo de $z = 2$.

4.2. Métodos Heurísticos

A continuación, vamos a mencionar algunos de los métodos heurísticos más comunes que existen para resolver el problema de la P-mediana. Estos métodos heurísticos los podemos dividir en dos grandes clases, que son algoritmos de construcción y algoritmos de mejora.

Nombraremos tres de los algoritmos heurísticos más populares: miope, intercambio y búsqueda de vecindario. Donde el primer heurístico presentado es un algoritmo de construcción y los otros dos son algoritmos de mejora [7].

4.2.1. Miope

Como ya se mencionó, el modelo de la P-mediana consiste en ubicar “P” instalaciones de tal manera que se minimice la distancia promedio ponderada, entonces si queremos localizar una sola instalación, por lo que se indicó en el capítulo 3, sabemos que una solución óptima consiste en ubicar las instalaciones en los nodos, para posteriormente utilizar el algoritmo de la 1-mediana y localizar la instalación en la ubicación óptima.

Con la idea de localizar las $P - 1$ instalaciones restantes, consideremos a X_{P-1} como el conjunto de las posibles ubicaciones para las $P - 1$ instalaciones, y sea $d(i, X_{P-1})$ la distancia más corta entre un nodo de demanda $i \in I$ con el nodo de la instalación más próxima en el conjunto X_{P-1} , proporcionada por la ubicación del sitio candidato $j \in J$.

De esta forma el mejor lugar para asignar una nueva instalación, dado que las primeras $P - 1$ instalaciones están localizadas en el conjunto X_{P-1} será la ubicación $j \in J$ tal que minimice $Z_j = \sum_{i \in I} h_i d(i, j \cup X_{P-1})$ [7].

Con esta idea en mente podemos formular el algoritmo Miope como sigue:

1. Inicialización $k = 0$, donde k contará el número de instalaciones localizadas, y $X_k = \phi$, será el conjunto que dará las ubicaciones de las k instalaciones que se han localizado en cada etapa del algoritmo.
2. Incrementar el contador k sobre el número de instalaciones ubicadas.
3. Calcular $Z_j^k = \sum_{i \in I} h_i d(i, j \cup X_{k-1})$ para cada nodo $j \in J$ que no esté en el conjunto X_{k-1} . Considerando que Z_j^k da el valor de la función objetivo de la P-mediana si localizamos la k -ésima instalación en el nodo $j \in J$, ya que las primeras $k - 1$ instalaciones están en las ubicaciones dadas en el conjunto X_{k-1} y el nuevo nodo $j \in J$ no pertenece al conjunto.
4. Encontrar el nodo $j^*(k)$ que minimice Z_j^k , es decir, $j^*(k) = \operatorname{argmin}_{j \in J} \{Z_j^k\}$. Donde $j^*(k)$ proporciona la mejor ubicación para la instalación k -ésima, dadas las primeras $k - 1$ instalaciones. Agregar el nodo $j^*(k)$ al conjunto X_{k-1} para obtener el conjunto X_k el cual es igual a $X_k = \{j^*(k) \cup X_{k-1}\}$
5. Si $k = p$, parar, el conjunto X_p es la solución al algoritmo miope. Si $k < p$, volver al paso 2 [7]

4.2.2. Algoritmo de búsqueda en vecindarios

En los problemas de localización de instalaciones se busca asignar la demanda de los nodos a su instalación más próxima. Debido a esto se van creando conjuntos de nodos de demanda los cuales han sido asignados a las mismas instalaciones, estos nodos forman una “vecindad”. Así, el algoritmo de búsqueda de vecindario, inicia con una localización

factible.

Sin importar si la localización es óptima o no (también podríamos usar el algoritmo miope para dar una localización inicial) y posteriormente aplicar el algoritmo de la 1-mediana en las vecindades, para localizar la instalación en el óptimo, formando nuevos vecindarios. Si los vecindarios se modifican, se aplica de nuevo el algoritmo de la 1-mediana dentro de cada nuevo vecindario y así sucesivamente hasta que las ubicaciones proporcionadas por el algoritmo de la 1-mediana no cambien, entonces detenemos el algoritmo de búsqueda de vecindario ya que tendríamos una ubicación “óptima” para las instalaciones. Una de las desventajas de este algoritmo es que solo evalúa el efecto de la reubicación de los nodos en el vecindario, es decir, la mejora potencial de nodos que se encuentren fuera del vecindario que pueden ser reubicados no se considera [7].

4.2.3. Intercambio

Este es un algoritmo heurístico que al igual que el heurístico anterior, parte de una solución inicial de la localización de las “ P ” instalaciones. Esta solución la podemos obtener a partir del uso del algoritmo miope, aunque también puede ser obtenida por algún otro método heurístico, o bien simplemente puede ser elegida al azar.

En caso de optar por una solución inicial al azar, hay que probar varias soluciones al azar para considerar la que nos dé un mejor resultado. En consecuencia, el algoritmo de intercambio comienza con una solución inicial (la localización de las “ P ” instalaciones), para posteriormente seleccionar la primera instalación que será intercambiada. Después identificar el mejor nodo de reemplazo para el sitio de la instalación que se considera para ser intercambiada. Entonces si el intercambio reduce la distancia promedio ponderada, cambiamos el sitio actual por el sitio de reemplazo. Y si no, verificamos todos los sitios candidatos a ser intercambiados. En caso de haber verificado todos los sitios (y no existir cambios) paramos y esa es la mejor solución para localizar las instalaciones. En otro caso seleccionar la próxima instalación a la cual queremos intercambiar [7].

Capítulo 5

Solución al Problema de Localización P-mediana Usando Descomposición de Benders

Con los métodos exactos y heurísticos que anteriormente se mencionaron, se podría resolver el problema de la P-mediana, pero para este trabajo se va a emplear un método heurístico no tan común para solucionarlo, el cual se trata de la descomposición de Benders. Ya que con este método se pueden aprovechar las estructuras especiales de algunos problemas de programación matemática. Como la estructura que tiene el problema de localización de la planta capacitada que es la variante del problema P-mediana con la que se va a trabajar.

5.1. Descomposición de Benders

Dentro de los problemas de localización, hay ocasiones en que los problemas se vuelven difíciles de resolver, debido a su gran tamaño o al tipo de variables que se manejan. Por tanto, es necesario utilizar otras técnicas para poder resolverlos. Algunas de esas, son las técnicas de descomposición. Estas técnicas se utilizan en problemas donde es posible identificar la parte “fácil” y “difícil” de ellos, para así poderlos resolver de forma iterativa.

Entre estos tipos de técnicas, se encuentra la descomposición de Benders [2], que nor-

malmente se utiliza para resolver problemas de programación lineal entera mixta, y como su nombre lo indica “descompone” al problema de programación lineal en dos partes, un problema maestro y un subproblema. Donde el problema maestro resuelve las variables complicadas, y el subproblema toma la solución del maestro para poder resolver. Para los problemas de programación entera mixta la parte “difícil” del problema son las variables enteras.

5.2. Principio de Descomposición de Benders

Consideremos el siguiente problema:

$$\text{Minimizar}_{x,y} \quad f = c^t x + d^t y \quad (5.1)$$

$$\text{Sujeto a :} \quad Ax + By \geq b \quad (5.2)$$

$$x \geq 0 \quad y \in Y \quad (5.3)$$

En donde la variable $y \in Y$ se puede considerar como la variable “complicada”, debido al conjunto que lo define, por ejemplo \mathbb{Z} o $\{0, 1\}$. Entonces el problema de minimización puede descomponerse como:

$$\text{Minimizar}_y \quad f = d^t y + q(x) \quad (5.4)$$

$$y \in Y \quad (5.5)$$

como problema maestro, y

$$q(x) = \quad \text{Min} \quad c^t x \quad (5.6)$$

$$\text{Sujeto a :} \quad Ax \geq b - B\bar{y} \quad (5.7)$$

$$x \geq 0 \quad (5.8)$$

que corresponde al subproblema. Donde \bar{y} corresponde a el valor fijo de y obtenido de la solución del problema maestro [3].

Para la descomposición de Benders se tiene una reformulación del problema maestro, que llamaremos problema maestro relajado. Esta reformulación del problema maestro se logra utilizando las variables duales asociadas a las restricciones del subproblema y la variable auxiliar θ que será la conexión entre ambos problemas (maestro y subproblema).

Así, el problema maestro relajado se puede escribir como sigue:

$$\text{Minimizar}_y \quad d^t y + \theta \quad (5.9)$$

$$y \in Y, \quad \theta \in \mathbb{R} \quad (5.10)$$

El problema maestro propone una solución para el conjunto de variables complicadas, que después se fijara en un subproblema, el cual posteriormente se pasara a su forma dual:

$$\text{Maximizar} \quad w(b - B\bar{y})^t \quad (5.11)$$

$$\text{Sujeto a :} \quad wA^t \leq c \quad (5.12)$$

$$w \geq 0 \quad (5.13)$$

Para así ir generando los cortes de Benders. Si w^* es el punto óptimo, entonces el corte de Benders es:

$$\bar{w}_k(b - By)^t \leq \theta \quad (5.14)$$

donde \bar{w} corresponde a el valor fijo en w^* .

Por otro lado, si el subproblema dual no es acotado, entonces se debe generar un corte de factibilidad. Pero para poder encontrarlo, primero hay que limitar los rayos no acotados del subproblema dual agregando la restricción $w(b - B\bar{y})^t \leq M$. Donde la $M \geq 0$ tiene que ser lo suficientemente grande para limitar a los rayos no acotados.

Al resolver el subproblema con la nueva restricción, se obtiene w^* , con la que se genera el corte de factibilidad:

$$\bar{w}_l(b - By)^t \leq 0 \quad (5.15)$$

Que elimina la propuesta anterior del problema maestro.

Entonces al agregar estos nuevos cortes al problema maestro relajado, este se reformula como sigue:

$$\text{Minimizar}_y \quad d^t y + \theta \quad (5.16)$$

$$\text{Sujeto a :} \quad \bar{w}_k(b - By)^t \leq \theta \quad k = 1 \dots K \quad (5.17)$$

$$\bar{w}_l(b - By)^t \leq 0 \quad l = 1 \dots L \quad (5.18)$$

$$y \in Y, \quad \theta \in \mathbb{R} \quad (5.19)$$

En donde las primeras “ K ” restricciones se llaman cortes de Benders (o de optimalidad), y las siguientes “ L ” restricciones se llaman cortes de factibilidad. Con los cortes de Benders se va formando una aproximación exterior de la función objetivo correspondiente al subproblema.

Cada iteración va dando un valor para la función objetivo del problema maestro $d^t y_i + \theta$ la cual es una cota inferior \underline{z} para el problema completo. Mientras que una cota superior \bar{z} viene dada en cada iteración por $c^t x_i + d^t y_i$, donde i es la iteración actual. El criterio de optimalidad se obtendrá al momento en que la cota superior e inferior coincidan salvo una tolerancia ϵ [3].

5.3. Algoritmo de Descomposición de Benders

A continuación vamos a dar una propuesta de algoritmo para la descomposición de Benders.

1. Inicialización, hacemos $\bar{z} = \infty$, $\underline{z} = -\infty$, $Tol = \epsilon$, $i = 1$.
2. Fijar el valor de y en una solución factible.
3. Resolver el subproblema dual con la variable y fija, siendo $\bar{y} = y^*$:

$$\text{Maximizar} \quad w(b - B\bar{y})^t \quad (5.20)$$

$$\text{Sujeto a :} \quad wA^t \leq c \quad (5.21)$$

$$w \geq 0 \quad (5.22)$$

a) Si el subproblema alcanza el punto óptimo w^* , construir el corte de optimalidad:

$$\theta \geq \bar{w}_k(b - By)^t \quad (5.23)$$

y agregarlo al problema maestro relajado.

b) Si el subproblema dual es no acotado, agregar el corte

$$w(b - B\bar{y})^t \leq M \quad (5.24)$$

al subproblema dual y resolver. Siendo $M \geq 0$ suficientemente grande para limitar a los rayos no acotados. Después construir el corte de factibilidad.

$$0 \geq \bar{w}_l(b - By)^t \quad (5.25)$$

y agregarlo al problema maestro relajado.

4. Resolver el problema maestro relajado

$$\text{Minimizar}_y \quad d^t y + \theta \quad (5.26)$$

$$\text{Sujeto a :} \quad \bar{w}_k(b - By)^t \leq \theta \quad k = 1 \dots K \quad (5.27)$$

$$\bar{w}_l(b - By)^t \leq 0 \quad l = 1 \dots L \quad (5.28)$$

$$y \in Y, \quad \theta \in \mathbb{R} \quad (5.29)$$

Hacer $\underline{z} = d^t y_i + \theta$ y $\bar{z} = c^t x_i + d^t y_i$ (donde i es la iteración actual) e ir al paso 5.

5. Si $\bar{z} - \underline{z} \leq \epsilon$, terminamos. La solución es $d^t y_i + \theta$, en otro caso hacer $i = i + 1$ y regresar al paso 3.

5.4. Solución al Problema de Localización P-mediana Usando Descomposición de Benders

En el capítulo 3 se habló de una variante para el modelo de la P-mediana. Esta variante es el modelo del problema de localización de la planta capacitada que es el modelo que se va a utilizar en este trabajo.

$$\text{Minimizar} \quad \sum_{j \in J} f_j X_j + \sum_{i \in I} \sum_{j \in J} c_{ij} Y_{ij} \quad (5.30)$$

$$\text{Sujeto a :} \quad \sum_{j \in J} Y_{ij} = 1 \quad i \in I \quad (5.31)$$

$$\sum_{j \in J} X_j = P \quad (5.32)$$

$$\sum_{i \in I} d_i Y_{ij} \leq q_j X_j \quad j \in J \quad (5.33)$$

$$X_j \in \{0, 1\} \quad j \in J \quad (5.34)$$

$$Y_{ij} \in [0, 1] \quad i \in I, \quad j \in J \quad (5.35)$$

Donde definimos la siguiente notación:

Conjuntos

$I = \{1, \dots, i, \dots, s\}$ Conjunto de los nodos de demanda

$J = \{1, \dots, j, \dots, r\}$ Conjunto de los nodos candidatos

Parámetros

d_i = Demanda del nodo i

q_j = Capacidad de la instalación j

c_{ij} = Costo por atender toda la demanda del nodo i en la instalación j

f_j = Costo fijo asociado con la instalación abierta j

P = Número deseado de instalaciones por abrir

Variabes

Y_{ij} = Variable continua de decisión, la cual corresponde al porcentaje de la demanda del nodo i atendida por la instalación j

$$X_j = \begin{cases} 1 & \text{si la instalación ubicada en el nodo } j \text{ está abierta} \\ 0 & \text{de lo contrario} \end{cases}$$

Este modelo cumple con las características necesarias para poder aplicar la descomposición de Benders, ya que contamos con las variables “fáciles” que son las Y_{ij} que se han conseguido al relajarlas para volverlas continuas, haciendo que su valor pueda caer en el intervalo $[0, 1]$, y las variables complicadas que son las X_j las cuales son binarias.

Entonces el problema maestro relajado puede formularse como:

$$\text{Minimizar} \quad \sum_{j \in J} f_j X_j + \theta \quad (5.36)$$

$$\text{Sujeto a :} \quad \sum_{j \in J} X_j = P \quad (5.37)$$

$$X_j \in \{0, 1\} \quad j \in J, \quad \theta \in \mathbb{R} \quad (5.38)$$

y el subproblema:

$$\text{Minimizar} \quad \sum_{i \in I} \sum_{j \in J} c_{ij} Y_{ij} \quad (5.39)$$

$$\text{Sujeto a :} \quad \sum_{j \in J} Y_{ij} = 1 \quad i \in I \quad (5.40)$$

$$\sum_{i \in I} d_i Y_{ij} \leq q_j \bar{X}_j \quad j \in J \quad (5.41)$$

$$Y_{ij} \in [0, 1] \quad i \in I, \quad j \in J \quad (5.42)$$

el subproblema dual es:

$$\text{Maximizar} \quad \sum_{i \in I} w_{1i} + \sum_{j \in J} q_j w_{2j} \quad (5.43)$$

$$\text{Sujeto a :} \quad w_{1i} + d_i w_{2j} \leq c_{ij} \quad i \in I, j \in J \quad (5.44)$$

$$w_{2j} \leq 0 \quad j \in J \quad (5.45)$$

De esta manera tenemos una alternativa para poder resolver el problema de localización P-mediana mediante la descomposición de Benders.

Vamos a ilustrar el algoritmo de la descomposición de Benders, para esto consideremos el siguiente ejemplo.

Lo primero que vamos a hacer, es dar los datos que tendrá el ejemplo. Donde $I = \{1, 2\}$ es el conjunto de los sitios de demanda, y $J = \{1, 2\}$ es el conjunto de los sitios candidatos.

El primer dato es la distancia que existe entre los sitios de demanda con los sitios candidatos para localizar las instalaciones.

-	1	2
1	8	5
2	9	4

Tabla 5.1: Tabla de las distancias entre los sitios candidatos con los sitios de demanda

Seguidamente del costo por localizar las instalaciones en cada uno de los sitios candidatos.

Sitio	Costo
1	22
2	14

Tabla 5.2: Costo de cada instalación de los sitios candidatos

El siguiente dato, es la capacidad con la que contara cada uno de los sitios candidatos.

Sitio	Capacidad
1	25

2	20
---	----

Tabla 5.3: Capacidad de cada sitio candidato

Y el último dato, es la demanda que tienen por satisfacer las instalaciones.

Sitio	Demanda
1	12
2	8

Tabla 5.4: Demanda de los sitios

Después de tener los datos para el ejemplo, el siguiente paso es introducirlos en el modelo del problema de localización de la planta capacitada descrito en el capítulo 3 para posteriormente aplicar el algoritmo de la descomposición de Benders descrito en el capítulo 5.

