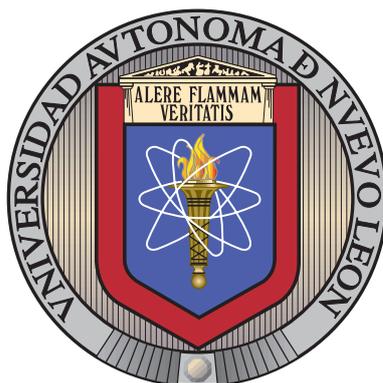


UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN

FACULTAD DE CIENCIAS FÍSICO-MATEMÁTICAS

CENTRO DE INVESTIGACIÓN EN CIENCIAS FÍSICO-MATEMÁTICAS



EL PROBLEMA  $(r|p)$ –*Centroide Negativo*  
CONSIDERANDO LEALTAD DE LOS CLIENTES

UNA TESIS PRESENTADA POR:

JUAN CARLOS GARCÍA VÉLEZ

COMO REQUISITO PARCIAL PARA OBTENER EL GRADO DE:

MAESTRÍA EN CIENCIAS  
CON ORIENTACIÓN EN MATEMÁTICAS

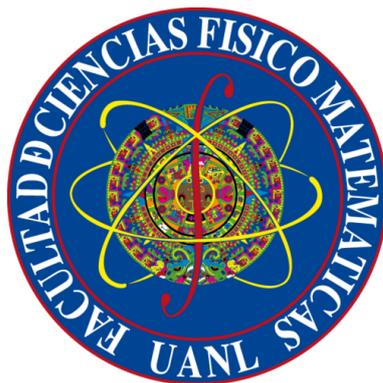
SAN NICOLÁS DE LOS GARZA, NUEVO LEÓN

ENERO 2019

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN

FACULTAD DE CIENCIAS FÍSICO-MATEMÁTICAS

CENTRO DE INVESTIGACIÓN EN CIENCIAS FÍSICO-MATEMÁTICAS



EL PROBLEMA  $(r|p)$ –*Centroide Negativo*  
CONSIDERANDO LEALTAD DE LOS CLIENTES

UNA TESIS PRESENTADA POR:

JUAN CARLOS GARCÍA VÉLEZ

COMO REQUISITO PARCIAL PARA OBTENER EL GRADO DE:

MAESTRÍA EN CIENCIAS

CON ORIENTACIÓN EN MATEMÁTICAS

SAN NICOLÁS DE LOS GARZA, NUEVO LEÓN

ENERO 2019

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN  
FACULTAD DE CIENCIAS FÍSICO-MATEMÁTICAS  
CENTRO DE INVESTIGACIÓN EN CIENCIAS FÍSICO-MATEMÁTICAS

Los miembros del comité de Tesis, por este medio, certifican que han leído la tesis “El Problema  $(r|p)$ –*Centroide Negativo* Considerando Lealtad de los Clientes” presentada por el **Lic. Juan Carlos García Vélez** y que es totalmente adecuada en alcance y calidad como un requisito parcial para obtener el grado de Maestría en Ciencias con Orientación en Matemáticas.

El Comité de Tesis

---

**Dr. José Fernando Camacho Vallejo**

Asesor

---

**Dr. Juan Antonio Díaz García**

Co-asesor

---

**Dr. Diego Rafael Ruiz-Hernández**

Co-asesor

---

**Dr. Omar Jorge Ibarra Rojas**

Sinodal

Vo. Bo.

---

**Dr. José Fernando Camacho Vallejo**

Coordinador del Posgrado en Ciencias  
con Orientación en Matemáticas

*“Hasta el amanecer.”*

*LyL Pro Game Team.*

# DEDICATORIA

---

*En memoria de Felicitas Hernández<sup>†</sup>,  
Gerardo García<sup>†</sup>  
e Inocencia Rojas<sup>†</sup>.  
Estarán por siempre en mi corazón.*

# AGRADECIMIENTOS

---

Esta tesis si bien ha requerido de esfuerzo y mucha dedicación por parte mía y del director de tesis, no hubiese sido posible su finalización sin la cooperación desinteresada de todas y cada una de las personas que a continuación mencionaré y muchas de las cuales han sido un soporte muy fuerte en momentos de angustia y desesperación. Gracias por brindarme su ayuda, sus conocimientos y su apoyo.

Dicen que la mejor herencia que nos pueden dejar los padres son los estudios, sin embargo, no creo que sea el único legado del cual yo particularmente me siento muy agradecido, mis padres me han permitido trazar mi camino y caminar con mis propios pies. Ellos son mis pilares de la vida, les dedico este trabajo de titulación. Gracias Leticia Vélez y Carlos García, padres queridos.

Quiero agradecer especialmente a mi asesor, el Dr. Fernando Camacho por darme la oportunidad de trabajar con él y ser mi asesor, por guiarme en mi desarrollo académico, por el esfuerzo y el apoyo que me ha brindado. También me gustaría darle las gracias por su paciencia. Sin su apoyo este trabajo no hubiera sido posible. Pero no solamente quiero agradecer por este período de tesis, sino por lo que antecede y trasciende a este período. Fue mi profesor en la licenciatura y desde ahí sentí un respeto y admiración profunda a su labor. Desde ahí me di cuenta de la magia que conlleva dedicarse a la optimización e investigación de operaciones. Por su apoyo y confianza en mi trabajo, por su paciencia y su capacidad para guiar, no solamente en el desarrollo de esta tesis, sino también en mi formación como

investigador; por todas las cosas nuevas que aprendí y que no olvidaré. Por sus comentarios, sugerencias, consejos y críticas que se ven reflejados en este trabajo y en mi persona porque sirvieron de motivación. Por escucharme y aconsejarme cuando lo necesité y por tenerlo siempre como un soporte. Es imposible describir en unas líneas el grado de agradecimiento que siento, así como de admiración por su labor docente, investigadora y calidad humana.