Así el problema anterior quedaría como:

$$\text{Minimizar} \quad z = 22x_1 + 14x_2 + 8y_{11} + 5y_{12} + 9y_{21} + 4y_{22} \quad (5.46)$$

$$\text{Sujeto a :} \quad y_{11} + y_{12} = 1 \quad (5.47)$$

$$y_{21} + y_{22} = 1 \quad (5.48)$$

$$x_1 + x_2 = 2 \quad (5.49)$$

$$12y_{11} + 12y_{12} \leq 25x_1 \quad (5.50)$$

$$8y_{21} + 8y_{22} \leq 20x_2 \quad (5.51)$$

$$x_j \in \{0, 1\} \quad j \in J \quad (5.52)$$

$$y_{ij} \in [0, 1] \quad i \in I, \quad j \in J \quad (5.53)$$

Una vez planteado comenzamos a dividir el problema en maestro y subproblema, quedando el problema maestro como:

$$\text{Minimizar} \quad z = 22x_1 + 14x_2 \quad (5.54)$$

$$\text{Sujeto a :} \quad x_1 + x_2 = 2 \quad (5.55)$$

$$x_j \in \{0, 1\} \quad j \in J \quad (5.56)$$

Y el subproblema como:

$$\text{Minimizar} \quad z = 8y_{11} + 5y_{12} + 9y_{21} + 4y_{22} \quad (5.57)$$

$$\text{Sujeto a :} \quad y_{11} + y_{12} = 1 \quad (5.58)$$

$$y_{21} + y_{22} = 1 \quad (5.59)$$

$$12y_{11} + 12y_{12} \leq 25x_1 \quad (5.60)$$

$$8y_{21} + 8y_{22} \leq 20x_2 \quad (5.61)$$

$$y_{ij} \in [0, 1] \quad i \in I, \quad j \in J \quad (5.62)$$

A continuación resolvemos el problema maestro relajado:

$$\text{Minimizar} \quad z = 22x_1 + 14x_2 + \theta \quad (5.63)$$

$$\text{Sujeto a :} \quad x_1 + x_2 = 2 \quad (5.64)$$

$$x_j \in \{0, 1\} \quad j \in J, \quad \theta \in \mathbb{R} \quad (5.65)$$

Que nos da una solución de $x_1 = 1, x_2 = 1, \theta = 0$, y $z = 36$. Con esta solución pasamos al subproblema, que quedaría como:

$$\text{Minimizar} \quad z = 8y_{11} + 5y_{12} + 9y_{21} + 4y_{22} \quad (5.66)$$

$$\text{Sujeto a :} \quad y_{11} + y_{12} = 1 \quad (5.67)$$

$$y_{21} + y_{22} = 1 \quad (5.68)$$

$$12y_{11} + 12y_{12} \leq 25 \quad (5.69)$$

$$8y_{21} + 8y_{22} \leq 20 \quad (5.70)$$

$$y_{ij} \in [0, 1] \quad i \in I, \quad j \in J \quad (5.71)$$

Posteriormente calculamos el dual del subproblema y lo resolvemos.

$$\text{Maximizar} \quad g = w_{11} + w_{12} + 25w_{21} + 20w_{22} \quad (5.72)$$

$$\text{Sujeto a :} \quad w_{11} + 12w_{21} \leq 8 \quad (5.73)$$

$$w_{11} + 12w_{22} \leq 5 \quad (5.74)$$

$$w_{12} + 8w_{21} \leq 9 \quad (5.75)$$

$$w_{12} + 8w_{22} \leq 4 \quad (5.76)$$

$$w_{2j} \leq 0 \quad j \in J \quad (5.77)$$

La solución que obtenemos es $w_{11} = 5, w_{12} = 4, w_{21} = 0, w_{22} = 0$, y $g = 9$. Entonces calculamos $\bar{z} - \underline{z}$. Donde $\underline{z} = 22x_1 + 14x_2 + \theta$ y $\bar{z} = 22x_1 + 14x_2 + w_{11} + w_{12} + 25w_{21} + 20w_{22}$. Tenemos que : $\bar{z} = 45$ y $\underline{z} = 36$, entonces $\bar{z} - \underline{z} = 45 - 36 = 9$.

Por tal razón tenemos que agregar el corte de optimalidad $5 + 4 + 0 + 0 \leq \theta$ al problema maestro, quedando como:

$$\text{Minimizar} \quad z = 22x_1 + 14x_2 + \theta \quad (5.78)$$

$$\text{Sujeto a :} \quad x_1 + x_2 = 2 \quad (5.79)$$

$$9 \leq \theta \quad (5.80)$$

$$x_j \in \{0, 1\} \quad j \in J, \quad \theta \in \mathbb{R} \quad (5.81)$$

Al resolverlo nos da una solución de $x_1 = 1, x_2 = 1, \theta = 9$, y $z = 45$. Con esta solución pasamos al subproblema, que quedaría como:

$$\text{Minimizar} \quad z = 8y_{11} + 5y_{12} + 9y_{21} + 4y_{22} \quad (5.82)$$

$$\text{Sujeto a :} \quad y_{11} + y_{12} = 1 \quad (5.83)$$

$$y_{21} + y_{22} = 1 \quad (5.84)$$

$$12y_{11} + 12y_{12} \leq 25 \quad (5.85)$$

$$8y_{21} + 8y_{22} \leq 20 \quad (5.86)$$

$$y_{ij} \in [0, 1] \quad i \in I, \quad j \in J \quad (5.87)$$

Transformando el subproblema a su forma dual:

$$\text{Maximizar} \quad g = w_{11} + w_{12} + 25w_{21} + 20w_{22} \quad (5.88)$$

$$\text{Sujeto a :} \quad w_{11} + 12w_{21} \leq 8 \quad (5.89)$$

$$w_{11} + 12w_{22} \leq 5 \quad (5.90)$$

$$w_{12} + 8w_{21} \leq 9 \quad (5.91)$$

$$w_{12} + 8w_{22} \leq 4 \quad (5.92)$$

$$w_{2j} \leq 0 \quad j \in J \quad (5.93)$$

Al resolverlo tenemos una solución de: $w_{11} = 5, w_{12} = 4, w_{21} = 0, w_{22} = 0$, y $g = 9$. Posteriormente calculamos $\bar{z} - \underline{z}$. Tenemos que : $\bar{z} = 45$ y $\underline{z} = 45$, entonces $\bar{z} - \underline{z} = 45 - 45 = 0$. Por lo tanto paramos, porque ya hemos resuelto el problema con una solución de: $x_1 = 1, x_2 = 1, y_{11} = 0, y_{12} = 1, y_{21} = 0, y_{22} = 1$, y $z = 45$

Posteriormente esta estrategia de solución fue implementada en el software GAMS con problemas de prueba, los cuales se resolvieron con esta herramienta, y cuyos resultados se muestran más adelante. El código de descomposición de Benders para resolver el problema de localización P-mediana en GAMS es el siguiente:

```
*-----
*Descomposición de Benders
*-----

*Inicialización
Scalar Zsup 'Cota superior' /INF/;
Scalar Zinf 'Cota inferior' /-INF/;
Solve maestroinicial minimizing zmi using mip;
Display ``Valores iniciales'', x.l;
*Algoritmo de Benders
Scalar converge /0/;
Scalar iteracion;
Scalar cota;
loop(iter$(not converge),
*Resolver subproblema
solve subproblema maximazing zsd using mip;
Display ``Solución al subproblema'', zsd.l;
*Resultados
abort$(subproblema.modelstat = 2)
``El subproblema dual es infactible''
*Subproblema no acotado
if(subproblema.modelstat = 3,
solve subproblemamodificado maximazing zsd using mip;
*Revisar resultados
abort$(subproblemamodificado.modelstat= 2)
```

```
``El problema modificado no alcanza la optimalidad'';  
*Agregar corte de factibilidad al maestro  
conjcortefac(iter)=yes;  
else  
*El problema es acotado  
cota=sum(j,f(j)*X.l(j))+zsd.l;  
if (cota < Zsup,  
Zsup=cota;  
);  
*Agregar corte de optimalidad  
conjcorte(iter)=yes;  
);  
*Constantes de Cortes  
constcorteu1(iter,i) = u1.l(i);  
constcorteu2(iter,j) = u2.l(j);  
*Resolver el problema maestro  
solve maestro minimizing zm using mip;  
*Resultados  
abort$(maestro.modelstat=4)  
``El problema maestro es infactible'';  
*Actualizar cota inferior  
Zinf=zm.l;  
iteracion=ord(iter);  
Display iteracion, Zinf, Zsup, zm.l;  
converge$((Zsup-Zinf)<0.1)=1;  
Display$converge ``El problema converge'';  
);  
abort$(not converge) ``El problema no converge'';  
*subproblema.solprint=2;  
solve subproblema minimazing zsp using mip;  
*original.solprint=2;  
solve original minimizing z using mip;  
abort$(original.modelstat>2) ``El problema no alcanza la  
optimalidad''  
Display ``Solución'', z.l, x.l, xo.l, yo.l, zm.l, rsub.m;
```

Los problemas prueba para el algoritmo anterior fueron obtenidos de un banco de problemas de localización [8]. Para mostrar que el algoritmo funciona, se compararon los resultados de la función objetivo que arroja el software de GAMS resolviendo con el solver CPLEX que tiene predeterminado contra el código de Benders anterior.

La forma que tienen los problemas es de $I \times J$, así por ejemplo para los tres primeros problemas, I tiene una cantidad de 20 nodos de demanda. Mientras que J tiene una cantidad de 10 nodos candidatos para localizar instalaciones.

Resultados			
Instancia	Problema	Solver de GAMS	Descomposición de Benders
1	20x10	2784.3	2784.3
3	20x10	8812.29	8812.29
6	20x10	2820.50	2820.50
7	30x15	11193.48	11193.48
12	30x15	7667.19	7667.19
17	30x15	32137.00	32137.00
18	40x20	31048.75	31048.75
22	40x20	15011.61	15011.61
25	40x20	47458.90	47458.90
26	50x20	8937.68	8937.68
30	50x20	14332.98	14332.98
33	50x20	58023.16	58023.16
34	60x30	32987.59	32987.59
38	60x30	115026.42	115026.42
41	60x30	442991.86	442991.86
42	75x30	38436.18	38436.18
46	75x30	264866.26	264866.26
49	75x30	274902.04	274902.04
50	90x30	58760.96	58760.96
54	90x30	359268.43	359268.43
57	90x30	531815.36	531815.36

Después de comprobar el buen funcionamiento del algoritmo de la descomposición de Benders, ya podemos pasar a ponerlo a prueba con el caso de estudio.

Capítulo 6

Caso de Estudio

6.1. Localización de moto-ambulancias para la atención de accidentes de tránsito en la Ciudad de México

Debido a que los accidentes viales son una de las principales causas de muerte en la Ciudad de México, es necesario localizar bases de emergencia; estas bases tendrán como principal objetivo brindar los servicios de atención prehospitalaria (servicios de primeros auxilios). Estos servicios prehospitalarios consisten en brindar atención médica a personas que hayan sido víctimas de algún percance, y en caso de ser necesario transportarlos a un hospital.

Según la ley de servicios médicos de emergencia de los Estados Unidos de 1973, se establece que para las áreas urbanas se debe poder atender el 95 % de las solicitudes dentro de los 10 minutos siguientes a la llamada, mientras que para las áreas rurales se debe poder atender dentro de los 30 minutos recibida la llamada [4].

Penosamente en el año 2018 en la ciudad de México se reportaron 76, 070 accidentes viales [5]. Por este motivo los accidentes de tránsito se han convertido en un importante problema de salud pública, debido a las muertes y lesiones que producen, ya que generan un impacto considerable en la economía del país, por los costos sanitarios, sociales y económicos que ocasionan.

Cuando ocurre un accidente, la muerte y los diversos daños originados por los accidentes

pueden disminuirse con la oportuna intervención de personal capacitado. La rapidez de la atención de la emergencia y el traslado de las víctimas con lesiones desde donde se originó el accidente hasta un centro de atención médica puede salvar vidas, y reducir las consecuencias negativas de los accidentes. Los principales puntos donde ocurren los accidentes de tránsito en la Ciudad de México son los cruces de vialidades primarias y secundarias.

Pensando en esta situación vamos a suponer que cada una de las bases que se quieren localizar, cuenta con una capacidad para albergar varias moto-ambulancias. Esto se debe a que es más caro mantener una ambulancia que una motocicleta, además de que las moto-ambulancias pueden desplazarse de una manera más eficiente en una ciudad tan congestionada como lo es la Ciudad de México. Cabe señalar que estas moto-ambulancias tienen únicamente capacidad para un paramédico y para los instrumentos más esenciales que necesitan al realizar su trabajo.

Se localizaron 71 sitios candidatos para la instalación de las bases de emergencia, que corresponden a los módulos de vigilancia de la ciudad y están distribuidos en las 16 alcaldías de la Ciudad de México figura 6.1.

Por otro lado, se identificaron 48 nodos de demanda [5], que corresponden a los principales puntos de accidentalidad de la CDMX figura 6.2. Por este motivo necesitamos localizar 4 bases de los sitios candidatos para satisfacer la demanda, además de tener la posibilidad de distribuir un máximo de hasta 86 vehículos en las bases ya localizadas.

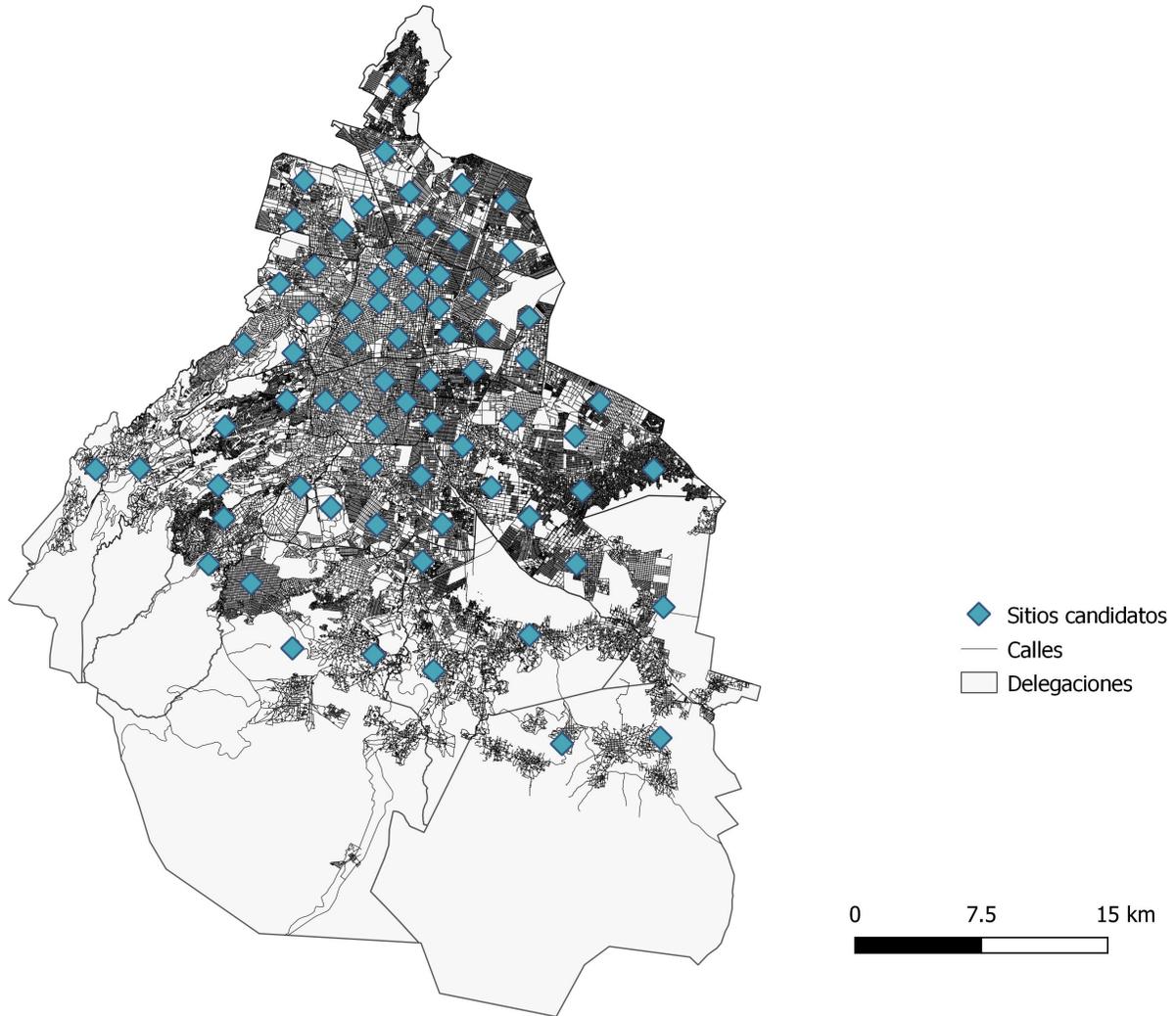


Figura 6.1: Sitios candidatos para localizar las bases de emergencias.

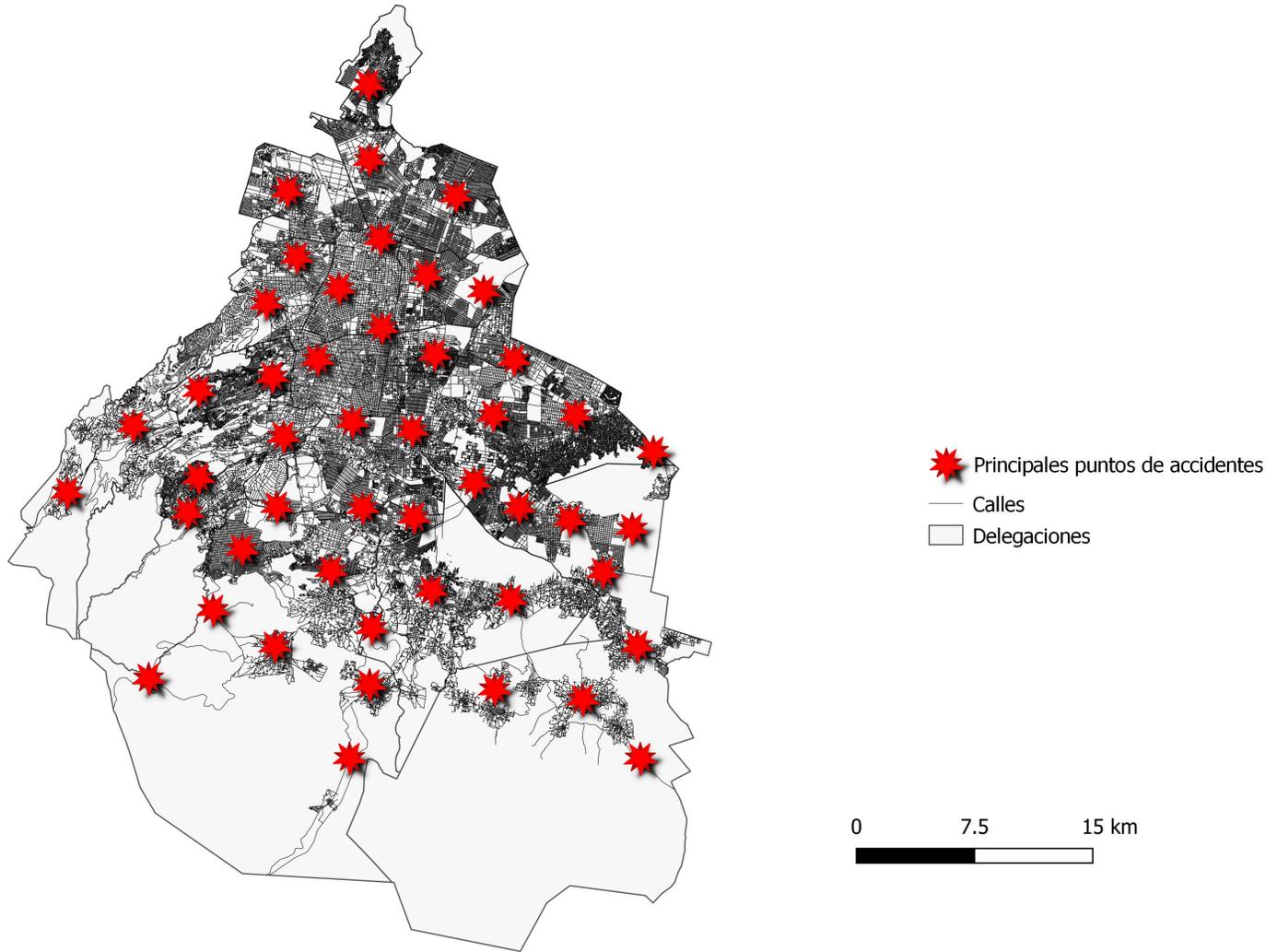


Figura 6.2: Principales puntos de accidentes viales.

El problema anterior puede formularse como:

$$\text{Minimizar} \quad \sum_{j \in J} f_j X_j + \sum_{j \in J} a_j W_j + \sum_{i \in I} \sum_{j \in J} c_{ij} Y_{ij} \quad (6.1)$$

$$\text{Sujeto a :} \quad \sum_{j \in J} Y_{ij} = 1 \quad i \in I \quad (6.2)$$

$$\sum_{j \in J} X_j = P \quad (6.3)$$

$$\sum_{j \in J} W_j \leq K \quad (6.4)$$

$$\sum_{i \in I} d_i Y_{ij} \leq W_j \quad j \in J \quad (6.5)$$

$$W_j \geq X_j \quad (6.6)$$

$$\sum_{j \in J} Y_{ij} \geq \lambda \quad (6.7)$$

$$Y_{ij} \leq X_j \quad (6.8)$$

$$Y_{ij} \leq 1 \quad (6.9)$$

$$W_j \in \mathbb{Z}^+ \quad j \in J \quad (6.10)$$

$$X_j \in \{0, 1\} \quad j \in J \quad (6.11)$$

$$Y_{ij} \in [0, 1] \quad i \in I, \quad j \in J \quad (6.12)$$

Donde definimos la siguiente notación:

Conjuntos

$I = \{1, \dots, i, \dots, s\}$ Conjunto de los nodos de demanda

$J = \{1, \dots, j, \dots, r\}$ Conjunto de los sitios candidatos

Parámetros

d_i = Demanda del nodo i

$c_{i,j}$ = Distancia entre el nodo de demanda i y la instalación j

f_j = Costo fijo asociado con la instalación abierta j

a_j = Costo fijo asociado con los vehículos

P = Número deseado de instalaciones por abrir

K = Número de vehículos por ubicar

λ = Porcentaje de solicitudes atendidas

Variables

W_j = Variable entera de decisión, la cual corresponde a la cantidad de vehículos por localizar

Y_{ij} = Variable continua de decisión, la cual corresponde al porcentaje de la demanda del nodo i atendida por la instalación j

$$X_j = \begin{cases} 1 & \text{si la instalación ubicada en el nodo } j \text{ está abierta} \\ 0 & \text{de lo contrario} \end{cases}$$

La función objetivo 6.1 expresa la minimización del costo total. La restricción 6.2 asegura que la demanda de cada nodo es satisfecha. La restricción 6.3 establece que el número de instalaciones abiertas sea P . La restricción 6.4 establece que el número de vehículos por localizar sea K . La restricción 6.5 establece la conexión entre las variables Y_{ij} y W_j . Estas variables establecen que ningún nodo de demanda puede ser atendido si no hay vehículos disponibles. La restricción 6.6 asegura que hay vehículos en las instalaciones localizadas. La restricción 6.7 obliga a la solución a cumplir con un cierto nivel de cobertura. La restricción 6.12 proporciona la relajación de la variable Y_{ij} . La restricción 6.8 garantiza que, si alguna instalación se encuentra en el nodo j , la demanda debe ser respondida por sí misma. La restricción 6.9 garantiza que la variable no exceda el límite de la cobertura. Finalmente la restricción 6.11 especifica que la variable de decisión es binaria, mientras que la restricción 6.10 establece que la variable de decisión es entera positiva.

Una vez planteado el problema, vamos a utilizar el algoritmo de descomposición de Benders en GAMS descrito en el apéndice B para poderlo resolver. Los datos del problema son los que aparecen en el apéndice A.

La propuesta de solución obtenida por el algoritmo es la siguiente:

Resultados			
Módulo	Alcaldía	Bases	Vehículos
4	Miguel Hidalgo	1	1
10	Benito Juárez	1	83
13	Azcapotzalco	1	1
68	Cuauhtémoc	1	1
Valor total de la función objetivo			80723846.16
Valor monetario			78723571.66
Valor distancia			2000274.50

A continuación se presenta una tabla que muestra a los sitios candidatos seleccionados, y los puntos de accidentes que cubren.

Cobertura de los sitios candidatos seleccionados			
Sitio Miguel Hidalgo	Sitio Benito Juárez	Sitio Azcapotzalco	Sitio Cuauhtémoc
6	1	21	21
7	2		
8	3		
13	4		
14	5		
15	9		
16	10		
17	11		
18	12		
19	20		
24	22		
44	23		
45	25		
46	26		
47	27		
	28		
	29		
	30		
	31		
	32		

Cobertura de los sitios candidatos seleccionados			
Sitio Miguel Hidalgo	Sitio Benito Juárez	Sitio Azcapotzalco	Sitio Cuauhtémoc
	33		
	34		
	35		
	36		
	37		
	38		
	39		
	40		
	41		
	42		
	43		
	48		

Tabla 6.1: Cobertura de los puntos de accidentes por los sitios seleccionados.

Seguidamente se presenta la figura 6.3 que muestra a las bases seleccionadas de los sitios candidatos, y la figura 6.4 muestra la cobertura que tendrán las bases.

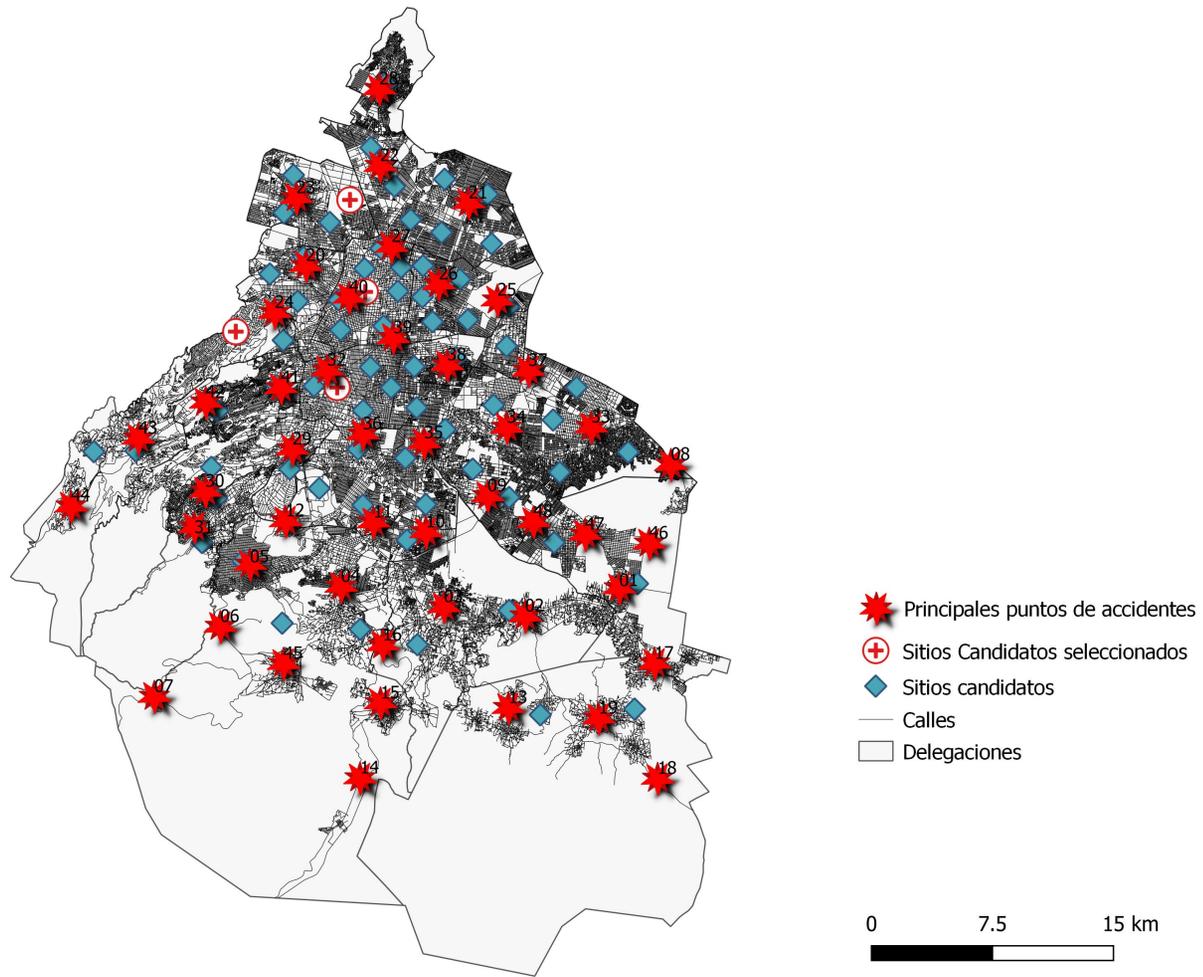


Figura 6.3: Mapa con las soluciones obtenidas por GAMS.

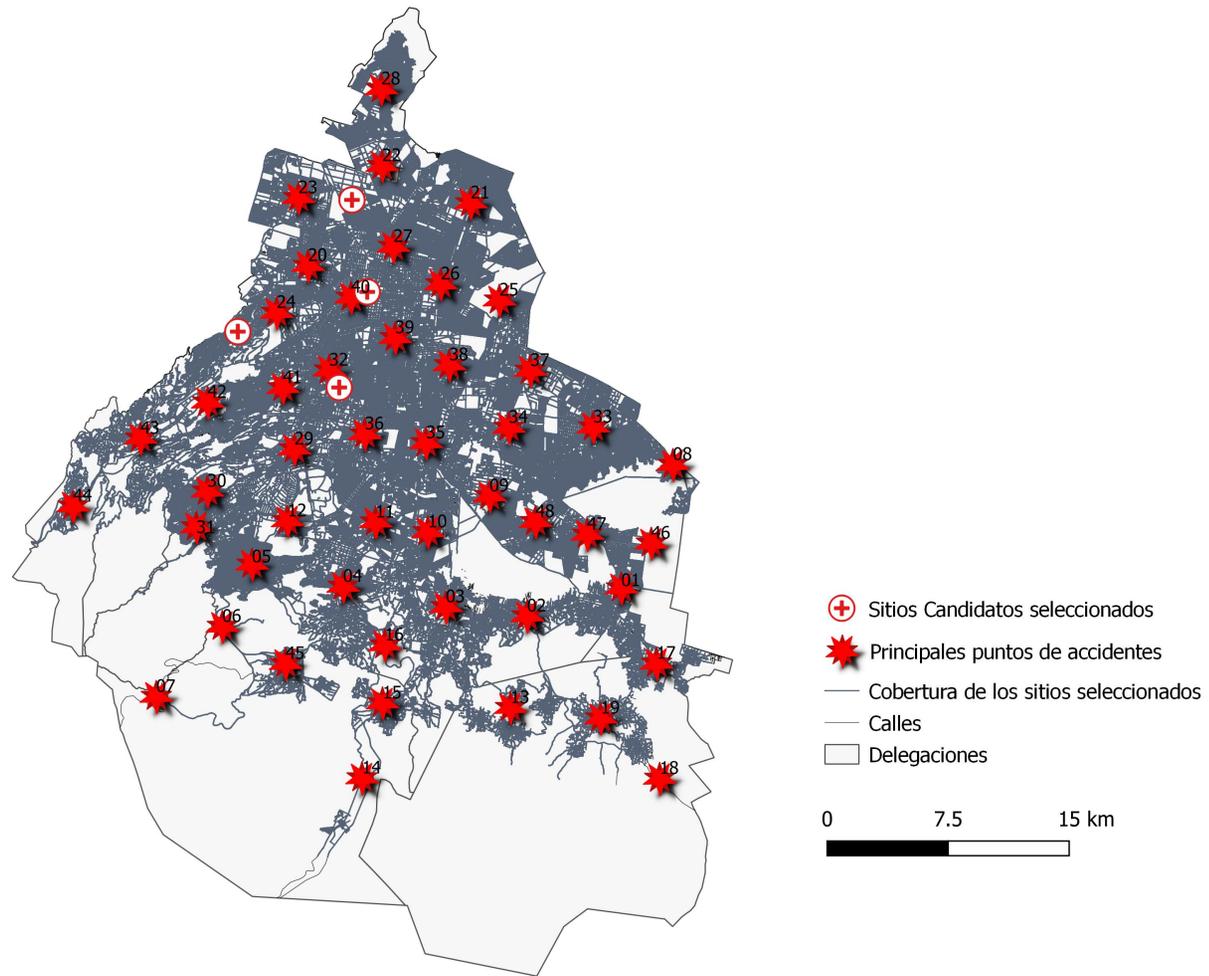


Figura 6.4: Cobertura de los sitios candidatos.

Capítulo 7

Conclusiones

En este trabajo se pueden observar las diferencias que hay entre los principales modelos de localización que existen, ya que para el modelo de localización continua se cuenta con la libertad de localizar las instalaciones en cualquier parte del plano, mientras que en el modelo de localización discreta se cuenta con un cierto conocimiento del lugar en donde se desea localizar las instalaciones, es decir se limitan los posibles sitios candidatos en donde se pueden localizar las instalaciones, para finalmente llegar a los modelos de localización en redes, los cuales podrían ser una mejor opción para resolver problemas reales ya que muchas situaciones pueden ser modeladas como redes, en donde los nodos pueden ser por ejemplo países, estados o alcaldías y las aristas representan las conexiones que existen entre ellos.

Posteriormente nos centramos en el modelo de la P-mediana el cual pertenece a los modelos de localización en redes, pues este modelo tiene como principal objetivo minimizar la distancia promedio ponderada que existe entre los nodos de demanda con las instalaciones, esto es muy importante debido a que hay casos en los cuales es necesario que la distancia entre las instalaciones y los puntos de demanda sea mínima, como por ejemplo las bases de emergencias, ya que estas tienen que brindar una respuesta rápida a los accidentes viales que se generan, por ejemplo en la Ciudad de México.

De este modelo aprendimos que es $NP - Duro$ en graficas generales (es decir, no cuenta con soluciones exactas eficientes), además, de que es mejor ubicar las instalaciones en los nodos y no en las aristas de la red, asimismo que el modelo de la P-mediana tiene una variante, la cual es el modelo del problema de localización de la planta capacitada. La

peculiaridad de esta variante es que se puede modelar de una mejor manera problemas reales con capacidades limitadas, de ahí su utilidad.

Las formas más comunes que existen para resolver el modelo de la P-mediana y que se mencionaron son: los métodos exactos y los métodos heurísticos. Pero para este trabajo se propuso una alternativa de solución con un método heurístico no tan conocido que es la descomposición de Benders.

Este método de descomposición se usa generalmente para resolver problemas de programación lineal entera mixta. Por eso fue necesario trabajar con la variante del modelo de la P-mediana, pues para que la descomposición de Benders pueda funcionar, es indispensable que la variable del subproblema sea continua. Por esta razón se consideró a la variable Y_{ij} como la proporción de la demanda del nodo que es atendida por la instalación. Con estas nuevas consideraciones pudimos aplicar la descomposición de Benders en la variante para solucionarla.

Esta forma alterna de solucionar el modelo de la P-mediana la pusimos a prueba en el software de GAMS con un banco de problemas de localización en donde vimos que efectivamente se puede resolver.

Gracias a esta forma alterna, se pudo dar una propuesta de solución al caso de estudio el cual consistió en la localización de bases de emergencia. Con la ayuda del código de descomposición de Benders escrito en GAMS aplicada al modelo de la planta capacitada, obtuvimos una propuesta de solución con la que fue suficiente localizar cuatro bases de emergencia de los sitios candidatos para cubrir el 95 % de las llamadas de emergencia de los principales puntos de accidentes que se puedan generar en la Ciudad de México.

Una observación importante de los resultados obtenidos, es que el código de la descomposición de Benders escrito en GAMS determino que era mejor localizar una base lo más céntrica posible para así poder cumplir con todos los requerimientos del problema: minimizar la distancia entre la base de emergencia con los sitios de accidentes viales, el costo de localizar la base, y el costo de las moto-ambulancias. Por esta razón, fue que seleccionó y doto de una gran cantidad de moto-ambulancias a la base de emergencia ubicada en la alcaldía Benito Juárez, ya que era la mejor opción en cuanto a costo y ubicación. Mientras que el resto de bases ubicadas fueron consideradas para atender a las demandas que escapaban de la cobertura del sitio ubicado en la alcaldía Benito Juárez.

Otra observación importante es que el problema resuelto en el caso de estudio, se trabajó

con el promedio total de los principales puntos de accidentes viales del año 2018 además de suponer que los mismos ocurrían diariamente. Por eso el número de vehículos que se necesitaba para cubrir los accidentes viales fue tan grande, en comparación a que si se hubiera trabajado únicamente con los datos de algún mes del año. Este cambio hubiera provocado un reajuste en el número de bases y posible ubicación, además de la cantidad de vehículos.

Por último, es posible que haciendo los ajustes pertinentes, el método de descomposición de Benders pueda utilizarse para resolver alguno de los otros modelos de localización en redes, con la finalidad de ubicar otro tipo de instalaciones que ofrezcan servicios diferentes al tratado en este trabajo. Además, fue bastante importante poder utilizar el método de descomposición de Benders en un tema que no estuviera relacionado con la optimización de producción de bienes.

Bibliografía

- [1] Jesús Alegre, Alberto Aragón, Silvia Casado, Cristina Delgado, and Joaquín Pacheco. Resolución de 2 modelos de localización mediante búsqueda dispersa. 2003.
- [2] J. F. Benders. Partitioning procedures for solving mixed-variables programming problems. *Numerische Mathematik*, 4(1):238–252, Dec 1962.
- [3] Zaida Estefanía Alarcón Bernal. *Estrategias de Solución Para Problemas Binivel Discretos-Continuos*. PhD thesis, Universidad Nacional Autónoma de México, 2017.
- [4] Luce Brotcorne, Gilbert Laporte, and Frederic Semet. Ambulance location and relocation models. *European journal of operational research*, 147(3):451–463, 2003.
- [5] C. C. Centro de Comando. *Datos Abiertos Ciudad de México. Obtenido de Datos Abiertos Ciudad de México. https://datos.cdmx.gob.mx/explore/dataset/incidentes-viales-c5/information/?disjunctive.incidente_c4&refine.ano=2018*. 2018.
- [6] María José Canós Darós, Manuel Mocholi Arce, and María Luisa Martínez Romero. Eficiencia versus equidad en localización: aplicación al diseño de infraestructuras. *Rect@: Revista Electrónica de Comunicaciones y Trabajos de ASEPUMA*, (10):59–76, 2009.
- [7] Mark S Daskin. Network and discrete location: Models, algorithms, and applications. 1995.
- [8] Elena Fernández Aréizaga . *Single Source Capacitated Plant Location Problem Instances. <http://www-eio.upc.es/~elena/sscplp/index.html>*. 2006.
- [9] Reza Zanjirani Farahani and Masoud Hekmatfar. *Facility location: concepts, models, algorithms and case studies*. 2009.