Les agradezco profundamente a mis co-asesores, el Dr. Juan Antonio Diaz por el tiempo que pudo dedicar a este tema de tesis, por sus sugerencias e ideas y por los nuevos conocimientos que me transmitió, lejos de eso quiero agradecerle por hacer de mí el enamorarme de la localización de instalaciones. Especialmente quiero agradecer al Dr. Diego Ruiz-Hernandez por la colaboración brindada en esta tesis y haberme recibido con los brazos abiertos en mis estancias de investigación en el Colegio Universitario de Estudios Financieros (CUNEF, Madrid) así como en la Management School Universidad de Sheffield, en Inglaterra. Por el tiempo que me pudo dedicar, por sus sugerencias e ideas y por los nuevos conocimientos que me transmitió, así como por todas las aventuras que vivimos juntos por Europa.

Gracias a mis revisores: Dr. Jorge Omar Ibarra por sus certeras observaciones y valiosas sugerencias en la construcción y mejora del trabajo y por todo su tiempo invertido en la revisión de esta tesis.

También les doy las gracias a las personas que hicieron mi estancia en España de lo más agradable: Dr. Jesús María Pinar, Dr. Roberto Morales, Keli Muasya, Giovvana e Ingrid García por su hospitalidad y amabilidad, hicieron mi estadía muy amena. En especial agradecido con mi española favorita Alba Navarro , por esas tardes en Madrid llenas de alegría.

Agradezco en general al Bi-level Group en especial a Lilian Lopez Vera quien ha sido mi colega favorita, por sus incontables consejos y opiniones, por cuidar de mí mientras estamos de congreso, sin duda alguna de esas

amistades que se conservan toda la vida. Agradezco a Héctor Maravillo y Martín Hernández por esas experiencias de congreso imperdibles y sus incontables sugerencias y opiniones de este presente trabajo. También quiero agradecer a mi compañera de maestría Irene Del Real por ser un soporte dentro y fuera de la escuela, muchas gracias por todo.

Quiero agradecer especialmente a todo el grupo LyL Pro Game Team, mis primos Pablo y Kevin García, Brandon y Rolando Peña, Angel y Julio Ibarra, Brayan y Miguel Espinoza y a mis amigos Iván Montalvo y Eliseo Cortes por sus ánimos de motivación hacia mi persona, sin duda amigos de toda la vida. Hasta el amanecer hoy y siempre.

Agradezco también a la familia García Hernández especialmente a mis tíos Antonio, Javier, Leticia, Manuel y Diana por sus consejos que me brindan día con día, en especial a mi abuelo Manuel García Rocha por el apoyo brindado todo este tiempo. Muchas gracias a mis tíos Leobardo Cortes, Rolando Peña, Miguel Espinoza, Maurilio Ibarra, Zulema Barrera, Reyna y Anselmo Vélez por sus incontables muestras de ánimo y apoyo hacia mi persona y mi familia.

Por último, pero no menos importante agradezco a la Universidad Autónoma de Nuevo León (UANL), a la Facultad de Ciencias Físico-Matemáticas (FCFM) y al Centro de Investigación en Ciencias Físico-Matemáticas (CICFIM) por todo el apoyo proporcionado durante este periodo de estudios. Además, es un placer para mí expresar mi gratitud al Consejo Nacional de Ciencia y Tecnología (CONACYT) por la beca de apoyo y la beca mixta para mis estancias de investigación en España e Inglaterra.

A todos ustedes, Muchas Gracias.

# RESUMEN

---

En el presente trabajo de tesis se analiza un modelo de programación bi-nivel para un problema de cierre de instalaciones competitivas considerando lealtad de los clientes. Tomar en cuenta la lealtad del cliente es de suma importancia debido a la competencia actual del mercado. El problema que analizamos es una variante del problema de localización de instalaciones competitivo  $(r|p)$ –Centroide, con la diferencia que ahora buscamos cerrar instalaciones operando en el mercado, es decir empresas que tienen tiendas abiertas brindando un servicio determinado. Suponemos además que una instalación puede abastecer la demanda de varios clientes, pero un cliente debe ser abastecido por solo una instalación y que los clientes mantengan cierto nivel de lealtad por ser servidos por la instalación de su preferencia.

El problema se formula como un modelo de programación bi-nivel donde el objetivo del líder y seguidor es capturar la demanda máxima, respectivamente. El objetivo del problema es cerrar un número conocido de instalaciones por cada tomador de decisiones de tal manera que maximicen la demanda capturada, pero considerando un radio de lealtad de los clientes.

Nosotros proponemos un algoritmo exacto basado en ramificación y acotamiento para resolver el problema de programación bi-nivel y comparamos los resultados con uno de enumeración completa. Se llevó a cabo experimentación computacional y se mostró que el algoritmo parece ser una buena forma de resolver el problema aquí estudiado, pues en todos los casos, el tiempo requerido es menor al del obtenido con la enumeración

completa, además de tener un ahorro en nodos considerable. No está de más mencionar que al tratarse de un algoritmo exacto, aseguramos que el problema se resuelve de manera óptima para todas las instancias.

# ÍNDICE GENERAL

---

<b>1. Introducción</b>	<b>1</b>
1.1. Descripción del problema . . . . .	1
1.2. Motivación y justificación . . . . .	2
1.3. Objetivo . . . . .	4
1.4. Metodología . . . . .	4
1.5. Estructura de la Tesis . . . . .	6
<b>2. Revisión de Literatura</b>	<b>8</b>
2.1. Localización de instalaciones competitivas. . . . .	8
2.2. Problemas de cierre de instalaciones. . . . .	16
<b>3. El problema <math>\overline{(r p)}^{\mathcal{L}}</math>–Centroide</b>	<b>21</b>
3.1. Programación Multi-nivel . . . . .	21
3.1.1. Programación Bi-nivel . . . . .	23
3.2. Descripción del problema . . . . .	25
3.3. Modelo matemático de programación tri-nivel para el problema $\overline{(r p)}$ –Centroide . . . . .	26

3.3.1.	Modelo matemático de programación bi-nivel para el problema $\overline{(r p)}$ –Centroide . . . . .	30
3.4.	El problema $\overline{(r/p)}^{\mathcal{L}}$ – centroide . . . . .	33
3.4.1.	Supuestos . . . . .	33
3.4.2.	Restricciones válidas de lealtad para los clientes . . . . .	34
3.4.3.	Modelo de programación bi-nivel para el problema $\overline{(r/p)}^{\mathcal{L}}$ – centroide . . . . .	36
3.5.	Restricciones de lealtad: un ejemplo ilustrativo. . . . .	39
<b>4.</b>	<b>Algoritmo de ramificación y acotamiento para resolver el <math>\overline{(r p)}^{\mathcal{L}}</math> –Centroide</b> . . . . .	<b>47</b>
4.1.	Algoritmo de ramificación y acotamiento adaptado para resolver el $\overline{(r p)}^{\mathcal{L}}$ –Centroide. . . . .	47
4.1.1.	Ejemplo ilustrativo. . . . .	51
<b>5.</b>	<b>Experimentación computacional</b> . . . . .	<b>61</b>
5.1.	Generación de instancias de prueba . . . . .	61
5.1.1.	Generador aleatorio (conjunto A) . . . . .	63
5.1.2.	Generador ordenado (conjunto B) . . . . .	64
5.1.2.1.	Modelo de la $p$ –Mediana . . . . .	65
5.1.2.2.	Modelo $(r X_p)$ –medianoide . . . . .	66
5.1.3.	Generador basado en la red de Swain (conjunto C) y en la red de Swain extendida (conjunto D) . . . . .	68
5.2.	Ambiente computacional . . . . .	70
5.3.	Resultados Numéricos . . . . .	71

---

5.3.1. Comparando el tiempo computacional requerido . . .	77
<b>6. Conclusiones y trabajo a futuro</b>	<b>82</b>
6.1. Conclusiones . . . . .	82
6.2. Trabajo futuro . . . . .	83
6.2.1. El problema $\overline{(r p)}^{\mathcal{L}}$ –Centroide estocástico. . . . .	84
6.2.2. El problema $\overline{(r p)}^{\mathcal{L}}$ –Centroide en varias dimensiones.	85
<b>A. Autobiografía</b>	<b>90</b>
A.1. Estancias de Investigación . . . . .	90
A.2. Conferencias . . . . .	91
<b>Bibliografía</b>	<b>92</b>

---

## CAPÍTULO 1

# INTRODUCCIÓN

---

En este capítulo se describe el problema que se aborda en esta tesis, es decir, un problema de cierre de instalaciones competitivas junto con la motivación, justificación y objetivo central de la misma. También se describe la metodología que realizamos para culminar con el trabajo de tesis. Por último mencionamos la estructura de la tesis.

### 1.1 DESCRIPCIÓN DEL PROBLEMA

La idea principal detrás de esta tesis es desarrollar y resolver un modelo matemático para cerrar algunas instalaciones de dos empresas competidoras. El objetivo es determinar, dado que conocemos donde están ubicadas las instalaciones de las empresas que están operando en el mercado y los clientes que están asignados a ellas para satisfacer su demanda, cuáles de esas instalaciones cerrar de tal manera que las empresas mantengan la máxima cantidad de demanda. Es evidente que los clientes asignados a una instalación que se cerró, tendrán que ser reasignados a otra instalación que permanezca abierta. Para hacer esto, consideramos un radio de lealtad hacia la empresa a la que actualmente están asignados.

Este tipo de problema lo podemos abordar mediante un modelo de programación tri-nivel, donde en el primer nivel una empresa dominante (A) en el mercado cierra sus instalaciones, en el segundo nivel la empresa con menor jerarquía (B) cierra sus instalaciones, y en un tercer nivel los clientes se asignan a la instalación que sea de su preferencia. Dado que

es bien sabido que los problemas de programación multi-nivel son muy difíciles de resolver (ver capítulo 3), nosotros agregamos un conjunto de índices auxiliar para reducirlo a un problema de programación bi-nivel, sin embargo, en esa situación los clientes se re-asignan a la instalación abierta más cercana.

Por lo que, como novedad de esta tesis, nosotros agregamos restricciones válidas para considerar la lealtad del cliente, definiendo un radio de lealtad para dicho cliente y que este no necesariamente se reasigne a la instalación abierta más cercana. Para ejemplificar, considere que un cliente está asignado a una instalación de la empresa A, y esta se cierra, dicho cliente buscará dentro de un radio predefinido otra instalación de la empresa A para satisfacer su demanda, incluso aunque haya una instalación más cerca de la empresa B. Este supuesto es natural, porque hay servicios ofrecidos por instalaciones para los cuales los clientes procuran no cambiar de empresa tan fácilmente, por ejemplo, las cuentas bancarias, el médico de cabecera, contratos de planes telefónicos.

## 1.2 MOTIVACIÓN Y JUSTIFICACIÓN

Los problemas de localización de instalaciones competitivas (CFL, por sus siglas en inglés) han sido de mucho interés a lo largo de los años, como veremos en el capítulo 2 existe una amplia literatura acerca de ese tema. Este tipo de problemas básicamente consiste en que una empresa busca localizar instalaciones con el objetivo de maximizar la demanda capturada, pero sabiendo que las empresas competidoras también van a localizar instalaciones, por lo que puede haber pérdida de clientes.

Este tipo de problemas se han modelado a lo largo de la historia mediante la programación bi-nivel en donde la empresa con mayor jerarquía o la más influyente en el mercado se posiciona en el nivel superior y otra empresa con menos poder se sitúa en el nivel inferior. Es decir, en lugar

de considerar dos procesos relacionados entre sí de manera independiente, ahora se consideran al mismo tiempo, pero respetando una jerarquía en el orden de la toma de decisiones. En problemas de localización es conveniente considerar la opinión de los clientes al abrir nuevas empresas debido a que en muchos casos reales los clientes son libres de satisfacer su demanda en alguna empresa en particular. Usualmente, esta decisión está basada en costos, preferencias, algún contrato predeterminado, o un coeficiente de lealtad, entre otros.