-
- [10] Ricardo Aceves García. El problema de localización de servicios. *Departamento de Matemáticas*, page 1, 2007.
- [11] AJ Goldman. Optimal center location in simple networks. *Transportation science*, 5(2):212–221, 1971.
- [12] S Louis Hakimi. Optimum locations of switching centers and the absolute centers and medians of a graph. *Operations research*, 12(3):450–459, 1964.
- [13] S Louis Hakimi. Optimum distribution of switching centers in a communication network and some related graph theoretic problems. *Operations research*, 13(3):462–475, 1965.
- [14] F Hillier and G Lieberman. *Investigación de operaciones*. 2010.
- [15] Masoomeh Jamshidi. Median location problem. In *Facility Location*, pages 177–191. Springer, 2009.
- [16] Pertti Järvinen, Jaakko Rajala, and Heikki Sinervo. A branch-and-bound algorithm for seeking the p-median. *Operations Research*, 20(1):173–178, 1972.
- [17] O. Kariv and S. Hakimi. An algorithmic approach to network location problems. ii: The p-medians. *SIAM Journal on Applied Mathematics*, 37(3):539–560, 1979.
- [18] Gilbert Laporte, Stefan Nickel, and Francisco Saldanha da Gama. *Location science*, volume 528. Springer, 2015.
- [19] Subhash C Narula, Ugonnaya I Ogbu, and Haakon M Samuelsson. An algorithm for the p-median problem. *Operations Research*, 25(4):709–713, 1977.
- [20] Mauricio GC Resende and Celso C Ribeiro. *Optimization by GRASP*. Springer, 2016.
- [21] Charles S ReVelle and Horst A Eiselt. Location analysis: A synthesis and survey. *European journal of operational research*, 165(1):1–19, 2005.
- [22] Kenneth E Rosing, CS ReVelle, and H Rosing-Vogelaar. The p-median and its linear programming relaxation: An approach to large problems. *Journal of the Operational Research Society*, 30(9):815–823, 1979.

- [23] Daniel Serra and Vladimir Marianov. The p-median problem in a changing network: the case of barcelona. *Location Science*, 6(1-4):383–394, 1998.
- [24] Stephanie A Snyder and Robert G Haight. Application of the maximal covering location problem to habitat reserve site selection: a review. *International Regional Science Review*, 39(1):28–47, 2016.

Apéndice A

Datos del problema del caso de estudio

A continuación presentaremos unas tablas que contienen la información utilizada para realizar el caso de estudio.

Costos de las instalaciones		
Módulo	Alcaldía	Costo
1	VENUSTIANO CARRANZA	187,069
2	CUAUHTEMOC	185,729
3	CUAUHTEMOC	185,729
4	MIGUEL HIDALGO	187,157
5	IZTAPALAPA	186,029
6	AZCAPOTZALCO	187,016
7	VENUSTIANO CARRANZA	187,069
8	GUSTAVO A MADERO	186,386
9	COYOACAN	186,593
10	BENITO JUAREZ	185,547
11	IZTAPALAPA	186,029
12	IZTACALCO	186,425
13	AZCAPOTZALCO	187,016
14	CUAUHTEMOC	185,729
15	BENITO JUAREZ	185,547
16	BENITO JUAREZ	185,547
17	IZTAPALAPA	186,029
18	IZTACALCO	186,425
19	BENITO JUAREZ	185,547
20	GUSTAVO A MADERO	186,386
21	IZTAPALAPA	186,029
22	GUSTAVO A MADERO	186,386
23	CUAUHTEMOC	185,729
24	IZTAPALAPA	186,029
25	MIGUEL HIDALGO	187,157
26	MIGUEL HIDALGO	187,157
27	COYOACAN	186,593
28	IZTAPALAPA	186,029

Costos de las instalaciones		
Módulo	Alcaldía	Costo
29	GUSTAVO A MADERO	186,386
30	IZTAPALAPA	186,029
31	IZTACALCO	186,425
32	CUAUHTEMOC	185,729
33	COYOACAN	186,593
34	TLAHUAC	187,221
35	VENUSTIANO CARRANZA	187,069
36	IZTAPALAPA	186,029
37	MILPA ALTA	186,546
38	BENITO JUAREZ	185,547
39	AZCAPOTZALCO	187,016
40	TLALPAN	185,677
41	GUSTAVO A MADERO	186,386
42	AZCAPOTZALCO	187,016
43	MIGUEL HIDALGO	187,157
44	MIGUEL HIDALGO	187,157
45	COYOACAN	186,593
46	CUAJIMALPA	186,090
47	CUAJIMALPA	186,090
48	ALVARO OBREGON	186,141
49	ALVARO OBREGON	186,141
50	MAGDALENA CONTRERAS	186,428
51	XOCHIMILCO	185,756
52	XOCHIMILCO	185,756
53	COYOACAN	186,593
54	VENUSTIANO CARRANZA	187,069
55	VENUSTIANO CARRANZA	187,069
56	GUSTAVO A MADERO	186,386
57	VENUSTIANO CARRANZA	187,069
58	MAGDALENA CONTRERAS	186,428
59	TLALPAN	185,677
60	TLAHUAC	187,221
61	ALVARO OBREGON	186,141
62	TLALPAN	185,677
63	TLALPAN	185,677
64	MILPA ALTA	186,546
65	GUSTAVO A MADERO	186,386
66	ALVARO OBREGON	186,141
67	CUAUHTEMOC	185,729
68	CUAUHTEMOC	185,729
69	CUAUHTEMOC	185,729
70	GUSTAVO A MADERO	186,386
71	IZTAPALAPA	186,029

Tabla A.1: Tabla de los costos por localizar las instalaciones en los sitios candidatos.

Apéndice A. Datos del problema del caso de estudio

Costos de los vehículos		
Módulo	Alcaldía	Costo
1	VENUSTIANO CARRANZA	914,755
2	CUAUHTEMOC	908,203
3	CUAUHTEMOC	908,203
4	MIGUEL HIDALGO	915,186
5	IZTAPALAPA	909,669
6	AZCAPOTZALCO	914,497
7	VENUSTIANO CARRANZA	914,755
8	GUSTAVO A MADERO	911,418
9	COYOACAN	912,426
10	BENITO JUAREZ	907,312
11	IZTAPALAPA	909,669
12	IZTACALCO	911,609
13	AZCAPOTZALCO	914,497
14	CUAUHTEMOC	908,203
15	BENITO JUAREZ	907,312
16	BENITO JUAREZ	907,312
17	IZTAPALAPA	909,669
18	IZTACALCO	911,609
19	BENITO JUAREZ	907,312
20	GUSTAVO A MADERO	911,418
21	IZTAPALAPA	909,669
22	GUSTAVO A MADERO	911,418
23	CUAUHTEMOC	908,203
24	IZTAPALAPA	909,669
25	MIGUEL HIDALGO	915,186
26	MIGUEL HIDALGO	915,186
27	COYOACAN	912,426
28	IZTAPALAPA	909,669
29	GUSTAVO A MADERO	911,418
30	IZTAPALAPA	909,669
31	IZTACALCO	911,609
32	CUAUHTEMOC	908,203
33	COYOACAN	912,426
34	TLAHUAC	915,497
35	VENUSTIANO CARRANZA	914,755
36	IZTAPALAPA	909,669
37	MILPA ALTA	912,200
38	BENITO JUAREZ	907,312
39	AZCAPOTZALCO	914,497
40	TLALPAN	907,951
41	GUSTAVO A MADERO	911,418
42	AZCAPOTZALCO	914,497
43	MIGUEL HIDALGO	915,186
44	MIGUEL HIDALGO	915,186
45	COYOACAN	912,426
46	CUAJIMALPA	909,970
47	CUAJIMALPA	909,970
48	ALVARO OBREGON	910,218
49	ALVARO OBREGON	910,218
50	MAGDALENA CONTRERAS	911,622
51	XOCHIMILCO	908,334
52	XOCHIMILCO	908,334

Costos de los vehículos		
Módulo	Alcaldía	Costo
53	COYOACAN	912,426
54	VENUSTIANO CARRANZA	914,755
55	VENUSTIANO CARRANZA	914,755
56	GUSTAVO A MADERO	911,418
57	VENUSTIANO CARRANZA	914,755
58	MAGDALENA CONTRERAS	911,622
59	TLALPAN	907,951
60	TLAHUAC	915,497
61	ALVARO OBREGON	910,218
62	TLALPAN	907,951
63	TLALPAN	907,951
64	MILPA ALTA	912,200
65	GUSTAVO A MADERO	911,418
66	ALVARO OBREGON	910,218
67	CUAUHTEMOC	908,203
68	CUAUHTEMOC	908,203
69	CUAUHTEMOC	908,203
70	GUSTAVO A MADERO	911,418
71	IZTAPALAPA	909,669

Tabla A.2: Tabla de los costos de los vehículos, incluyen el mantenimiento de los vehículos, y el sueldo de los paramédicos.

Apéndice A. Datos del problema del caso de estudio

Accidentes (demandas)		
Id.	Año 2018	promedio diario
1	434	1
2	187	1
3	401	1
4	406	1
5	792	2
6	68	0
7	8	0
8	149	0
9	671	2
10	329	1
11	751	2
12	737	2
13	94	0
14	105	0
15	162	0
16	0	0
17	88	0
18	27	0
19	101	0
20	1,736	5
21	726	2
22	733	2
23	1,018	3
24	309	1
25	1,587	4
26	2,018	6
27	1,442	4
28	662	2
29	825	2
30	402	1
31	25	0
32	2,639	7
33	583	2
34	1,018	3
35	750	2
36	1,072	3
37	1,148	3
38	1,107	3
39	1,743	5
40	2,349	6
41	434	1

Accidentes (demandas)		
Id.	Año 2018	promedio diario
42	579	2
43	401	1
44	142	0
45	22	0
46	0	0
47	0	0
48	1,150	3

Tabla A.3: Tabla del promedio de los accidentes ocurridos en el año 2018.

Apéndice A. Datos del problema del caso de estudio

	1	2	3	4	5	6	7	8	9	10	11	12	13
1	100000	100000	100000	100000	100000	100000	100000	100000	100000	100000	13	100000	100000
2	100000	100000	100000	100000	100000	100000	100000	100000	12	100000	100000	100000	100000
3	100000	100000	100000	100000	13	100000	100000	100000	7	100000	11	100000	100000
4	100000	100000	100000	100000	13	100000	100000	100000	8	14	13	100000	100000
5	100000	100000	100000	100000	100000	100000	100000	100000	13	14	100000	100000	100000
6	100000	100000	100000	100000	100000	100000	100000	100000	100000	100000	100000	100000	100000
7	100000	100000	100000	100000	100000	100000	100000	100000	100000	100000	100000	100000	100000
8	100000	100000	100000	100000	100000	100000	100000	100000	100000	100000	14	100000	100000
9	14	14	100000	100000	8	100000	100000	100000	5	13	3	11	100000
10	100000	14	100000	100000	8	100000	100000	100000	2	12	7	11	100000
11	100000	13	100000	100000	9	100000	100000	100000	4	10	9	11	100000
12	100000	15	100000	100000	13	100000	100000	100000	10	10	15	15	100000
13	100000	100000	100000	100000	100000	100000	100000	100000	100000	100000	100000	100000	100000
14	100000	100000	100000	100000	100000	100000	100000	100000	100000	100000	100000	100000	100000
15	100000	100000	100000	100000	100000	100000	100000	100000	100000	100000	100000	100000	100000
16	100000	100000	100000	100000	100000	100000	100000	100000	11	100000	100000	100000	100000
17	100000	100000	100000	100000	100000	100000	100000	100000	100000	100000	100000	100000	100000
18	100000	100000	100000	100000	100000	100000	100000	100000	100000	100000	100000	100000	100000
19	100000	100000	100000	100000	100000	100000	100000	100000	100000	100000	100000	100000	100000
20	14	7	5	7	13	4	8	10	100000	9	100000	10	6
21	10	11	9	100000	14	13	7	3	100000	100000	100000	11	8
22	100000	11	8	100000	100000	8	10	7	100000	15	100000	14	4
23	100000	11	7	10	100000	1	11	10	100000	13	100000	15	4
24	100000	8	8	3	12	8	10	13	100000	7	100000	11	10
25	1	9	11	100000	10	100000	7	9	100000	13	13	9	14
26	6	5	6	14	8	12	2	4	15	11	100000	6	9
27	10	5	3	13	11	8	4	4	100000	10	100000	9	4
28	100000	100000	13	100000	100000	13	15	11	100000	100000	100000	100000	9
29	100000	11	14	12	9	100000	14	100000	10	6	13	11	100000
30	100000	100000	100000	100000	100000	100000	100000	100000	100000	12	100000	100000	100000
31	100000	100000	100000	100000	100000	100000	100000	100000	100000	14	100000	100000	100000
32	13	5	8	8	7	11	9	13	12	2	14	6	11
33	11	100000	100000	100000	12	100000	100000	100000	14	100000	9	13	100000
34	9	12	100000	100000	6	100000	12	100000	10	12	5	8	100000
35	11	9	14	100000	2	100000	9	14	5	8	5	5	100000
36	14	7	12	13	5	100000	11	100000	7	4	9	6	100000
37	5	10	14	100000	9	100000	9	12	14	14	9	8	100000
38	6	5	10	100000	4	100000	6	9	10	8	10	2	14
39	8	1	6	11	6	12	4	8	12	5	13	3	10
40	11	3	3	8	9	8	5	8	100000	6	100000	7	7
41	100000	8	11	7	9	13	12	100000	13	4	100000	9	13
42	100000	14	100000	6	100000	100000	100000	100000	100000	10	100000	100000	100000
43	100000	100000	100000	10	100000	100000	100000	100000	100000	14	100000	100000	100000
44	100000	100000	100000	100000	100000	100000	100000	100000	100000	100000	100000	100000	100000
45	100000	100000	100000	100000	100000	100000	100000	100000	15	100000	100000	100000	100000
46	100000	100000	100000	100000	100000	100000	100000	100000	100000	100000	13	100000	100000
47	100000	100000	100000	100000	14	100000	100000	100000	12	100000	9	100000	100000
48	100000	100000	100000	100000	12	100000	100000	100000	9	100000	6	14	100000

Tabla A.4: Distancia de los nodos de demanda a los sitios candidatos.

Apéndice A. Datos del problema del caso de estudio

14	15	16	17	18	19	20	21	22	23	24	25	26
100000	100000	100000	15	100000	100000	100000	9	100000	100000	13	100000	100000
100000	100000	100000	100000	100000	100000	100000	100000	100000	100000	100000	100000	100000
100000	100000	100000	100000	100000	14	100000	15	100000	100000	100000	100000	100000
100000	14	14	100000	100000	12	100000	100000	100000	100000	100000	100000	100000
100000	13	100000	100000	100000	14	100000	100000	100000	100000	100000	100000	100000
100000	100000	100000	100000	100000	100000	100000	100000	100000	100000	100000	100000	100000
100000	100000	100000	100000	100000	100000	100000	100000	100000	100000	100000	100000	100000
100000	100000	100000	9	14	100000	100000	10	100000	100000	9	100000	100000
100000	15	11	10	11	11	100000	6	100000	100000	8	100000	100000
100000	13	10	100000	15	9	100000	11	100000	15	13	100000	100000
100000	11	9	100000	100000	8	100000	14	100000	13	100000	100000	100000
100000	9	13	100000	100000	10	100000	100000	100000	13	100000	100000	100000
100000	100000	100000	100000	100000	100000	100000	100000	100000	100000	100000	100000	100000
100000	100000	100000	100000	100000	100000	100000	100000	100000	100000	100000	100000	100000
100000	100000	100000	100000	100000	100000	100000	100000	100000	100000	100000	100000	100000
100000	100000	100000	100000	100000	100000	100000	100000	100000	100000	100000	100000	100000
100000	100000	100000	100000	100000	100000	100000	100000	100000	100000	100000	100000	100000
100000	100000	100000	100000	100000	100000	100000	100000	100000	100000	100000	100000	100000
100000	100000	100000	100000	100000	100000	100000	100000	100000	100000	100000	100000	100000
100000	100000	100000	100000	100000	100000	100000	100000	100000	100000	100000	100000	100000
7	8	10	100000	14	11	14	100000	11	5	100000	3	1
7	100000	14	100000	11	100000	2	100000	2	13	100000	100000	13
7	100000	15	100000	100000	100000	8	100000	5	11	100000	12	9
9	12	15	100000	100000	15	14	100000	11	9	100000	6	4
9	6	10	100000	100000	9	100000	100000	100000	5	100000	3	5
9	14	11	9	3	13	10	13	13	12	10	100000	15
3	12	8	12	6	11	7	100000	7	8	13	12	9
2	11	9	100000	10	11	8	100000	6	7	100000	9	6
13	100000	100000	100000	100000	100000	12	100000	10	100000	100000	100000	14
14	5	8	100000	100000	6	100000	100000	100000	9	100000	12	13
100000	11	15	100000	100000	12	100000	100000	100000	100000	100000	100000	100000
100000	13	100000	100000	100000	14	100000	100000	100000	100000	100000	100000	100000
9	2	5	100000	12	4	100000	100000	15	3	100000	8	8
100000	100000	14	4	9	100000	100000	4	100000	100000	3	100000	100000
14	13	9	7	6	10	100000	5	100000	14	4	100000	100000
12	9	5	12	9	5	100000	10	100000	10	10	100000	100000
11	5	4	100000	12	2	100000	14	100000	7	13	13	12
11	15	10	4	3	12	13	8	100000	13	4	100000	100000
8	9	5	9	5	8	12	12	13	8	9	14	12
4	6	3	13	7	6	13	100000	11	4	13	10	8
4	6	7	100000	11	8	12	100000	10	2	100000	6	4
11	3	7	100000	100000	6	100000	100000	100000	6	100000	8	9
100000	9	14	100000	100000	13	100000	100000	100000	12	100000	11	13
100000	13	100000	100000	100000	100000	100000	100000	100000	100000	100000	15	100000
100000	100000	100000	100000	100000	100000	100000	100000	100000	100000	100000	100000	100000
100000	100000	100000	100000	100000	100000	100000	100000	100000	100000	100000	100000	100000
100000	100000	100000	14	100000	100000	100000	9	100000	100000	12	100000	100000
100000	100000	100000	11	15	100000	100000	5	100000	100000	9	100000	100000
100000	100000	14	10	13	14	100000	4	100000	100000	8	100000	100000

Tabla A.5: Distancia de los nodos de demanda a los sitios candidatos.