Como mencionamos en la descripción del problema tratado en esta tesis, consideramos un problema de cierre de instalaciones competitivas, por lo que es natural modelar nuestro problema mediante programación bi-nivel, ya que existe una jerarquía entre las empresas que están operando en el mercado, donde en el nivel superior la empresa con mayor jerarquía busca maximizar la demanda capturada una vez que cierra cierto número de instalaciones y a su vez en el nivel inferior la empresa con menos jerarquía busca maximizar la demanda capturada. Además, nosotros agregamos restricciones válidas para la asignación de los clientes que estaban asignados a una instalación que se cierra, es decir, en este problema si a un cliente se le cierra la instalación que le brindaba su demanda, este no se resigna a la más cercana sino que busca dentro en un radio de lealtad predeterminado alguna otra instalación de la misma empresa que le satisficía su demanda. Para resolver el problema presentamos un algoritmo basado en ramificación y acotamiento adaptado de uno de la literatura para problemas de programación bi-nivel binarios.

Además, la escasez de trabajos acerca de problemas de cierre de instalaciones competitivas nos motivó realizar esta investigación para generar nuevo conocimiento, ya que en la literatura el único que aborda el cierre de instalaciones competitivas es el trabajo de Ruiz-Hernández *et al.* (2017).

### 1.3 OBJETIVO

El objetivo de esta tesis consiste en estudiar y definir un problema de cierre de instalaciones competitivas considerando la reacción de los clientes al agregarles la característica de lealtad hacia una empresa en particular y proponer una metodología de solución para resolver el problema a optimalidad.

Primeramente, proponemos una formulación de programación tri-nivel sin considerar lealtad de los clientes, por lo cual agregamos un conjunto de índices auxiliar que nos ayuda a reducir el problema de programación tri-nivel en uno bi-nivel y después agregamos restricciones válidas que nos ayudan a satisfacer las restricciones de lealtad de los clientes. Además, adaptamos un algoritmo basado en ramificación y acotamiento para resolver nuestro problema de cierre de instalaciones considerando la lealtad de los clientes.

Por último, como solo se encontró un trabajo relacionado con el cierre de instalaciones, podemos decir que nuestro trabajo es el primero en considerar el cierre de instalaciones competitivas visto desde el enfoque de programación bi-nivel.

### 1.4 METODOLOGÍA

Para que esta tesis pudiera realizarse de manera satisfactoria se siguieron los siguientes pasos como parte de la metodología implementada.

1. Revisión de la literatura de programación bi-nivel.
2. Realizar una investigación de los antecedentes del problema  $(r|p)$ -Centroide.
3. Revisión de la literatura sobre problemas de cierre de instalaciones.

4. Revisión de la literatura sobre problemas de cierre de instalaciones competitivas.
5. Análisis las propiedades del problema así como el planteamiento del modelo.
6. Analizar el modelo, así como agregar un conjunto de índices auxiliar para reducir un modelo tri-nivel en uno bi-nivel.
7. Agregar como novedad la lealtad de los clientes hacia su empresa actual, por lo que agregamos restricciones que cumplieran dicha característica en el modelo matemático bi-nivel.
8. Desarrollamos un método de solución basado en enumeración exhaustiva para resolver nuestro problema utilizando instancias generadas aleatoriamente.
9. Revisión de la literatura sobre algoritmos basados en ramificación y acotamiento para problemas de localización de instalaciones.
10. Examinar la metodología del algoritmo de ramificación y acotamiento para problemas de programación bi-nivel binarios.
11. Desarrollo en un algoritmo basado en ramificación y acotamiento para resolver nuestro problema.
12. Durante una estancia de investigación en el Colegio Universitario de Estudios Financieros en España se realizaron dos generadores de instancias para nuestro problema, así como crear otras instancias al utilizar una red de nodos bien conocida en la literatura y una versión extendida de esta, dándolas como datos de entrada al segundo generador.
13. Se comenzó a escribir un artículo para nuestro problema con el fin de publicarlo en una revista indexada.

14. Desarrollo e implementación computacional del algoritmo propuesto, así como el análisis de los resultados obtenidos.
15. Presentación de los avances de tesis en el XXVI Escuela Nacional de Optimización y Análisis Numérico (ENOAN 2017), VI Congreso de la Sociedad Mexicana de Investigación de Operaciones (CSMIO 2017), 2nd International Workshop on Bilevel Programming (IWOBIP'18) y en el EURO/ALIO International Conference 2018 on Applied Combinatorial Optimization.
16. Comparación del desempeño del algoritmo propuesto con uno de enumeración completa.

## 1.5 ESTRUCTURA DE LA TESIS

En este primer capítulo se presentó la descripción del problema  $(r|p)$ -centroide negativo considerando lealtad de los clientes y se dio a conocer el objetivo central de la tesis. También se describió la justificación de la investigación y la motivación que se tuvo para realizarla. Por último, se detalló la metodología propuesta para la obtención del producto final, es decir, de la tesis completa.

En el capítulo 2, se presenta la revisión de literatura acerca de los problemas de localización de instalaciones competitivas modelados mediante programación bi-nivel. Además, se hace la revisión de artículos relacionados con los problemas de cierre de instalaciones.

En el capítulo 3 se define el problema, se inicia con una introducción a la programación multi-nivel, se describe el problema abordado en esta tesis, se presenta un modelo matemático tri-nivel y se justifica por qué podemos reducirlo a un problema bi-nivel al agregar un conjunto de índices auxiliar, después agregamos restricciones válidas que ayudan a satisfacer las restricciones de lealtad de los clientes. Por último, se presenta un ejemplo ilustrativo del funcionamiento de dichas restricciones.

En el capítulo 4 se describe a detalle el algoritmo de ramificación y acotamiento adaptado que implementamos para resolver el problema propuesto en esta tesis. Además, se da una ejemplificación para mostrar el buen funcionamiento de este.

En el capítulo 5 se presentan los resultados obtenidos de la experimentación computacional. Se describen a detalle los dos generadores de instancias que se realizaron, así como una red de nodos de demanda muy conocida encontrada en la literatura para tener instancias de prueba. Se describen los escenarios de los experimentos computacionales, así como el entorno computacional. Por último, se presentan los resultados obtenidos del algoritmo propuesto y hacemos una comparativa con otro algoritmo de enumeración exhaustiva y se presentan los gráficos que ilustran una comparativa de los tiempos de ejecución correspondientes.

En el capítulo 6, se presentan las conclusiones realizadas producto de la investigación hecha en esta tesis y se presentan algunas posibles extensiones de trabajo futuro.

---

## CAPÍTULO 2

# REVISIÓN DE LITERATURA

---

En este capítulo se revisa la literatura relacionada con la rama de localización de instalaciones competitivas (CFL, por sus siglas en inglés). Es de particular interés reseñar los trabajos que utilizan un enfoque de programación bi-nivel. Además, se presentan las diferentes formas en las que se ha hecho frente a la problemática del cierre de instalaciones. Este enfoque ha sido poco estudiado en la literatura, apenas unos cuantos investigadores han empezado a estudiar este tipo de problemas debido a las situaciones reales que enfrentan las empresas hoy en día.

## 2.1 LOCALIZACIÓN DE INSTALACIONES COMPETITIVAS.

La ciencia que estudia problemas de localización de instalaciones (FLP, por sus siglas en inglés) ha sido ampliamente abordada y sigue creciendo día con día en el área de investigación de operaciones. En general, el objetivo de este campo de investigación es formular modelos matemáticos para determinar el mejor subconjunto de instalaciones para abrir en una región. Estos problemas pueden modelarse mediante una red, un espacio continuo o mediante un conjunto de puntos discretos. Existen modelos para localizar instalaciones públicas como escuelas u oficinas de correo, modelos para localizar instalaciones de venta al por menor en presencia de competencia, modelos para localizar servicios de emergencia como estaciones de bomberos o ambulancias, y modelos para localizar plantas y almacenes, entre otros. La mayoría de ellos busca maximizar su número de clientes, o bien,

brindar el servicio a la mayor cantidad de usuarios con el fin de capturar la mayor demanda posible. Para una visión más general del estado del arte se recomienda ver Drezner & Hamacher (2001).

A continuación, mencionamos diversos trabajos dedicados al área de localización, pero que en particular, consideran la presencia de empresas competidoras. A esta rama se le conoce como problemas de localización de instalaciones competitivas (CFL, por sus siglas en inglés) introducido por primera vez en Hotelling (1929), que básicamente consideraban el problema de localizar instalaciones sabiendo que las empresas competidoras también van a localizar instalaciones y por ende puede haber pérdida de clientes. Por lo regular, el objetivo de cada empresa es maximizar la demanda capturada. Nosotros nos centraremos en el problema  $(r|p)$ –centroide, que de aquí en adelante lo llamaremos RPCP por sus siglas en inglés. El RPCP fue introducido por primera vez en Hakimi (1983), en donde usaron un enfoque de programación bi-nivel para modelar el problema. Antes de describir el problema en cuestión, considere a  $I$  como el conjunto de sitios potenciales, a  $p$  y  $r$  como el número de instalaciones a abrir por los tomadores de decisiones, respectivamente. Entonces, el RPCP consiste en considerar a dos empresas competidoras que tienen cierta jerarquía entre ellas y desean entrar en un mercado determinado. Entonces, la empresa con mayor jerarquía selecciona las  $p$  instalaciones a abrir de los  $|I|$  sitios potenciales, y después, la empresa con menor jerarquía abre las  $r$  instalaciones de los  $|I| - p$  sitios potenciales restantes. Ambas empresas toman sus decisiones tomando en cuenta la maximización de la demanda capturada.

Los trabajos que abordan el RPCP en un espacio continuo son: Bhadury *et al.* (2003) donde proponen dos heurísticas para los problemas de líder-seguidor. La primera heurística adopta un enfoque voraz y la otra uno de diferenciación mínima. Los resultados de las pruebas computacionales indican una rápida convergencia de ambas heurísticas, pero la heurística de diferenciación mínima lleva la ventaja. Por lo que los autores llegan a

la conclusión de que ésta última parece ser la más robusta de las dos. En Carrizosa *et al.* (2012) presentan una heurística alternativa mejorada. En esta nueva heurística alternativa resuelven el problema del seguidor a optimalidad y se utiliza un procedimiento de agrupación con un método exacto de tiempo polinomial para el problema (1|1)-centroide. Los resultados computacionales para instancias generadas al azar muestran las ventajas del enfoque propuesto en comparación con el de Bhadury *et al.* (2003). En Rodríguez *et al.* (2009) estudiaron métodos de optimización basados en un enjambre de partículas para resolver el problema. Evidentemente en estos trabajos, las instalaciones se pueden localizar en cualquier lugar del plano euclidiano.

En la literatura también se encuentran trabajos dedicados al RPCP discreto, por ejemplo, en Alekseeva *et al.* (2010) se presentan varios enfoques para abordar el problema, introducen algunas heurísticas e incluyen un método exacto, la primera heurística es un algoritmo memético híbrido (HMA, por sus siglas en inglés) que utiliza una búsqueda tabú probabilística como procedimiento de mejora local, emplea operadores genéticos simples y en la búsqueda tabú utiliza una estructura de vecindario de intercambio probabilístico, la cual es bien conocida y utilizada en el problema de la  $p$ -mediana, vea el artículo de Mladenović *et al.* (2007) para una visión general de este problema. Con esta estructura, una vecindad solamente contiene ciertos elementos con una probabilidad dada para acelerar la búsqueda. Es decir, utilizan un vecindario reducido. También utilizan una reformulación para reducir el problema bi-nivel en un problema mixto entero (MIP) de un solo nivel. Con esta reformulación, fueron capaces de resolver el RPCP de manera eficiente, este enfoque supera a varias heurísticas más simples, incluida la heurística alternativa propuesta en Bhadury *et al.* (2003) para la versión continua del problema. En Alekseeva & Kochetov (2013) se presentan resultados obtenidos mediante una búsqueda tabú y los resultados son similares a los obtenidos mediante el HMA propuesto