Apéndice A. Datos del problema del caso de estudio

27	28	29	30	31	32	33	34	35	36	37	38	39
100000	14	100000	10	100000	100000	100000	7	100000	100000	13	100000	100000
100000	100000	100000	100000	100000	100000	14	14	100000	100000	7	100000	100000
10	100000	100000	10	100000	100000	9	13	100000	12	12	100000	100000
10	100000	100000	14	100000	100000	6	100000	100000	13	100000	15	100000
13	100000	100000	100000	100000	100000	10	100000	100000	100000	100000	100000	100000
100000	100000	100000	100000	100000	100000	15	100000	100000	100000	100000	100000	100000
100000	100000	100000	100000	100000	100000	100000	100000	100000	100000	100000	100000	100000
100000	4	100000	12	100000	100000	100000	11	100000	100000	100000	100000	100000
6	11	100000	2	11	100000	9	6	13	6	100000	13	100000
5	100000	100000	7	12	100000	5	10	14	7	100000	12	100000
5	100000	100000	10	13	100000	2	13	100000	9	100000	10	100000
10	100000	100000	100000	100000	100000	6	100000	100000	13	100000	12	100000
100000	100000	100000	100000	100000	100000	100000	100000	100000	100000	6	100000	100000
100000	100000	100000	100000	100000	100000	100000	100000	100000	100000	100000	100000	100000
100000	100000	100000	100000	100000	100000	15	100000	100000	100000	100000	100000	100000
14	100000	100000	15	100000	100000	10	100000	100000	100000	100000	100000	100000
100000	100000	100000	100000	100000	100000	100000	14	100000	100000	9	100000	100000
100000	100000	100000	100000	100000	100000	100000	100000	100000	100000	11	100000	100000
100000	100000	100000	100000	100000	100000	100000	100000	100000	100000	5	100000	100000
100000	100000	8	100000	12	5	100000	100000	11	100000	100000	8	6
100000	100000	5	100000	10	8	100000	100000	8	100000	100000	14	12
100000	100000	5	100000	15	6	100000	100000	13	100000	100000	13	7
100000	100000	8	100000	100000	7	100000	100000	14	100000	100000	12	2
14	100000	12	100000	13	9	100000	100000	13	15	100000	8	10
13	14	11	14	6	10	100000	100000	4	9	100000	11	100000
12	100000	5	100000	5	5	100000	100000	3	10	100000	8	13
14	100000	3	100000	9	1	100000	100000	7	14	100000	8	9
100000	100000	10	100000	100000	11	100000	100000	100000	100000	100000	100000	11
8	100000	100000	100000	14	15	7	100000	100000	10	100000	8	100000
14	100000	100000	100000	100000	100000	12	100000	100000	100000	100000	14	100000
100000	100000	100000	100000	100000	100000	12	100000	100000	100000	100000	100000	100000
9	100000	11	100000	9	9	10	100000	10	9	100000	3	13
14	3	100000	8	11	100000	100000	9	12	10	100000	100000	100000
8	9	100000	5	7	100000	12	9	9	4	100000	11	100000
2	14	15	7	6	14	7	11	9	2	100000	7	100000
4	100000	14	11	9	12	5	100000	12	6	100000	5	100000
12	9	14	9	5	13	100000	12	6	8	100000	11	100000
8	14	11	11	1	10	12	100000	4	5	100000	5	100000
8	100000	8	15	5	6	11	100000	5	8	100000	3	14
12	100000	7	100000	8	4	14	100000	8	12	100000	5	9
10	100000	14	100000	12	12	11	100000	13	11	100000	6	14
100000	100000	100000	100000	100000	100000	100000	100000	100000	100000	100000	12	100000
100000	100000	100000	100000	100000	100000	100000	100000	100000	100000	100000	100000	100000
100000	100000	100000	100000	100000	100000	100000	100000	100000	100000	100000	100000	100000
100000	100000	100000	100000	100000	100000	13	100000	100000	100000	100000	100000	100000
100000	13	100000	10	100000	100000	100000	7	100000	100000	100000	100000	100000
13	10	100000	6	100000	100000	100000	3	100000	12	100000	100000	100000
10	9	100000	3	13	100000	13	2	100000	10	100000	100000	100000

Tabla A.6: Distancia de los nodos de demanda a los sitios candidatos.

Apéndice A. Datos del problema del caso de estudio

40	41	42	43	44	45	46	47	48	49	50	51	52
100000	100000	100000	100000	100000	100000	100000	100000	100000	100000	100000	100000	8
100000	100000	100000	100000	100000	100000	100000	100000	100000	100000	100000	8	1
15	100000	100000	100000	100000	12	100000	100000	100000	100000	100000	4	5
8	100000	100000	100000	100000	10	100000	100000	14	100000	12	9	12
1	100000	100000	100000	100000	12	13	100000	9	100000	6	100000	100000
6	100000	100000	100000	100000	100000	100000	100000	13	100000	10	100000	100000
100000	100000	100000	100000	100000	100000	100000	100000	100000	100000	100000	100000	100000
100000	100000	100000	100000	100000	100000	100000	100000	100000	100000	100000	100000	100000
100000	100000	100000	100000	100000	9	100000	100000	100000	100000	100000	13	14
13	100000	100000	100000	100000	7	100000	100000	100000	100000	15	8	9
10	100000	100000	100000	100000	6	100000	100000	13	100000	11	9	11
4	100000	100000	15	13	8	13	100000	8	14	5	14	100000
100000	100000	100000	100000	100000	100000	100000	100000	100000	100000	100000	9	10
100000	100000	100000	100000	100000	100000	100000	100000	100000	100000	100000	12	100000
100000	100000	100000	100000	100000	100000	100000	100000	100000	100000	100000	6	12
12	100000	100000	100000	100000	15	100000	100000	100000	100000	100000	7	12
100000	100000	100000	100000	100000	100000	100000	100000	100000	100000	100000	100000	12
100000	100000	100000	100000	100000	100000	100000	100000	100000	100000	100000	100000	100000
100000	100000	100000	100000	100000	100000	100000	100000	100000	100000	100000	100000	10
100000	14	4	3	6	13	100000	100000	100000	13	100000	100000	100000
100000	11	10	14	100000	100000	100000	100000	100000	100000	100000	100000	100000
100000	6	7	12	14	100000	100000	100000	100000	100000	100000	100000	100000
100000	12	3	7	10	100000	100000	100000	100000	100000	100000	100000	100000
100000	100000	8	2	2	11	13	100000	13	9	100000	100000	100000
100000	100000	15	15	100000	15	100000	100000	100000	100000	100000	100000	100000
100000	100000	9	10	11	13	100000	100000	100000	100000	100000	100000	100000
100000	11	5	8	10	13	100000	100000	100000	100000	100000	100000	100000
100000	1	11	100000	100000	100000	100000	100000	100000	100000	100000	100000	100000
9	100000	100000	10	8	5	12	100000	6	9	6	100000	100000
6	100000	100000	100000	13	11	7	11	3	10	1	100000	100000
5	100000	100000	100000	100000	12	9	12	6	12	3	100000	100000
14	100000	11	5	4	6	14	100000	11	8	12	100000	100000
100000	100000	100000	100000	100000	100000	100000	100000	100000	100000	100000	100000	100000
100000	100000	100000	100000	100000	11	100000	100000	100000	100000	100000	100000	100000
100000	100000	100000	14	12	5	100000	100000	14	100000	15	14	100000
13	100000	15	11	9	1	100000	100000	10	12	10	100000	100000
100000	100000	100000	100000	100000	14	100000	100000	100000	100000	100000	100000	100000
100000	100000	14	12	11	10	100000	100000	100000	100000	100000	100000	100000
100000	100000	10	7	7	8	100000	100000	100000	13	100000	100000	100000
100000	15	6	4	5	10	100000	100000	100000	12	100000	100000	100000
13	100000	12	6	3	7	11	14	8	5	10	100000	100000
100000	100000	100000	11	8	14	6	9	7	5	11	100000	100000
13	100000	100000	14	12	100000	2	4	6	7	8	100000	100000
100000	100000	100000	100000	100000	100000	7	5	12	13	14	100000	100000
11	100000	100000	100000	100000	100000	100000	100000	100000	100000	15	13	100000
100000	100000	100000	100000	100000	100000	100000	100000	100000	100000	100000	100000	12
100000	100000	100000	100000	100000	100000	100000	100000	100000	100000	100000	100000	12
100000	100000	100000	100000	100000	13	100000	100000	100000	100000	100000	14	100000

Tabla A.7: Distancia de los nodos de demanda a los sitios candidatos.

Apéndice A. Datos del problema del caso de estudio

53	54	55	56	57	58	59	60	61	62	63	64	65
100000	100000	100000	100000	100000	100000	100000	2	100000	100000	100000	10	100000
100000	100000	100000	100000	100000	100000	10	9	100000	13	100000	11	100000
12	100000	100000	100000	100000	100000	5	14	100000	7	13	100000	100000
8	100000	100000	100000	100000	11	6	100000	100000	4	6	100000	100000
8	100000	100000	100000	100000	4	11	100000	13	10	5	100000	100000
14	100000	100000	100000	100000	7	100000	100000	100000	11	8	100000	100000
100000	100000	100000	100000	100000	100000	100000	100000	100000	100000	100000	100000	100000
100000	100000	100000	100000	100000	100000	100000	100000	100000	100000	100000	100000	100000
12	100000	100000	100000	13	100000	7	12	100000	14	100000	100000	100000
8	100000	100000	100000	14	100000	2	100000	14	10	13	100000	100000
5	100000	100000	100000	14	13	3	100000	13	9	11	100000	100000
4	100000	100000	100000	100000	7	8	100000	10	10	8	100000	100000
100000	100000	100000	100000	100000	100000	14	100000	100000	15	100000	11	100000
100000	100000	100000	100000	100000	100000	100000	100000	100000	12	100000	100000	100000
100000	100000	100000	100000	100000	100000	13	100000	100000	6	11	100000	100000
12	100000	100000	100000	100000	100000	8	100000	100000	2	8	100000	100000
100000	100000	100000	100000	100000	100000	100000	8	100000	100000	100000	5	100000
100000	100000	100000	100000	100000	100000	100000	100000	100000	100000	100000	6	100000
100000	100000	100000	100000	100000	100000	100000	11	100000	100000	100000	3	100000
100000	10	8	13	10	100000	100000	100000	9	100000	100000	100000	9
100000	5	6	3	8	100000	100000	100000	100000	100000	100000	100000	5
100000	10	8	9	12	100000	100000	100000	100000	100000	100000	100000	2
100000	13	10	13	13	100000	100000	100000	13	100000	100000	100000	7
13	13	11	100000	11	100000	100000	100000	6	100000	100000	100000	13
100000	5	8	8	6	100000	100000	100000	100000	100000	100000	100000	14
100000	2	2	4	2	100000	100000	100000	14	100000	100000	100000	8
100000	5	3	7	6	100000	100000	100000	13	100000	100000	100000	4
100000	100000	13	14	100000	100000	100000	100000	100000	100000	100000	100000	7
4	100000	100000	100000	14	10	11	100000	5	14	13	100000	100000
9	100000	100000	100000	100000	4	14	100000	11	100000	11	100000	100000
10	100000	100000	100000	100000	2	14	100000	13	15	10	100000	100000
8	12	10	14	8	100000	13	100000	4	100000	100000	100000	13
100000	14	100000	100000	14	100000	100000	13	100000	100000	100000	100000	100000
15	12	13	14	10	100000	12	14	100000	100000	100000	100000	100000
9	11	11	14	8	100000	7	100000	11	100000	100000	100000	100000
5	13	13	100000	10	14	8	100000	7	14	100000	100000	100000
100000	8	11	11	8	100000	100000	100000	100000	100000	100000	100000	100000
14	6	7	9	3	100000	13	100000	12	100000	100000	100000	13
12	7	6	9	3	100000	13	100000	9	100000	100000	100000	10
13	7	6	10	6	100000	100000	100000	8	100000	100000	100000	8
9	15	13	100000	11	13	14	100000	0	100000	100000	100000	100000
100000	100000	100000	100000	100000	14	100000	100000	6	100000	100000	100000	100000
15	100000	100000	100000	100000	11	100000	100000	10	100000	100000	100000	100000
100000	100000	100000	100000	100000	100000	100000	100000	100000	100000	100000	100000	100000
15	100000	100000	100000	100000	12	13	100000	100000	7	8	100000	100000
100000	100000	100000	100000	100000	100000	100000	3	100000	100000	100000	13	100000
100000	100000	100000	100000	100000	100000	13	5	100000	100000	100000	14	100000
100000	100000	100000	100000	100000	100000	9	9	100000	100000	100000	100000	100000

Tabla A.8: Distancia de los nodos de demanda a los sitios candidatos.

66	67	68	69	70	71
100000	100000	100000	100000	100000	15
100000	100000	100000	100000	100000	100000
15	100000	100000	100000	100000	100000
10	100000	100000	100000	100000	100000
7	100000	100000	100000	100000	100000
13	100000	100000	100000	100000	100000
100000	100000	100000	100000	100000	100000
100000	100000	100000	100000	100000	13
13	100000	100000	100000	100000	7
11	100000	100000	100000	100000	11
7	15	100000	100000	100000	13
4	100000	100000	100000	100000	100000
100000	100000	100000	100000	100000	100000
100000	100000	100000	100000	100000	100000
100000	100000	100000	100000	100000	100000
14	100000	100000	100000	100000	100000
100000	100000	100000	100000	100000	100000
100000	100000	100000	100000	100000	100000
100000	100000	100000	100000	100000	100000
14	3	4	6	10	100000
100000	11	9	8	8	15
100000	10	9	9	2	100000
100000	8	8	10	8	100000
12	5	6	9	14	100000
100000	12	10	8	100000	7
100000	7	5	3	11	10
100000	5	4	3	7	14
100000	100000	14	14	4	100000
2	11	12	13	100000	14
6	100000	100000	100000	100000	100000
8	100000	100000	100000	100000	100000
7	5	6	8	100000	11
100000	100000	100000	100000	100000	7
100000	100000	15	13	100000	2
10	12	11	10	100000	6
6	9	9	10	100000	10
100000	14	13	11	100000	4
14	9	8	7	100000	5
11	5	4	3	13	9
12	1	1	3	11	13
6	7	9	11	100000	14
13	13	14	100000	100000	100000
12	100000	100000	100000	100000	100000
100000	100000	100000	100000	100000	100000
100000	100000	100000	100000	100000	100000
100000	100000	100000	100000	100000	14
100000	100000	100000	100000	100000	11
100000	100000	100000	100000	100000	9

Tabla A.9: Distancia de los nodos de demanda a los sitios candidatos.

Apéndice B

Algoritmo escrito en GAMS

A continuación se presentará el código escrito en GAMS que se utilizó para resolver el caso de estudio.

```
*-----  
*Descomposición de Benders  
*-----  
*Inicialización  
Scalar Zsup 'Cota superior' /INF/;  
Scalar Zinf 'Cota inferior' /-INF/;  
Solve maestroinicial minimizing zmi using mip;  
Display ``Valores iniciales'', x.l, w.l;  
*Algoritmo de Benders  
Scalar converge /0/;  
Scalar iteracion;  
Scalar cota;  
loop(iter$(not converge),  
*Resolver subproblema  
solve subproblema maximazing zsd using mip;  
Display ``Solución al subproblema'', zsd.l;  
*Resultados  
abort$(subproblema.modelstat = 4)  
``El subproblema dual es infactible''
```

```

*Subproblema no acotado if(subproblema.modelstat = 3, solve
subproblemamodificado maximizing zsd using mip;
*Revisar resultados
abort$(subproblemamodificado.modelstat >= 2)
``El problema modificado no alcanza la optimalidad``;
*Agregar corte de factibilidad al maestro
conjcortefac(iter)=yes;
else
*El problema es acotado
cota=sum(j, f(j)*X.l(j))+sum(j, a(j)*W.l(j))+zsd.l;
if (cota < Zsup,
Zsup=cota;
);
*Agregar corte de optimalidad
conjcorte(iter)=yes;
);
*Constantes de Cortes
constcorteu1(iter,i) = u1.l(i);
constcorteu2(iter,j) = u2.l(j);
constcorteu3(iter,i) = u3.l(i);
constcorteu4(iter,i,j) = u4.l(i,j);
constcorteu5(iter,i,j) = u5.l(i,j);
*Resolver el problema maestro
solve maestro minimizing zm using mip;
*Resultados
abort$(maestro.modelstat=4)
``El problema maestro es infactible``;
*Actualizar cota inferior
Zinf=zm.l;
iteracion=ord(iter);
Display iteracion, Zinf, Zsup, zm.l;
converge$((Zsup-Zinf)<1)=1;
Display$converge ``El problema converge``;
);
abort$(not converge) ``El problema no converge``;
*Solucion
*original.solprint=2;

```

```
solve original minimizing z using mip;  
Display ``Solución'', z.l, w.l, x.l ,wo.l, xo.l, yo.l,  
zm.l,  
rspd.m;
```

Apéndice C

Captura de pantalla de GAMS

```

GAMS 24.7.1 r56632 Released Mar 14, 2016 WEX-WEI x86 64bit/MS Windows 02/20/20
General Algebraic Modeling System
Execution

---- 450 Valores iniciales

---- 450 VARIABLE X.L
10 1.000, 15 1.000, 19 1.000, 38 1.000

---- 450 VARIABLE W.L
10 83.000, 15 1.000, 19 1.000, 38 1.000

---- 461 Solución al subproblema
VARIABLE zsd.L = 1950289.000 objetivo subproblema
dual

---- 461 VARIABLE X.L
10 1.000, 15 1.000, 19 1.000, 38 1.000

---- 461 VARIABLE W.L
10 83.000, 15 1.000, 19 1.000, 38 1.000

GAMS 24.7.1 r56632 Released Mar 14, 2016 WEX-WEI x86 64bit/MS Windows 02/20/20
General Algebraic Modeling System
Model Analysis SOLVE maestro Using MIP From line 493

****
**** 71 Integer +INF Bounds have been reset to 100 (see Option IntVarUp)
****
GAMS 24.7.1 r56632 Released Mar 14, 2016 WEX-WEI x86 64bit/MS Windows 02/20/20
General Algebraic Modeling System
Execution

---- 508 PARAMETER iteracion = 1.000
PARAMETER Zinf = 6.986679E+7 Cota inferior

```

```

PARAMETER Zsup = 8.072131E+7 Cota superior
VARIABLE zm.L = 6.986679E+7 objetivo maestro
---- 461 Solución al subproblema
VARIABLE zsd.L = 1700265.000 objetivo subproblema
dual

---- 461 VARIABLE X.L
52 1.000, 59 1.000, 62 1.000, 64 1.000

---- 461 VARIABLE W.L
52 1.000, 59 83.000, 62 1.000, 64 1.000

GAMS 24.7.1 r56632 Released Mar 14, 2016 WEX-WEI x86 64bit/MS Windows 02/20/20
General Algebraic Modeling System
Model Analysis SOLVE maestro Using MIP From line 493

****
**** 71 Integer +INF Bounds have been reset to 100 (see Option IntVarUp)
****
GAMS 24.7.1 r56632 Released Mar 14, 2016 WEX-WEI x86 64bit/MS Windows 02/20/20
General Algebraic Modeling System
Execution

---- 508 PARAMETER iteracion = 2.000
PARAMETER Zinf = 7.312734E+7 Cota inferior
PARAMETER Zsup = 8.053234E+7 Cota superior
VARIABLE zm.L = 7.312734E+7 objetivo maestro
GAMS 24.7.1 r56632 Released Mar 14, 2016 WEX-WEI x86 64bit/MS Windows 02/20/20
General Algebraic Modeling System
Model Analysis SOLVE subproblema Using MIP From line 460

**** ERRORS/WARNINGS IN VARIABLE u4(2,21)
1 error(s): Unbounded variable
GAMS 24.7.1 r56632 Released Mar 14, 2016 WEX-WEI x86 64bit/MS Windows 02/20/20
General Algebraic Modeling System
Execution