en Alekseeva *et al.* (2010). Además, presentan un método iterativo exacto basado en un problema de programación binaria de un solo nivel, el cual agrega restricciones y variables de manera exponencial conforme el tamaño del problema crece. Debido a la gran cantidad de variables, este problema de un solo nivel es resuelto mediante un enfoque de generación de columnas. El valor óptimo de este modelo define una cota superior y al resolver el problema del seguidor utilizando soluciones del modelo de un solo nivel, se obtiene una cota inferior. Si las cotas coinciden se ha encontrado el valor óptimo. El algoritmo memético híbrido se aplica para encontrar la familia inicial de soluciones del seguidor y pueden resolver instancias de manera óptima con hasta 100 clientes y  $p = r = 5$ . En Alekseeva *et al.* (2015), proporcionan un método iterativo exacto. Utilizaron un algoritmo de búsqueda local en cada iteración para encontrar una solución factible para un sistema de restricciones y muestran buenos resultados. Este enfoque requiere mucho esfuerzo computacional, pero obtienen el óptimo global en un número relativamente pequeño de pasos. Además compara sus resultados de tiempo contra los obtenidos por Roboredo & Pessoa (2013), siendo más rápido en la mayoría de los casos.

En Davydov *et al.* (2012) proponen otro algoritmo de búsqueda tabú y utilizan una estructura de vecindad probabilística basada en intercambios de instalaciones (entiéndase esto como localizar una instalación del líder en otra localización en la estructura de vecindario) similar a la desarrollada en Alekseeva *et al.* (2010). Para resolver el problema bi-nivel, el problema del seguidor se resuelve mediante relajación Lagrangiana, por lo que no se asegura el valor óptimo. Sin embargo, para muchos de las instancias se obtienen soluciones óptimas. El método se prueba en las instancias utilizadas en Alekseeva *et al.* (2010) y adicionalmente, en algunos casos no euclidianos. Luego, en Davydov *et al.* (2014) se proponen dos metaheurísticas que se basan en la modificación del vecindario de búsqueda. En la primera, utilizan una búsqueda de vecindarios variable (VNS, por sus siglas en

inglés) donde consideraban sub-vecindarios denominados *Fswap*, *Nswap* y *Cswap*. Un elemento del vecindario *Fswap* se determina cerrando una instalación del líder y abriendo alguna otra en un lugar elegido por el seguidor. *Nswap* consiste en soluciones candidatas que se generan al cerrar una instalación líder y abrir una instalación en una localización cercana. *Cswap* consta de todas las soluciones que se encuentran en la estructura de intercambio de vecindario pero que aún no están en *Fswap* o *Nswap*. En la segunda metaheurística, utilizan un método llamado búsqueda tabú estocástica (STS, por sus siglas en inglés), en el cual se utiliza el vecindario de intercambio probabilístico que ha sido propuesto por Alekseeva *et al.* (2010). El STS usa la misma división de vecindario que el VNS y se mantiene una lista tabú para eliminar elementos del vecindario que consisten en parejas de instalaciones del líder que se han cerrado y abierto durante las últimas iteraciones. Ambos métodos consideran el modelo binivel introducido en Alekseeva *et al.* (2010) para estudiar el problema. Se encuentran soluciones más rápido que la búsqueda tabú propuesta en Davydov *et al.* (2012) para muchos casos. Además, probaron sus algoritmos en instancias no euclidianas, en las que ambos métodos, VNS y STS, mostraron un rendimiento similar. En Campos-Rodríguez *et al.* (2012) optimizan el problema discreto mediante un algoritmo de enjambre de partículas en salto (JPSO, por sus siglas en inglés) que ellos proponen, esto debido a que el método de enjambre de partículas funciona para problemas continuos. Este método usa dos enjambres, uno para el líder y otro para el seguidor. Recibe este nombre ya que las partículas saltan de una solución a otra dependiendo del vecindario y la mejor posición obtenida para todas las partícula hasta el momento, es decir, la mejor posición global. En los experimentos, este algoritmo fue capaz de resolver instancias con 25 clientes,  $p = 3$  y  $r = 2$  a optimalidad.

En Roboredo & Pessoa (2013) desarrollaron un algoritmo exacto basado en ramificación y corte. Utilizan un modelo de programación entera

de un solo nivel que es similar al modelo de Alekseeva *et al.* (2010), pero con un número polinomial de variables. En el se tienen muchas restricciones, una para cada estrategia del seguidor, es decir, para cada conjunto de posibles localizaciones de instalaciones del seguidor. Una razón importante para el éxito de su método es la introducción del fortalecimiento de las desigualdades mediante la anulación de muchas de las restricciones que aumentaban de manera exponencial. Debido a la suposición de que los clientes son conservadores, la cota inferior de la solución del líder se convierte en cero si el seguidor elige la misma localización de la instalación que ha localizado previamente el líder. Por lo tanto, para cada una de ellas se da una localización alternativa que se elige si la posición ya ha sido utilizada por el líder. Estos cortes están separados por una heurística voraz o por la resolución de un modelo de programación mixto entero. Para la mayoría de las instancias, presentan mejores resultados que los de Alekseeva *et al.* (2010), es decir, encontraron soluciones óptimas en menos tiempo. Las instancias con 100 clientes y con valores de hasta  $p = r = 15$  instalaciones a abrir podrían resolverse de manera óptima. Los autores también presentan resultados prometedores para  $p = r = 20$ , pero no pueden demostrar su optimalidad dentro del límite de un tiempo determinado en diez horas.