```

Figura C.1: Captura de pantalla del algoritmo en GAMS

```

---- 461 Solución al subproblema
      VARIABLE zsd.L          = 8300217.462 objetivo subproblema
                                dual

---- 461 VARIABLE X.L

30 1.000,   52 1.000,   62 1.000,   64 1.000

---- 461 VARIABLE W.L

21 19.000,   30 1.000,   38 63.000,   52 1.000,   62 1.000,   64 1.000

GAMS 24.7.1 r56632 Released Mar 14, 2016 WEX-WEI x86 64bit/MS Windows 02/20/20
General Algebraic Modeling System
Model Analysis SOLVE maestro Using MIP From line 493

****
**** 71 Integer +INF Bounds have been reset to 100 (see Option IntVarUp)
****
GAMS 24.7.1 r56632 Released Mar 14, 2016 WEX-WEI x86 64bit/MS Windows 02/20/20
General Algebraic Modeling System
Execution

---- 508 PARAMETER iteracion      = 3.000
      PARAMETER Zinf              = 7.525954E+7 Cota inferior
      PARAMETER Zsup              = 8.053234E+7 Cota superior
      VARIABLE zm.L               = 7.525954E+7 objetivo maestro

GAMS 24.7.1 r56632 Released Mar 14, 2016 WEX-WEI x86 64bit/MS Windows 02/20/20
General Algebraic Modeling System
Model Analysis SOLVE subproblema Using MIP From line 460

**** ERRORS/WARNINGS IN VARIABLE u4(30,67)
      1 error(s): Unbounded variable
GAMS 24.7.1 r56632 Released Mar 14, 2016 WEX-WEI x86 64bit/MS Windows 02/20/20
General Algebraic Modeling System
Execution

---- 461 Solución al subproblema
      VARIABLE zsd.L          = 8300212.355 objetivo subproblema

```

```

dual

---- 461 VARIABLE X.L

24 1.000,   37 1.000,   40 1.000,   50 1.000

---- 461 VARIABLE W.L

24 1.000,   37 1.000,   40 1.000,   50 1.000,   67 82.000

GAMS 24.7.1 r56632 Released Mar 14, 2016 WEX-WEI x86 64bit/MS Windows 02/20/20
General Algebraic Modeling System
Model Analysis SOLVE maestro Using MIP From line 493

****
**** 71 Integer +INF Bounds have been reset to 100 (see Option IntVarUp)
****
GAMS 24.7.1 r56632 Released Mar 14, 2016 WEX-WEI x86 64bit/MS Windows 02/20/20
General Algebraic Modeling System
Execution

---- 508 PARAMETER iteracion      = 4.000
      PARAMETER Zinf              = 7.590641E+7 Cota inferior
      PARAMETER Zsup              = 8.053234E+7 Cota superior
      VARIABLE zm.L               = 7.590641E+7 objetivo maestro

GAMS 24.7.1 r56632 Released Mar 14, 2016 WEX-WEI x86 64bit/MS Windows 02/20/20
General Algebraic Modeling System
Model Analysis SOLVE subproblema Using MIP From line 460

**** ERRORS/WARNINGS IN VARIABLE u4(2,68)
      1 error(s): Unbounded variable
GAMS 24.7.1 r56632 Released Mar 14, 2016 WEX-WEI x86 64bit/MS Windows 02/20/20
General Algebraic Modeling System
Execution

---- 461 Solución al subproblema
      VARIABLE zsd.L          = 8300379.886 objetivo subproblema
                                dual

```

Figura C.2: Captura de pantalla del algoritmo en GAMS

```

---- 461 VARIABLE X.L
15 1.000, 40 1.000, 60 1.000, 67 1.000

---- 461 VARIABLE W.L
15 1.000, 40 1.000, 60 1.000, 67 1.000, 68 82.000

GAMS 24.7.1 r56632 Released Mar 14, 2016 WEX-WEI x86 64bit/MS Windows 02/20/20
General Algebraic Modeling System
Model Analysis SOLVE maestro Using MIP From line 493

****
**** 71 Integer +INF Bounds have been reset to 100 (see Option IntVarUp)
****
GAMS 24.7.1 r56632 Released Mar 14, 2016 WEX-WEI x86 64bit/MS Windows 02/20/20
General Algebraic Modeling System
Execution

---- 508 PARAMETER iteracion = 5.000
PARAMETER Zinf = 7.619058E+7 Cota inferior
PARAMETER Zsup = 8.053234E+7 Cota superior
VARIABLE zm.L = 7.619058E+7 objetivo maestro
GAMS 24.7.1 r56632 Released Mar 14, 2016 WEX-WEI x86 64bit/MS Windows 02/20/20
General Algebraic Modeling System
Model Analysis SOLVE subproblema Using MIP From line 460

**** ERRORS/WARNINGS IN VARIABLE u4(2,69)
1 error(s): Unbounded variable
GAMS 24.7.1 r56632 Released Mar 14, 2016 WEX-WEI x86 64bit/MS Windows 02/20/20
General Algebraic Modeling System
Execution

---- 461 Solución al subproblema
VARIABLE zsd.L = 8300150.905 objetivo subproblema
dual

---- 461 VARIABLE X.L

```

```

5 1.000, 48 1.000, 58 1.000, 67 1.000

---- 461 VARIABLE W.L
5 1.000, 48 1.000, 58 1.000, 67 1.000, 69 82.000

GAMS 24.7.1 r56632 Released Mar 14, 2016 WEX-WEI x86 64bit/MS Windows 02/20/20
General Algebraic Modeling System
Model Analysis SOLVE maestro Using MIP From line 493

****
**** 71 Integer +INF Bounds have been reset to 100 (see Option IntVarUp)
****
GAMS 24.7.1 r56632 Released Mar 14, 2016 WEX-WEI x86 64bit/MS Windows 02/20/20
General Algebraic Modeling System
Execution

---- 508 PARAMETER iteracion = 6.000
PARAMETER Zinf = 7.621292E+7 Cota inferior
PARAMETER Zsup = 8.053234E+7 Cota superior
VARIABLE zm.L = 7.621292E+7 objetivo maestro
GAMS 24.7.1 r56632 Released Mar 14, 2016 WEX-WEI x86 64bit/MS Windows 02/20/20
General Algebraic Modeling System
Model Analysis SOLVE subproblema Using MIP From line 460

**** ERRORS/WARNINGS IN VARIABLE u4(10,70)
1 error(s): Unbounded variable
GAMS 24.7.1 r56632 Released Mar 14, 2016 WEX-WEI x86 64bit/MS Windows 02/20/20
General Algebraic Modeling System
Execution

---- 461 Solución al subproblema
VARIABLE zsd.L = 8300115.902 objetivo subproblema
dual

---- 461 VARIABLE X.L
17 1.000, 38 1.000, 40 1.000, 67 1.000

```

Figura C.3: Captura de pantalla del algoritmo en GAMS

```

---- 461 VARIABLE W.L

17 1.000, 38 1.000, 40 1.000, 67 1.000, 70 82.000

GAMS 24.7.1 r56632 Released Mar 14, 2016 WEX-WEI x86 64bit/MS Windows 02/20/20
General Algebraic Modeling System
Model Analysis SOLVE maestro Using MIP From line 493

****
**** 71 Integer +INF Bounds have been reset to 100 (see Option IntVarUp)
****
GAMS 24.7.1 r56632 Released Mar 14, 2016 WEX-WEI x86 64bit/MS Windows 02/20/20
General Algebraic Modeling System
Execution

---- 508 PARAMETER iteracion = 7.000
PARAMETER Zinf = 7.659028E+7 Cota inferior
PARAMETER Zsup = 8.053234E+7 Cota superior
VARIABLE zm.L = 7.659028E+7 objetivo maestro
GAMS 24.7.1 r56632 Released Mar 14, 2016 WEX-WEI x86 64bit/MS Windows 02/20/20
General Algebraic Modeling System
Model Analysis SOLVE subproblema Using MIP From line 460

**** ERRORS/WARNINGS IN VARIABLE u4(2,1)
1 error(s): Unbounded variable
GAMS 24.7.1 r56632 Released Mar 14, 2016 WEX-WEI x86 64bit/MS Windows 02/20/20
General Algebraic Modeling System
Execution

---- 461 Solución al subproblema
VARIABLE zsd.L = 8100113.833 objetivo subproblema
dual

---- 461 VARIABLE X.L

47 1.000, 48 1.000, 59 1.000, 63 1.000

---- 461 VARIABLE W.L

```

```

1 82.000, 47 1.000, 48 1.000, 59 1.000, 63 1.000

GAMS 24.7.1 r56632 Released Mar 14, 2016 WEX-WEI x86 64bit/MS Windows 02/20/20
General Algebraic Modeling System
Model Analysis SOLVE maestro Using MIP From line 493

****
**** 71 Integer +INF Bounds have been reset to 100 (see Option IntVarUp)
****
GAMS 24.7.1 r56632 Released Mar 14, 2016 WEX-WEI x86 64bit/MS Windows 02/20/20
General Algebraic Modeling System
Execution

---- 508 PARAMETER iteracion = 8.000
PARAMETER Zinf = 7.449100E+7 Cota inferior
PARAMETER Zsup = 8.053234E+7 Cota superior
VARIABLE zm.L = 7.449100E+7 objetivo maestro
GAMS 24.7.1 r56632 Released Mar 14, 2016 WEX-WEI x86 64bit/MS Windows 02/20/20
General Algebraic Modeling System
Model Analysis SOLVE subproblema Using MIP From line 460

**** ERRORS/WARNINGS IN VARIABLE u4(2,3)
1 error(s): Unbounded variable
GAMS 24.7.1 r56632 Released Mar 14, 2016 WEX-WEI x86 64bit/MS Windows 02/20/20
General Algebraic Modeling System
Execution

---- 461 Solución al subproblema
VARIABLE zsd.L = 8300230.376 objetivo subproblema
dual

---- 461 VARIABLE X.L

21 1.000, 24 1.000, 52 1.000, 64 1.000

---- 461 VARIABLE W.L

3 82.000, 21 1.000, 24 1.000, 52 1.000, 64 1.000

```

Figura C.4: Captura de pantalla del algoritmo en GAMS

```

GAMS 24.7.1 r56632 Released Mar 14, 2016 WEX-WEI x86 64bit/MS Windows 02/20/20
General Algebraic Modeling System
Model Analysis SOLVE maestro Using MIP From line 493

****
**** 71 Integer +INF Bounds have been reset to 100 (see Option IntVarUp)
****
GAMS 24.7.1 r56632 Released Mar 14, 2016 WEX-WEI x86 64bit/MS Windows 02/20/20
General Algebraic Modeling System
Execution

---- 508 PARAMETER iteracion = 9.000
PARAMETER Zinf = 7.936531E+7 Cota inferior
PARAMETER Zsup = 8.053234E+7 Cota superior
VARIABLE zm.L = 7.936531E+7 objetivo maestro
GAMS 24.7.1 r56632 Released Mar 14, 2016 WEX-WEI x86 64bit/MS Windows 02/20/20
General Algebraic Modeling System
Model Analysis SOLVE subproblema Using MIP From line 460

**** ERRORS/WARNINGS IN VARIABLE u4(2,4)
1 error(s): Unbounded variable
GAMS 24.7.1 r56632 Released Mar 14, 2016 WEX-WEI x86 64bit/MS Windows 02/20/20
General Algebraic Modeling System
Execution

---- 461 Solución al subproblema
VARIABLE zsd.L = 8300171.257 objetivo subproblema
dual

---- 461 VARIABLE X.L
30 1.000, 50 1.000, 51 1.000, 64 1.000

---- 461 VARIABLE W.L
4 82.000, 30 1.000, 50 1.000, 51 1.000, 64 1.000
GAMS 24.7.1 r56632 Released Mar 14, 2016 WEX-WEI x86 64bit/MS Windows 02/20/20
General Algebraic Modeling System

```

```

Model Analysis SOLVE maestro Using MIP From line 493

****
**** 71 Integer +INF Bounds have been reset to 100 (see Option IntVarUp)
****
GAMS 24.7.1 r56632 Released Mar 14, 2016 WEX-WEI x86 64bit/MS Windows 02/20/20
General Algebraic Modeling System
Execution

---- 508 PARAMETER iteracion = 10.000
PARAMETER Zinf = 7.626625E+7 Cota inferior
PARAMETER Zsup = 8.053234E+7 Cota superior
VARIABLE zm.L = 7.626625E+7 objetivo maestro
GAMS 24.7.1 r56632 Released Mar 14, 2016 WEX-WEI x86 64bit/MS Windows 02/20/20
General Algebraic Modeling System
Model Analysis SOLVE subproblema Using MIP From line 460

**** ERRORS/WARNINGS IN VARIABLE u4(2,5)
1 error(s): Unbounded variable
GAMS 24.7.1 r56632 Released Mar 14, 2016 WEX-WEI x86 64bit/MS Windows 02/20/20
General Algebraic Modeling System
Execution

---- 461 Solución al subproblema
VARIABLE zsd.L = 8300164.305 objetivo subproblema
dual

---- 461 VARIABLE X.L
9 1.000, 47 1.000, 49 1.000, 70 1.000

---- 461 VARIABLE W.L
5 82.000, 9 1.000, 47 1.000, 49 1.000, 70 1.000
GAMS 24.7.1 r56632 Released Mar 14, 2016 WEX-WEI x86 64bit/MS Windows 02/20/20
General Algebraic Modeling System
Model Analysis SOLVE maestro Using MIP From line 493

```

Figura C.5: Captura de pantalla del algoritmo en GAMS

```

**** 71 Integer +INF Bounds have been reset to 100 (see Option IntVarUp)
****
GAMS 24.7.1 r56632 Released Mar 14, 2016 WEX-WEI x86 64bit/MS Windows 02/20/20
General Algebraic Modeling System
Execution

---- 508 PARAMETER iteracion          =      11.000
      PARAMETER Zinf                  = 7.640939E+7 Cota inferior
      PARAMETER Zsup                  = 8.053234E+7 Cota superior
      VARIABLE zm.L                   = 7.640939E+7 objetivo maestro
GAMS 24.7.1 r56632 Released Mar 14, 2016 WEX-WEI x86 64bit/MS Windows 02/20/20
General Algebraic Modeling System
Model Analysis SOLVE subproblema Using MIP From line 460

**** ERRORS/WARNINGS IN VARIABLE u4(2,28)
1 error(s): Unbounded variable
GAMS 24.7.1 r56632 Released Mar 14, 2016 WEX-WEI x86 64bit/MS Windows 02/20/20
General Algebraic Modeling System
Execution

---- 461 Solución al subproblema
      VARIABLE zsd.L                  = 8300163.214 objetivo subproblema
                                          dual

---- 461 VARIABLE X.L
9 1.000, 47 1.000, 49 1.000, 70 1.000

---- 461 VARIABLE W.L
9 1.000, 47 1.000, 49 1.000, 65 82.000, 70 1.000

GAMS 24.7.1 r56632 Released Mar 14, 2016 WEX-WEI x86 64bit/MS Windows 02/20/20
General Algebraic Modeling System
Model Analysis SOLVE maestro Using MIP From line 493

****
**** 71 Integer +INF Bounds have been reset to 100 (see Option IntVarUp)
****

```

```

GAMS 24.7.1 r56632 Released Mar 14, 2016 WEX-WEI x86 64bit/MS Windows 02/20/20
General Algebraic Modeling System
Execution

---- 508 PARAMETER iteracion          =      12.000
      PARAMETER Zinf                  = 7.615639E+7 Cota inferior
      PARAMETER Zsup                  = 8.053234E+7 Cota superior
      VARIABLE zm.L                   = 7.615639E+7 objetivo maestro
GAMS 24.7.1 r56632 Released Mar 14, 2016 WEX-WEI x86 64bit/MS Windows 02/20/20
General Algebraic Modeling System
Model Analysis SOLVE subproblema Using MIP From line 460

**** ERRORS/WARNINGS IN VARIABLE u4(2,6)
1 error(s): Unbounded variable
GAMS 24.7.1 r56632 Released Mar 14, 2016 WEX-WEI x86 64bit/MS Windows 02/20/20
General Algebraic Modeling System
Execution

---- 461 Solución al subproblema
      VARIABLE zsd.L                  = 8300148.933 objetivo subproblema
                                          dual

---- 461 VARIABLE X.L
30 1.000, 58 1.000, 69 1.000, 71 1.000

---- 461 VARIABLE W.L
6 82.000, 30 1.000, 58 1.000, 69 1.000, 71 1.000

GAMS 24.7.1 r56632 Released Mar 14, 2016 WEX-WEI x86 64bit/MS Windows 02/20/20
General Algebraic Modeling System
Model Analysis SOLVE maestro Using MIP From line 493

****
**** 71 Integer +INF Bounds have been reset to 100 (see Option IntVarUp)
****
GAMS 24.7.1 r56632 Released Mar 14, 2016 WEX-WEI x86 64bit/MS Windows 02/20/20
General Algebraic Modeling System

```

Figura C.6: Captura de pantalla del algoritmo en GAMS

```

Execution

---- 508 PARAMETER iteracion          =      13.000
      PARAMETER Zinf                  = 7.668197E+7 Cota inferior
      PARAMETER Zsup                  = 8.053234E+7 Cota superior
      VARIABLE zm.L                   = 7.668197E+7 objetivo maestro
GAMS 24.7.1 r56632 Released Mar 14, 2016 WEX-WEI x86 64bit/MS Windows 02/20/20
General Algebraic Modeling System
Model Analysis SOLVE subproblema Using MIP From line 460

**** ERRORS/WARNINGS IN VARIABLE u4(2,7)
1 error(s): Unbounded variable
GAMS 24.7.1 r56632 Released Mar 14, 2016 WEX-WEI x86 64bit/MS Windows 02/20/20
General Algebraic Modeling System
Execution

---- 461 Solución al subproblema
      VARIABLE zsd.L                  = 8300101.657 objetivo subproblema
                                         dual

---- 461 VARIABLE X.L
6 1.000, 15 1.000, 21 1.000, 58 1.000

---- 461 VARIABLE W.L
6 1.000, 7 82.000, 15 1.000, 21 1.000, 58 1.000

GAMS 24.7.1 r56632 Released Mar 14, 2016 WEX-WEI x86 64bit/MS Windows 02/20/20
General Algebraic Modeling System
Model Analysis SOLVE maestro Using MIP From line 493

****
**** 71 Integer +INF Bounds have been reset to 100 (see Option IntVarUp)
****
GAMS 24.7.1 r56632 Released Mar 14, 2016 WEX-WEI x86 64bit/MS Windows 02/20/20
General Algebraic Modeling System
Execution

```

```

---- 508 PARAMETER iteracion          =      14.000
      PARAMETER Zinf                  = 7.600180E+7 Cota inferior
      PARAMETER Zsup                  = 8.053234E+7 Cota superior
      VARIABLE zm.L                   = 7.600180E+7 objetivo maestro
GAMS 24.7.1 r56632 Released Mar 14, 2016 WEX-WEI x86 64bit/MS Windows 02/20/20
General Algebraic Modeling System
Model Analysis SOLVE subproblema Using MIP From line 460

**** ERRORS/WARNINGS IN VARIABLE u4(2,8)
1 error(s): Unbounded variable
GAMS 24.7.1 r56632 Released Mar 14, 2016 WEX-WEI x86 64bit/MS Windows 02/20/20
General Algebraic Modeling System
Execution

---- 461 Solución al subproblema
      VARIABLE zsd.L                  = 8300153.476 objetivo subproblema
                                         dual

---- 461 VARIABLE X.L
3 1.000, 28 1.000, 30 1.000, 49 1.000

---- 461 VARIABLE W.L
3 1.000, 8 82.000, 28 1.000, 30 1.000, 49 1.000

GAMS 24.7.1 r56632 Released Mar 14, 2016 WEX-WEI x86 64bit/MS Windows 02/20/20
General Algebraic Modeling System
Model Analysis SOLVE maestro Using MIP From line 493

****
**** 71 Integer +INF Bounds have been reset to 100 (see Option IntVarUp)
****
GAMS 24.7.1 r56632 Released Mar 14, 2016 WEX-WEI x86 64bit/MS Windows 02/20/20
General Algebraic Modeling System
Execution

---- 508 PARAMETER iteracion          =      15.000
      PARAMETER Zinf                  = 7.650393E+7 Cota inferior