Se recomienda leer el trabajo de Alekseeva & Kochetov (2013) donde ofrecen una visión más general de las investigaciones recientes sobre el RPCP discreto. Además, mejoran su método iterativo exacto mediante un modelo que tiene un número polinomial de variables, en lugar de un número exponencial. También utilizan las desigualdades de fortalecimiento presentadas en Roboredo & Pessoa (2013). Este enfoque iterativo mejorado es capaz de encontrar soluciones óptimas para instancias con hasta 100 clientes y  $p = r = 15$ . Especialmente para las instancias con  $p = r \in \{5, 10\}$ , las soluciones óptimas se encuentran significativamente más rápido que con el algoritmo de ramificación y corte propuesto por Roboredo & Pessoa (2013). En Biesinger *et al.* (2015) proponen un algo-

ritmo genético híbrido con varias mejoras. Ellos archivaron las soluciones exploradas en estructuras de datos para reducir el número de evaluaciones de soluciones innecesarias y evitar la convergencia prematura, lo que conduce a una ganancia de eficiencia significativa. Otra parte importante del algoritmo es el procedimiento de mejora local integrado. Investigaron varias formas de combinar la búsqueda local con el registro de soluciones e identificaron el vecindario reducido para un mejor funcionamiento en la práctica. Consideraron diferentes métodos de evaluación de soluciones y encontraron una forma efectiva de combinarlos, lo que llevó al esquema de evaluación multinivel. Por último, mejoraron los resultados del algoritmo utilizando una búsqueda tabú en la mejora local. En Alekseeva *et al.* (2015), proporcionan un método iterativo exacto. Utilizaron un algoritmo de búsqueda local en cada iteración para encontrar una solución factible para un sistema de restricciones y muestran buenos resultados. Este enfoque requiere mucho tiempo, pero obtienen el óptimo global en un número relativamente pequeño de pasos. Además compara sus resultados de tiempo contra los obtenidos por Roboredo & Pessoa (2013), siendo más rápido en la mayoría de los casos.

En la literatura solo encontramos dos trabajos dedicados al RPCP en redes: Benati & Laporte (1994) desarrollaron una heurística de búsqueda tabú para abordar el problema. También utilizan una búsqueda tabú en el nivel inferior para encontrar cotas primales del problema  $(r|X_p)$ –medianoide, por lo que la calidad de la solución obtenida es aproximada, ya que el problema  $(r|X_p)$ –medianoide no se resuelve a optimalidad y Kress & Pesch (2012a) consideran la demanda de los clientes tanto en los nodos como en las aristas de la red. Además, ofrecen una visión general de los problemas de localización secuencial competitiva en redes.

Es importante hacer notar que para los trabajos descritos anteriormente los clientes adoptan un comportamiento conservador, es decir, se asignan a la instalación de la empresa con mayor jerarquía si es que hay

dos instalaciones a la misma distancia. Además, la asignación se basa únicamente en la distancia y la demanda será cubierta por solo una instalación, es decir, el cliente se va a asignar a la instalación más cercana y esta le va a satisfacer toda la demanda.

Por último, mencionaremos algunos trabajos con algunas variantes del RPCP, por ejemplo en Kochetov *et al.* (2013) describen un algoritmo para el RPCP con costos fijos para abrir una instalación. En esta versión, los clientes dividen su demanda de forma proporcional a las instalaciones, con respecto a unos factores de atracción. El principio del algoritmo es similar a la heurística alternativa para el RPCP propuesta en Bhadury *et al.* (2003). Otro trabajo que asume la división proporcional de las demandas es realizado por Biesinger *et al.* (2014). Allí, los autores emplean un algoritmo genético en combinación con un registro de las soluciones similar al propuesto por Biesinger *et al.* (2015) y en Serra & ReVelle (1994), proponen un enfoque heurístico para una variante del RPCP discreto que se basa en la resolución repetida del problema de captura máxima (MAXCAP). El problema MAXCAP es similar al problema  $(r|X_p)$ -medianoide con la diferencia de que es posible colocar una instalación en una de las localizaciones del líder. Este supuesto, nos lleva a que la demanda capturada se comparte de manera equitativa entre los dos jugadores. El algoritmo es básicamente una búsqueda local que utiliza una estructura de vecindad basada en intercambio y las soluciones candidatas se evalúan resolviendo el problema de MAXCAP mediante programación entera o utilizando una heurística de búsqueda local para obtener buenas soluciones factibles en instancias más grandes.

Una revisión de literatura reciente sobre varios tipos de modelos de localización de instalaciones competitivas se puede encontrar en Kress & Pesch (2012b).

## 2.2 PROBLEMAS DE CIERRE DE INSTALACIONES.

Como se mencionó anteriormente, los problemas de localización de instalaciones han sido ampliamente estudiados y existe una amplia variedad de algoritmos para resolver el problema, sin embargo no muchos investigadores se han interesado por los problemas en los cuales se requiera el cierre de instalaciones en lugar de su apertura. Esta problemática de cerrar instalaciones ya existe desde hace algún tiempo. En ReVelle *et al.* (2007) se menciona que en el sector privado, en particular, en el mercado de bienes y servicios industriales, el cual es altamente competitivo, las empresas eligen sus estrategias a la vista de las empresas competidoras, pero dichas estrategias no siempre son efectivas por lo que el cierre de instalaciones puede ocurrir como parte integral de la estrategia. El sector minorista también se ve afectado por la desaceleración comercial que puede causar el cierre de muchas tiendas minoristas debido a las crisis económicas que ocurren dentro de la empresa. En el sector público es común ver hospitales o centros de salud cerrados. También, en el área de telefonía se realizan la apertura y cierre de instalaciones de manera frecuente para reordenar oficinas nuevas o existentes con el fin de ahorrar en costos de telecomunicaciones y tener una mayor utilización de personal de respuesta telefónica. En el transporte urbano, puede ser necesario eliminar algunas paradas y reubicar otras debido a los cambios en la densidad de población y los cambios en la red. Para una visión más general de estos problemas, se recomienda ver Weston Jr (1982) y Murray & Wu (2003). Por último, se recomienda la lectura de la problemática para la consolidación del sistema escolar como resultado de la disminución de la matrícula, la cual puede leerse en Bruno & Andersen (1982), Diamond & Wright (1987) y Church & Murray (1993). Cabe mencionar que estos tres trabajos consideran el cierre y apertura de escuelas de manera simultánea.