```

Figura C.7: Captura de pantalla del algoritmo en GAMS

```

PARAMETER Zsup          = 8.053234E+7  Cota superior
VARIABLE zm.L          = 7.650393E+7  objetivo maestro
GAMS 24.7.1 r56632 Released Mar 14, 2016 WEX-WEI x86 64bit/MS Windows 02/20/20
General Algebraic Modeling System
Model Analysis SOLVE subproblema Using MIP From line 460

**** ERRORS/WARNINGS IN VARIABLE u4(2,17)
1 error(s): Unbounded variable
GAMS 24.7.1 r56632 Released Mar 14, 2016 WEX-WEI x86 64bit/MS Windows 02/20/20
General Algebraic Modeling System
Execution

---- 461 Solución al subproblema
      VARIABLE zsd.L          = 8300162.748  objetivo subproblema
                                dual

---- 461 VARIABLE X.L
5 1.000, 46 1.000, 62 1.000, 64 1.000

---- 461 VARIABLE W.L
5 1.000, 9 82.000, 46 1.000, 62 1.000, 64 1.000

GAMS 24.7.1 r56632 Released Mar 14, 2016 WEX-WEI x86 64bit/MS Windows 02/20/20
General Algebraic Modeling System
Model Analysis SOLVE maestro Using MIP From line 493

****
**** 71 Integer +INF Bounds have been reset to 100 (see Option IntVarUp)
****
GAMS 24.7.1 r56632 Released Mar 14, 2016 WEX-WEI x86 64bit/MS Windows 02/20/20
General Algebraic Modeling System
Execution

---- 508 PARAMETER iteracion          = 16.000
      PARAMETER Zinf                = 7.981281E+7  Cota inferior
      PARAMETER Zsup                = 8.053234E+7  Cota superior
      VARIABLE zm.L                = 7.981281E+7  objetivo maestro

```

```

GAMS 24.7.1 r56632 Released Mar 14, 2016 WEX-WEI x86 64bit/MS Windows 02/20/20
General Algebraic Modeling System
Model Analysis SOLVE subproblema Using MIP From line 460

**** ERRORS/WARNINGS IN VARIABLE u4(2,25)
1 error(s): Unbounded variable
GAMS 24.7.1 r56632 Released Mar 14, 2016 WEX-WEI x86 64bit/MS Windows 02/20/20
General Algebraic Modeling System
Execution

---- 461 Solución al subproblema
      VARIABLE zsd.L          = 8300131.571  objetivo subproblema
                                dual

---- 461 VARIABLE X.L
5 1.000, 46 1.000, 62 1.000, 64 1.000

---- 461 VARIABLE W.L
5 1.000, 25 82.000, 46 1.000, 62 1.000, 64 1.000

GAMS 24.7.1 r56632 Released Mar 14, 2016 WEX-WEI x86 64bit/MS Windows 02/20/20
General Algebraic Modeling System
Model Analysis SOLVE maestro Using MIP From line 493

****
**** 71 Integer +INF Bounds have been reset to 100 (see Option IntVarUp)
****
GAMS 24.7.1 r56632 Released Mar 14, 2016 WEX-WEI x86 64bit/MS Windows 02/20/20
General Algebraic Modeling System
Execution

---- 508 PARAMETER iteracion          = 17.000
      PARAMETER Zinf                = 8.037776E+7  Cota inferior
      PARAMETER Zsup                = 8.053234E+7  Cota superior
      VARIABLE zm.L                = 8.037776E+7  objetivo maestro
GAMS 24.7.1 r56632 Released Mar 14, 2016 WEX-WEI x86 64bit/MS Windows 02/20/20
General Algebraic Modeling System

```

Figura C.8: Captura de pantalla del algoritmo en GAMS

```

**** ERRORS/WARNINGS IN VARIABLE u4(2,10)
      1 error(s): Unbounded variable
GAMS 24.7.1 r56632 Released Mar 14, 2016 WEX-WEI x86 64bit/MS Windows 02/20/20
General Algebraic Modeling System
Execution

---- 461 Solución al subproblema
      VARIABLE zsd.L          = 8300275.212  objetivo subproblema
                                   dual

---- 461 VARIABLE X.L
8 1.000, 25 1.000, 48 1.000, 58 1.000

---- 461 VARIABLE W.L
8 1.000, 10 82.000, 25 1.000, 48 1.000, 58 1.000

GAMS 24.7.1 r56632 Released Mar 14, 2016 WEX-WEI x86 64bit/MS Windows 02/20/20
General Algebraic Modeling System
Model Analysis SOLVE maestro Using MIP From line 493

****
**** 71 Integer +INF Bounds have been reset to 100 (see Option IntVarUp)
****
GAMS 24.7.1 r56632 Released Mar 14, 2016 WEX-WEI x86 64bit/MS Windows 02/20/20
General Algebraic Modeling System
Execution

---- 508 PARAMETER iteracion = 18.000
      PARAMETER Zinf = 7.613423E+7 Cota inferior
      PARAMETER Zsup = 8.053234E+7 Cota superior
      VARIABLE zm.L = 7.613423E+7 objetivo maestro

GAMS 24.7.1 r56632 Released Mar 14, 2016 WEX-WEI x86 64bit/MS Windows 02/20/20
General Algebraic Modeling System
Model Analysis SOLVE subproblema Using MIP From line 460

**** ERRORS/WARNINGS IN VARIABLE u4(2,11)
      1 error(s): Unbounded variable

```

```

GAMS 24.7.1 r56632 Released Mar 14, 2016 WEX-WEI x86 64bit/MS Windows 02/20/20
General Algebraic Modeling System
Execution

---- 461 Solución al subproblema
      VARIABLE zsd.L          = 8300171.119  objetivo subproblema
                                   dual

---- 461 VARIABLE X.L
9 1.000, 46 1.000, 65 1.000, 70 1.000

---- 461 VARIABLE W.L
9 1.000, 11 82.000, 46 1.000, 65 1.000, 70 1.000

GAMS 24.7.1 r56632 Released Mar 14, 2016 WEX-WEI x86 64bit/MS Windows 02/20/20
General Algebraic Modeling System
Model Analysis SOLVE maestro Using MIP From line 493

****
**** 71 Integer +INF Bounds have been reset to 100 (see Option IntVarUp)
****
GAMS 24.7.1 r56632 Released Mar 14, 2016 WEX-WEI x86 64bit/MS Windows 02/20/20
General Algebraic Modeling System
Execution

---- 508 PARAMETER iteracion = 19.000
      PARAMETER Zinf = 7.654385E+7 Cota inferior
      PARAMETER Zsup = 8.053234E+7 Cota superior
      VARIABLE zm.L = 7.654385E+7 objetivo maestro

GAMS 24.7.1 r56632 Released Mar 14, 2016 WEX-WEI x86 64bit/MS Windows 02/20/20
General Algebraic Modeling System
Model Analysis SOLVE subproblema Using MIP From line 460

**** ERRORS/WARNINGS IN VARIABLE u4(2,27)
      1 error(s): Unbounded variable
GAMS 24.7.1 r56632 Released Mar 14, 2016 WEX-WEI x86 64bit/MS Windows 02/20/20
General Algebraic Modeling System

```

Figura C.9: Captura de pantalla del algoritmo en GAMS

```

Execution

---- 461 Solución al subproblema
      VARIABLE zsd.L          = 8300168.624  objetivo subproblema
                                dual

---- 461 VARIABLE X.L

9 1.000, 46 1.000, 65 1.000, 70 1.000

---- 461 VARIABLE W.L

9 1.000, 27 82.000, 46 1.000, 65 1.000, 70 1.000

GAMS 24.7.1 r56632 Released Mar 14, 2016 WEX-WEI x86 64bit/MS Windows 02/20/20
General Algebraic Modeling System
Model Analysis SOLVE maestro Using MIP From line 493

****
**** 71 Integer +INF Bounds have been reset to 100 (see Option IntVarUp)
****
GAMS 24.7.1 r56632 Released Mar 14, 2016 WEX-WEI x86 64bit/MS Windows 02/20/20
General Algebraic Modeling System
Execution

---- 508 PARAMETER iteracion          = 20.000
      PARAMETER Zinf                 = 7.613414E+7  Cota inferior
      PARAMETER Zsup                 = 8.053234E+7  Cota superior
      VARIABLE zm.L                  = 7.613414E+7  objetivo maestro

GAMS 24.7.1 r56632 Released Mar 14, 2016 WEX-WEI x86 64bit/MS Windows 02/20/20
General Algebraic Modeling System
Model Analysis SOLVE subproblema Using MIP From line 460

**** ERRORS/WARNINGS IN VARIABLE u4(2,71)
      1 error(s): Unbounded variable

GAMS 24.7.1 r56632 Released Mar 14, 2016 WEX-WEI x86 64bit/MS Windows 02/20/20
General Algebraic Modeling System
Execution

```

```

---- 461 Solución al subproblema
      VARIABLE zsd.L          = 8300164.433  objetivo subproblema
                                dual

---- 461 VARIABLE X.L

9 1.000, 46 1.000, 65 1.000, 70 1.000

---- 461 VARIABLE W.L

9 1.000, 46 1.000, 65 1.000, 70 1.000, 71 82.000

GAMS 24.7.1 r56632 Released Mar 14, 2016 WEX-WEI x86 64bit/MS Windows 02/20/20
General Algebraic Modeling System
Model Analysis SOLVE maestro Using MIP From line 493

****
**** 71 Integer +INF Bounds have been reset to 100 (see Option IntVarUp)
****
GAMS 24.7.1 r56632 Released Mar 14, 2016 WEX-WEI x86 64bit/MS Windows 02/20/20
General Algebraic Modeling System
Execution

---- 508 PARAMETER iteracion          = 21.000
      PARAMETER Zinf                 = 8.046189E+7  Cota inferior
      PARAMETER Zsup                 = 8.053234E+7  Cota superior
      VARIABLE zm.L                  = 8.046189E+7  objetivo maestro

GAMS 24.7.1 r56632 Released Mar 14, 2016 WEX-WEI x86 64bit/MS Windows 02/20/20
General Algebraic Modeling System
Model Analysis SOLVE subproblema Using MIP From line 460

**** ERRORS/WARNINGS IN VARIABLE u4(2,61)
      1 error(s): Unbounded variable

GAMS 24.7.1 r56632 Released Mar 14, 2016 WEX-WEI x86 64bit/MS Windows 02/20/20
General Algebraic Modeling System
Execution

---- 461 Solución al subproblema
      VARIABLE zsd.L          = 8300323.979  objetivo subproblema

```

Figura C.10: Captura de pantalla del algoritmo en GAMS

```

---- 461 VARIABLE X.L
9 1.000, 46 1.000, 65 1.000, 70 1.000

---- 461 VARIABLE W.L
9 1.000, 46 1.000, 61 82.000, 65 1.000, 70 1.000

GAMS 24.7.1 r56632 Released Mar 14, 2016 WEX-WEI x86 64bit/MS Windows 02/20/20
General Algebraic Modeling System
Model Analysis SOLVE maestro Using MIP From line 493

****
**** 71 Integer +INF Bounds have been reset to 100 (see Option IntVarUp)
****
GAMS 24.7.1 r56632 Released Mar 14, 2016 WEX-WEI x86 64bit/MS Windows 02/20/20
General Algebraic Modeling System
Execution

---- 508 PARAMETER iteracion = 22.000
PARAMETER Zinf = 7.646066E+7 Cota inferior
PARAMETER Zsup = 8.053234E+7 Cota superior
VARIABLE zm.L = 7.646066E+7 objetivo maestro
GAMS 24.7.1 r56632 Released Mar 14, 2016 WEX-WEI x86 64bit/MS Windows 02/20/20
General Algebraic Modeling System
Model Analysis SOLVE subproblema Using MIP From line 460

**** ERRORS/WARNINGS IN VARIABLE u4(2,41)
1 error(s): Unbounded variable
GAMS 24.7.1 r56632 Released Mar 14, 2016 WEX-WEI x86 64bit/MS Windows 02/20/20
General Algebraic Modeling System
Execution

---- 461 Solución al subproblema
VARIABLE zsd.L = 8300174.024 objetivo subproblema
dual

---- 461 VARIABLE X.L

```

```

9 1.000, 46 1.000, 65 1.000, 70 1.000

---- 461 VARIABLE W.L
9 1.000, 41 82.000, 46 1.000, 65 1.000, 70 1.000

GAMS 24.7.1 r56632 Released Mar 14, 2016 WEX-WEI x86 64bit/MS Windows 02/20/20
General Algebraic Modeling System
Model Analysis SOLVE maestro Using MIP From line 493

****
**** 71 Integer +INF Bounds have been reset to 100 (see Option IntVarUp)
****
GAMS 24.7.1 r56632 Released Mar 14, 2016 WEX-WEI x86 64bit/MS Windows 02/20/20
General Algebraic Modeling System
Execution

---- 508 PARAMETER iteracion = 23.000
PARAMETER Zinf = 7.637713E+7 Cota inferior
PARAMETER Zsup = 8.053234E+7 Cota superior
VARIABLE zm.L = 7.637713E+7 objetivo maestro
GAMS 24.7.1 r56632 Released Mar 14, 2016 WEX-WEI x86 64bit/MS Windows 02/20/20
General Algebraic Modeling System
Model Analysis SOLVE subproblema Using MIP From line 460

**** ERRORS/WARNINGS IN VARIABLE u4(2,5)
1 error(s): Unbounded variable
GAMS 24.7.1 r56632 Released Mar 14, 2016 WEX-WEI x86 64bit/MS Windows 02/20/20
General Algebraic Modeling System
Execution

---- 461 Solución al subproblema
VARIABLE zsd.L = 8300157.638 objetivo subproblema
dual

---- 461 VARIABLE X.L
6 1.000, 30 1.000, 41 1.000, 46 1.000

```

Figura C.11: Captura de pantalla del algoritmo en GAMS

```

---- 461 VARIABLE W.L

6 1.000, 12 82.000, 30 1.000, 41 1.000, 46 1.000

GAMS 24.7.1 r56632 Released Mar 14, 2016 WEX-WEI x86 64bit/MS Windows 02/20/20
General Algebraic Modeling System
Model Analysis SOLVE maestro Using MIP From line 493

****
**** 71 Integer +INF Bounds have been reset to 100 (see Option IntVarUp)
****
GAMS 24.7.1 r56632 Released Mar 14, 2016 WEX-WEI x86 64bit/MS Windows 02/20/20
General Algebraic Modeling System
Execution

---- 508 PARAMETER iteracion = 24.000
PARAMETER Zinf = 7.591411E+7 Cota inferior
PARAMETER Zsup = 8.053234E+7 Cota superior
VARIABLE zm.L = 7.591411E+7 objetivo maestro
GAMS 24.7.1 r56632 Released Mar 14, 2016 WEX-WEI x86 64bit/MS Windows 02/20/20
General Algebraic Modeling System
Model Analysis SOLVE subproblema Using MIP From line 460

**** ERRORS/WARNINGS IN VARIABLE u4(2,13)
1 error(s): Unbounded variable
GAMS 24.7.1 r56632 Released Mar 14, 2016 WEX-WEI x86 64bit/MS Windows 02/20/20
General Algebraic Modeling System
Execution

---- 461 Solución al subproblema
VARIABLE zsd.L = 8300167.029 objetivo subproblema
dual

---- 461 VARIABLE X.L

11 1.000, 27 1.000, 64 1.000, 65 1.000

---- 461 VARIABLE W.L

```

```

11 1.000, 13 82.000, 27 1.000, 64 1.000, 65 1.000

GAMS 24.7.1 r56632 Released Mar 14, 2016 WEX-WEI x86 64bit/MS Windows 02/20/20
General Algebraic Modeling System
Model Analysis SOLVE maestro Using MIP From line 493

****
**** 71 Integer +INF Bounds have been reset to 100 (see Option IntVarUp)
****
GAMS 24.7.1 r56632 Released Mar 14, 2016 WEX-WEI x86 64bit/MS Windows 02/20/20
General Algebraic Modeling System
Execution

---- 508 PARAMETER iteracion = 25.000
PARAMETER Zinf = 7.602827E+7 Cota inferior
PARAMETER Zsup = 8.053234E+7 Cota superior
VARIABLE zm.L = 7.602827E+7 objetivo maestro
GAMS 24.7.1 r56632 Released Mar 14, 2016 WEX-WEI x86 64bit/MS Windows 02/20/20
General Algebraic Modeling System
Model Analysis SOLVE subproblema Using MIP From line 460

**** ERRORS/WARNINGS IN VARIABLE u4(2,14)
1 error(s): Unbounded variable
GAMS 24.7.1 r56632 Released Mar 14, 2016 WEX-WEI x86 64bit/MS Windows 02/20/20
General Algebraic Modeling System
Execution

---- 461 Solución al subproblema
VARIABLE zsd.L = 8300272.905 objetivo subproblema
dual

---- 461 VARIABLE X.L

7 1.000, 40 1.000, 53 1.000, 68 1.000

---- 461 VARIABLE W.L

7 1.000, 14 82.000, 40 1.000, 53 1.000, 68 1.000

```

Figura C.12: Captura de pantalla del algoritmo en GAMS

```

GAMS 24.7.1 r56632 Released Mar 14, 2016 WEX-WEI x86 64bit/MS Windows 02/20/20
General Algebraic Modeling System
Model Analysis SOLVE maestro Using MIP From line 493

****
**** 71 Integer +INF Bounds have been reset to 100 (see Option IntVarUp)
****
GAMS 24.7.1 r56632 Released Mar 14, 2016 WEX-WEI x86 64bit/MS Windows 02/20/20
General Algebraic Modeling System
Execution

---- 508 PARAMETER iteracion = 26.000
PARAMETER Zinf = 8.007241E+7 Cota inferior
PARAMETER Zsup = 8.053234E+7 Cota superior
VARIABLE zm.L = 8.007241E+7 objetivo maestro
GAMS 24.7.1 r56632 Released Mar 14, 2016 WEX-WEI x86 64bit/MS Windows 02/20/20
General Algebraic Modeling System
Model Analysis SOLVE subproblema Using MIP From line 460

**** ERRORS/WARNINGS IN VARIABLE u4(2,15)
1 error(s): Unbounded variable
GAMS 24.7.1 r56632 Released Mar 14, 2016 WEX-WEI x86 64bit/MS Windows 02/20/20
General Algebraic Modeling System
Execution

---- 461 Solución al subproblema
VARIABLE zsd.L = 8300336.702 objetivo subproblema
dual

---- 461 VARIABLE X.L
8 1.000, 9 1.000, 46 1.000, 48 1.000

---- 461 VARIABLE W.L
8 1.000, 9 1.000, 15 82.000, 46 1.000, 48 1.000
GAMS 24.7.1 r56632 Released Mar 14, 2016 WEX-WEI x86 64bit/MS Windows 02/20/20
General Algebraic Modeling System

```

```

Model Analysis SOLVE maestro Using MIP From line 493

****
**** 71 Integer +INF Bounds have been reset to 100 (see Option IntVarUp)
****
GAMS 24.7.1 r56632 Released Mar 14, 2016 WEX-WEI x86 64bit/MS Windows 02/20/20
General Algebraic Modeling System
Execution

---- 508 PARAMETER iteracion = 27.000
PARAMETER Zinf = 7.632525E+7 Cota inferior
PARAMETER Zsup = 8.053234E+7 Cota superior
VARIABLE zm.L = 7.632525E+7 objetivo maestro
GAMS 24.7.1 r56632 Released Mar 14, 2016 WEX-WEI x86 64bit/MS Windows 02/20/20
General Algebraic Modeling System
Model Analysis SOLVE subproblema Using MIP From line 460

**** ERRORS/WARNINGS IN VARIABLE u4(2,31)
1 error(s): Unbounded variable
GAMS 24.7.1 r56632 Released Mar 14, 2016 WEX-WEI x86 64bit/MS Windows 02/20/20
General Algebraic Modeling System
Execution

---- 461 Solución al subproblema
VARIABLE zsd.L = 8300158.602 objetivo subproblema
dual

---- 461 VARIABLE X.L
8 1.000, 9 1.000, 46 1.000, 48 1.000

---- 461 VARIABLE W.L
8 1.000, 9 1.000, 31 82.000, 46 1.000, 48 1.000
GAMS 24.7.1 r56632 Released Mar 14, 2016 WEX-WEI x86 64bit/MS Windows 02/20/20
General Algebraic Modeling System
Model Analysis SOLVE maestro Using MIP From line 493

```

Figura C.13: Captura de pantalla del algoritmo en GAMS

```

**** 71 Integer +INF Bounds have been reset to 100 (see Option IntVarUp)
****
GAMS 24.7.1 r56632 Released Mar 14, 2016 WEX-WEI x86 64bit/MS Windows 02/20/20
General Algebraic Modeling System
Execution

---- 508 PARAMETER iteracion          =      28.000
      PARAMETER Zinf                  = 7.606997E+7 Cota inferior
      PARAMETER Zsup                  = 8.053234E+7 Cota superior
      VARIABLE zm.L                   = 7.606997E+7 objetivo maestro
GAMS 24.7.1 r56632 Released Mar 14, 2016 WEX-WEI x86 64bit/MS Windows 02/20/20
General Algebraic Modeling System
Model Analysis SOLVE subproblema Using MIP From line 460

**** ERRORS/WARNINGS IN VARIABLE u4(2,16)
      1 error(s): Unbounded variable
GAMS 24.7.1 r56632 Released Mar 14, 2016 WEX-WEI x86 64bit/MS Windows 02/20/20
General Algebraic Modeling System
Execution

---- 461 Solución al subproblema
      VARIABLE zsd.L                  = 8300170.710 objetivo subproblema
                                          dual

---- 461 VARIABLE X.L
10 1.000,  27 1.000,  45 1.000,  60 1.000

12 1.000,  17 1.000,  46 1.000,  49 1.000

---- 461 VARIABLE W.L
10 1.000,  16 82.000,  17 1.000,  46 1.000,  49 1.000

GAMS 24.7.1 r56632 Released Mar 14, 2016 WEX-WEI x86 64bit/MS Windows 02/20/20
General Algebraic Modeling System
Model Analysis SOLVE maestro Using MIP From line 493

****
**** 71 Integer +INF Bounds have been reset to 100 (see Option IntVarUp)
****

```

```

GAMS 24.7.1 r56632 Released Mar 14, 2016 WEX-WEI x86 64bit/MS Windows 02/20/20
General Algebraic Modeling System
Execution

---- 508 PARAMETER iteracion          =      29.000
      PARAMETER Zinf                  = 7.623693E+7 Cota inferior
      PARAMETER Zsup                  = 8.053234E+7 Cota superior
      VARIABLE zm.L                   = 7.623693E+7 objetivo maestro
GAMS 24.7.1 r56632 Released Mar 14, 2016 WEX-WEI x86 64bit/MS Windows 02/20/20
General Algebraic Modeling System
Model Analysis SOLVE subproblema Using MIP From line 460

**** ERRORS/WARNINGS IN VARIABLE u4(2,17)
      1 error(s): Unbounded variable
GAMS 24.7.1 r56632 Released Mar 14, 2016 WEX-WEI x86 64bit/MS Windows 02/20/20
General Algebraic Modeling System
Execution

---- 461 Solución al subproblema
      VARIABLE zsd.L                  = 8300164.712 objetivo subproblema
                                          dual

---- 461 VARIABLE X.L
10 1.000,  27 1.000,  45 1.000,  60 1.000

---- 461 VARIABLE W.L
10 1.000,  17 82.000,  27 1.000,  45 1.000,  60 1.000

GAMS 24.7.1 r56632 Released Mar 14, 2016 WEX-WEI x86 64bit/MS Windows 02/20/20
General Algebraic Modeling System
Model Analysis SOLVE maestro Using MIP From line 493

****
**** 71 Integer +INF Bounds have been reset to 100 (see Option IntVarUp)
****
GAMS 24.7.1 r56632 Released Mar 14, 2016 WEX-WEI x86 64bit/MS Windows 02/20/20
General Algebraic Modeling System

```

Figura C.14: Captura de pantalla del algoritmo en GAMS

```

Execution

---- 508 PARAMETER iteracion          =      30.000
      PARAMETER Zinf                  =  7.919637E+7  Cota inferior
      PARAMETER Zsup                  =  8.053234E+7  Cota superior
      VARIABLE zm.L                   =  7.919637E+7  objetivo maestro
GAMS 24.7.1 r56632 Released Mar 14, 2016 WEX-WEI x86 64bit/MS Windows 02/20/20
General Algebraic Modeling System
Model Analysis SOLVE subproblema Using MIP From line 460

**** ERRORS/WARNINGS IN VARIABLE u4(1,33)
1 error(s): Unbounded variable
GAMS 24.7.1 r56632 Released Mar 14, 2016 WEX-WEI x86 64bit/MS Windows 02/20/20
General Algebraic Modeling System
Execution

---- 461 Solución al subproblema
      VARIABLE zsd.L                  =  8300268.083  objetivo subproblema
                                          dual

---- 461 VARIABLE X.L
10 1.000,   27 1.000,   45 1.000,   60 1.000

---- 461 VARIABLE W.L
10 1.000,   27 1.000,   33 82.000,   45 1.000,   60 1.000

GAMS 24.7.1 r56632 Released Mar 14, 2016 WEX-WEI x86 64bit/MS Windows 02/20/20
General Algebraic Modeling System
Model Analysis SOLVE maestro Using MIP From line 493

****
**** 71 Integer +INF Bounds have been reset to 100 (see Option IntVarUp)
****
GAMS 24.7.1 r56632 Released Mar 14, 2016 WEX-WEI x86 64bit/MS Windows 02/20/20
General Algebraic Modeling System
Execution

```

```

---- 508 PARAMETER iteracion          =      31.000
      PARAMETER Zinf                  =  8.016069E+7  Cota inferior
      PARAMETER Zsup                  =  8.053234E+7  Cota superior
      VARIABLE zm.L                   =  8.016069E+7  objetivo maestro
GAMS 24.7.1 r56632 Released Mar 14, 2016 WEX-WEI x86 64bit/MS Windows 02/20/20
General Algebraic Modeling System
Model Analysis SOLVE subproblema Using MIP From line 460

**** ERRORS/WARNINGS IN VARIABLE u4(2,47)
1 error(s): Unbounded variable
GAMS 24.7.1 r56632 Released Mar 14, 2016 WEX-WEI x86 64bit/MS Windows 02/20/20
General Algebraic Modeling System
Execution

---- 461 Solución al subproblema
      VARIABLE zsd.L                  =  8300463.057  objetivo subproblema
                                          dual

---- 461 VARIABLE X.L
10 1.000,   27 1.000,   45 1.000,   60 1.000

---- 461 VARIABLE W.L
10 1.000,   27 1.000,   45 1.000,   47 82.000,   60 1.000

GAMS 24.7.1 r56632 Released Mar 14, 2016 WEX-WEI x86 64bit/MS Windows 02/20/20
General Algebraic Modeling System
Model Analysis SOLVE maestro Using MIP From line 493

****
**** 71 Integer +INF Bounds have been reset to 100 (see Option IntVarUp)
****
GAMS 24.7.1 r56632 Released Mar 14, 2016 WEX-WEI x86 64bit/MS Windows 02/20/20
General Algebraic Modeling System
Execution

---- 508 PARAMETER iteracion          =      32.000
      PARAMETER Zinf                  =  7.638027E+7  Cota inferior

```

Figura C.15: Captura de pantalla del algoritmo en GAMS

```

PARAMETER Zsup          = 8.053234E+7  Cota superior
VARIABLE zm.L          = 7.638027E+7  objetivo maestro
GAMS 24.7.1 r56632 Released Mar 14, 2016 WEX-WEI x86 64bit/MS Windows 02/20/20
General Algebraic Modeling System
Model Analysis SOLVE subproblema Using MIP From line 460

**** ERRORS/WARNINGS IN VARIABLE u4(2,13)
1 error(s): Unbounded variable
GAMS 24.7.1 r56632 Released Mar 14, 2016 WEX-WEI x86 64bit/MS Windows 02/20/20
General Algebraic Modeling System
Execution

---- 461 Solución al subproblema
      VARIABLE zsd.L          = 8300457.960  objetivo subproblema
                                dual
---- 461 VARIABLE X.L
10 1.000, 27 1.000, 45 1.000, 60 1.000
---- 461 VARIABLE W.L
10 1.000, 20 82.000, 27 1.000, 45 1.000, 60 1.000
GAMS 24.7.1 r56632 Released Mar 14, 2016 WEX-WEI x86 64bit/MS Windows 02/20/20
General Algebraic Modeling System
Model Analysis SOLVE maestro Using MIP From line 493

****
**** 71 Integer +INF Bounds have been reset to 100 (see Option IntVarUp)
****
GAMS 24.7.1 r56632 Released Mar 14, 2016 WEX-WEI x86 64bit/MS Windows 02/20/20
General Algebraic Modeling System
Execution

---- 508 PARAMETER iteracion          = 33.000
      PARAMETER Zinf                  = 7.697657E+7  Cota inferior
      PARAMETER Zsup                  = 8.053234E+7  Cota superior
      VARIABLE zm.L                   = 7.697657E+7  objetivo maestro

```

```

GAMS 24.7.1 r56632 Released Mar 14, 2016 WEX-WEI x86 64bit/MS Windows 02/20/20
General Algebraic Modeling System
Model Analysis SOLVE subproblema Using MIP From line 460

**** ERRORS/WARNINGS IN VARIABLE u4(2,18)
1 error(s): Unbounded variable
GAMS 24.7.1 r56632 Released Mar 14, 2016 WEX-WEI x86 64bit/MS Windows 02/20/20
General Algebraic Modeling System
Execution

---- 461 Solución al subproblema
      VARIABLE zsd.L          = 8300172.155  objetivo subproblema
                                dual
---- 461 VARIABLE X.L
4 1.000, 10 1.000, 13 1.000, 68 1.000
---- 461 VARIABLE W.L
4 1.000, 10 1.000, 13 1.000, 18 82.000, 68 1.000
GAMS 24.7.1 r56632 Released Mar 14, 2016 WEX-WEI x86 64bit/MS Windows 02/20/20
General Algebraic Modeling System
Model Analysis SOLVE maestro Using MIP From line 493

****
**** 71 Integer +INF Bounds have been reset to 100 (see Option IntVarUp)
****
GAMS 24.7.1 r56632 Released Mar 14, 2016 WEX-WEI x86 64bit/MS Windows 02/20/20
General Algebraic Modeling System
Execution

---- 508 PARAMETER iteracion          = 34.000
      PARAMETER Zinf                  = 7.666207E+7  Cota inferior
      PARAMETER Zsup                  = 8.053234E+7  Cota superior
      VARIABLE zm.L                   = 7.666207E+7  objetivo maestro
GAMS 24.7.1 r56632 Released Mar 14, 2016 WEX-WEI x86 64bit/MS Windows 02/20/20
General Algebraic Modeling System

```

Figura C.16: Captura de pantalla del algoritmo en GAMS

```

Model Analysis      SOLVE subproblema Using MIP From line 460

**** ERRORS/WARNINGS IN VARIABLE u4(2,16)
      1 error(s): Unbounded variable
GAMS 24.7.1 r56632 Released Mar 14, 2016 WEX-WEI x86 64bit/MS Windows 02/20/20
General Algebraic Modeling System
Execution

---- 461 Solución al subproblema
      VARIABLE zsd.L          = 8300455.371  objetivo subproblema
                                dual

---- 461 VARIABLE X.L
4 1.000, 10 1.000, 13 1.000, 68 1.000

---- 461 VARIABLE W.L
4 1.000, 10 1.000, 13 1.000, 34 82.000, 68 1.000

GAMS 24.7.1 r56632 Released Mar 14, 2016 WEX-WEI x86 64bit/MS Windows 02/20/20
General Algebraic Modeling System
Model Analysis      SOLVE maestro Using MIP From line 493

****
**** 71 Integer +INF Bounds have been reset to 100 (see Option IntVarUp)
****
GAMS 24.7.1 r56632 Released Mar 14, 2016 WEX-WEI x86 64bit/MS Windows 02/20/20
General Algebraic Modeling System
Execution

---- 508 PARAMETER iteracion      = 35.000
      PARAMETER Zinf              = 7.669720E+7  Cota inferior
      PARAMETER Zsup              = 8.053234E+7  Cota superior
      VARIABLE zm.L               = 7.669720E+7  objetivo maestro
---- 461 Solución al subproblema
      VARIABLE zsd.L              = 2500204.500  objetivo subproblema
                                dual

```

```

---- 461 VARIABLE X.L
4 1.000, 10 1.000, 13 1.000, 68 1.000

---- 461 VARIABLE W.L
4 1.000, 10 1.000, 13 1.000, 68 83.000

GAMS 24.7.1 r56632 Released Mar 14, 2016 WEX-WEI x86 64bit/MS Windows 02/20/20
General Algebraic Modeling System
Model Analysis      SOLVE maestro Using MIP From line 493

****
**** 71 Integer +INF Bounds have been reset to 100 (see Option IntVarUp)
****
GAMS 24.7.1 r56632 Released Mar 14, 2016 WEX-WEI x86 64bit/MS Windows 02/20/20
General Algebraic Modeling System
Execution

---- 508 PARAMETER iteracion      = 36.000
      PARAMETER Zinf              = 7.312742E+7  Cota inferior
      PARAMETER Zsup              = 8.053234E+7  Cota superior
      VARIABLE zm.L               = 7.312742E+7  objetivo maestro
GAMS 24.7.1 r56632 Released Mar 14, 2016 WEX-WEI x86 64bit/MS Windows 02/20/20
General Algebraic Modeling System
Model Analysis      SOLVE subproblema Using MIP From line 460

**** ERRORS/WARNINGS IN VARIABLE u4(2,10)
      1 error(s): Unbounded variable
GAMS 24.7.1 r56632 Released Mar 14, 2016 WEX-WEI x86 64bit/MS Windows 02/20/20
General Algebraic Modeling System
Execution

---- 461 Solución al subproblema
      VARIABLE zsd.L              = 7600248.638  objetivo subproblema
                                dual

---- 461 VARIABLE X.L

```

Figura C.17: Captura de pantalla del algoritmo en GAMS

```

30 1.000,    52 1.000,    62 1.000,    64 1.000

---- 461 VARIABLE W.L

16 40.000,    21 19.000,    30 1.000,    38 23.000,    52 1.000,    62 1.000
64 1.000

GAMS 24.7.1 r56632 Released Mar 14, 2016 WEX-WEI x86 64bit/MS Windows 02/20/20
General Algebraic Modeling System
Model Analysis SOLVE maestro Using MIP From line 493

****
**** 71 Integer +INF Bounds have been reset to 100 (see Option IntVarUp)
****
GAMS 24.7.1 r56632 Released Mar 14, 2016 WEX-WEI x86 64bit/MS Windows 02/20/20
General Algebraic Modeling System
Execution

---- 508 PARAMETER iteracion          =          37.000
PARAMETER Zinf                      = 7.314337E+7 Cota inferior
PARAMETER Zsup                      = 8.053234E+7 Cota superior
VARIABLE zm.L                       = 7.314337E+7 objetivo maestro
GAMS 24.7.1 r56632 Released Mar 14, 2016 WEX-WEI x86 64bit/MS Windows 02/20/20
General Algebraic Modeling System
Model Analysis SOLVE subproblema Using MIP From line 460

**** ERRORS/WARNINGS IN VARIABLE u4(10,25)
1 error(s): Unbounded variable
GAMS 24.7.1 r56632 Released Mar 14, 2016 WEX-WEI x86 64bit/MS Windows 02/20/20
General Algebraic Modeling System
Execution

---- 461 Solución al subproblema
VARIABLE zsd.L                      = 7166485.350 objetivo subproblema
                                         dual

---- 461 VARIABLE X.L

30 1.000,    52 1.000,    62 1.000,    64 1.000

```

```

---- 461 VARIABLE W.L

16 3.000,    30 20.000,    38 42.000,    52 1.000,    62 1.000,    64 1.000
68 18.000

GAMS 24.7.1 r56632 Released Mar 14, 2016 WEX-WEI x86 64bit/MS Windows 02/20/20
General Algebraic Modeling System
Model Analysis SOLVE maestro Using MIP From line 493

****
**** 71 Integer +INF Bounds have been reset to 100 (see Option IntVarUp)
****
GAMS 24.7.1 r56632 Released Mar 14, 2016 WEX-WEI x86 64bit/MS Windows 02/20/20
General Algebraic Modeling System
Execution

---- 508 PARAMETER iteracion          =          38.000
PARAMETER Zinf                      = 8.072385E+7 Cota inferior
PARAMETER Zsup                      = 8.053234E+7 Cota superior
VARIABLE zm.L                       = 8.072385E+7 objetivo maestro

---- 512 El problema converge
GAMS 24.7.1 r56632 Released Mar 14, 2016 WEX-WEI x86 64bit/MS Windows 02/20/20
General Algebraic Modeling System
Equation Listing SOLVE sub Using MIP From line 518

```

Figura C.18: Captura de pantalla del algoritmo en GAMS

```

---- 527 VARIABLE W.L
4 1.000, 10 83.000, 13 1.000, 68 1.000

---- 527 VARIABLE X.L
4 1.000, 10 1.000, 13 1.000, 68 1.000

---- 527 VARIABLE Y.L
      4      10      13      68
1
2      1.000
3      1.000
4      1.000
5      1.000
6      1.000
7      1.000
8      1.000
9
10      1.000
11      1.000
12      1.000
13      1.000
14      1.000
15      1.000
16      1.000
17      1.000
18      1.000
19      1.000
20      1.000
21      0.500 0.500
22      1.000
23      1.000
24      1.000
25      1.000
26      1.000
27      1.000

28      1.000
29      1.000
30      1.000
31      1.000
32      1.000
33      1.000
34      1.000
35      1.000
36      1.000
37      1.000
38      1.000
39      1.000
40      1.000
41      1.000
42      1.000
43      1.000
44      1.000
45      1.000
46      1.000
47      1.000
48      1.000

```

Figura C.19: Captura de pantalla del algoritmo en GAMS