A continuación mencionaremos los trabajos más recientes en los cuales los autores consideran el cierre de instalaciones. Por ejemplo, en Wang *et al.* (2003) modelan una situación en la cual, debido al cambio en la distribución de la demanda, la distribución de la población en ciertas zonas geográficas, o simplemente la baja demanda en las instalaciones que tiene operando una empresa ya no brindan un buen servicio. Por esto, se ven en la necesidad de cerrar instalaciones de las que tienen operando con normalidad y abrir algunas otras de manera simultánea con el fin de brindar un mejor servicio a sus clientes. Para lograr esto, toman en cuenta el presupuesto de la empresa, así como el número deseado de instalaciones que se quieren tener operando en el mercado sin exceder el costo anual operativo. Demuestran que su problema pertenece a la clase de complejidad NP-duro, por lo que obtienen soluciones factibles al problema utilizando tres algoritmos heurísticos y para conocer la solución óptima utilizan el optimizador comercial CPLEX obteniendo buenas soluciones incluso para instancias grandes. Basados en los resultados obtenidos, se demuestra la eficiencia de los tres algoritmos propuestos. También discuten la situación donde los costos operativos son parte de la función objetivo.

En ReVelle *et al.* (2007) se presentan dos modelos que abordan el cierre de las instalaciones debido a emergencias financieras. Uno de ellos investiga el cierre de las instalaciones en un entorno donde se considera la presencia de competidores, en el cual se busca ceder la cantidad mínima de demanda, y el otro examina el cierre en una situación de exigencia financiera pero sin competencia.

En Bhaumik (2010), se define la palabra deslocalización (*delocation*, en inglés) como el hecho de cerrar una instalación existente y no en el sentido de reubicar la instalación como se conoce en el sector económico. Presenta un problema en el cual una empresa desea reducir el tamaño de su red de distribución debido a la baja demanda que enfrentan las empresas por diversos factores, como los cambios competitivos, tecnológicos, ambien-

tales o de estilo de vida. Con esta decisión se espera que el cierre de algunas instalaciones reduzca los costos y mejore el nivel de servicio. Proporciona dos modelos alternativos, el primero es una formulación de carga fija que permite la reasignación de nodos de demanda entre las instalaciones restantes, y la segunda formulación (denominada modelo de deslocalización puro) en donde se agrega la restricción adicional de reasignación para que los clientes a los que se les cerró la instalación que les brindaba el servicio sean reasignados.

En Ruiz-Hernández & Delgado-Gómez (2016), presentan un problema de reducción de instalaciones bancarias capacitadas redundantes en una red después de la fusión y adquisición financiera, tomando en cuenta el tamaño de las sucursales cerradas, la existencia de competidores y la incertidumbre en la respuesta de la demanda, en el cual considera dos tipos de clientes, los que siguen con la misma institución bancaria después de la reestructuración de la red y los que son cedidos a una institución bancaria diferente. Formulan el problema como un modelo de programación estocástico binario de dos etapas y lo resuelven para tres escenarios diferentes. En el primer escenario el banco realiza la reestructuración completa durante la primera etapa y luego, durante la segunda etapa, observa la reacción de los clientes. En el segundo escenario durante la primera etapa, el banco solo decide cuáles sucursales cerrar y cuáles mantener abiertas. En la segunda etapa, se observa la reacción de los dos tipos de clientes y se realizan las modificaciones necesarias en las capacidades virtuales y/o físicas de la red. Por último, en el tercer escenario durante la primera etapa, el banco cierra las sucursales estratégicamente y re-ubica a los clientes que se les cerró su instalación y los cambios en las capacidades físicas de las sucursales abiertas se realizan en la segunda etapa.

En Ruiz-Hernández *et al.* (2015), presentan un nuevo modelo de optimización binario de reestructuración de sucursales capacitadas que amplía la literatura disponible en modelos de cierre de instalaciones. En él se con-

sideran tanto los costos de cierre como los de operaciones a largo plazo, y aborda el problema de cambiar el tamaño de las sucursales abiertas para mantener un nivel de servicio constante. También toman en cuenta la presencia de competidores y la oportunidad de ceder la participación de mercado siempre que los costos de reestructuración sean prohibitivamente costosos. Obtienen buenos resultados y realizan un análisis de sensibilidad para parámetros críticos del modelo y llegan a la conclusión de que el diseño final de la red depende de ciertas decisiones estratégicas relacionadas con la redundancia de las sucursales, así como su cercanía a los nodos de demanda y a las sucursales de la competencia.

Por último, en Ruiz-Hernández *et al.* (2017) es el único trabajo encontrado en la literatura que considera el problema de cierre de instalaciones competitivas. Aquí presentan un problema donde dos empresas pretenden cerrar instalaciones que ya están operando en el mercado, esto debido a la contracción económica que viven algunos países teniendo en cuenta tanto las posibles acciones de sus empresas rivales como la reacción de los clientes afectados. El objetivo es minimizar la pérdida de participación de mercado. Presentan un modelo de deslocalización competitiva desde un enfoque de teoría de juegos y el resultado del juego se analiza para tres escenarios diferentes: comportamiento voraz, conjeturas de Cournot y estrategias Stackelberg. Consideran la existencia de ciertos niveles de lealtad por parte de los clientes hacia su proveedor original mediante una función de decaimiento lineal. Los resultados numéricos consiguen obtener equilibrios de Nash para pequeñas reducciones en el tamaño de la red (menos del 20% de instalaciones) bajo un comportamiento voraz. Para conseguir reducciones más grandes, se indica la necesidad de considerar las acciones de las otras empresas. Además, menciona la existencia de una ventaja en el juego Stackelberg si las empresas tienen un poder de mercado diferente. Con respecto al efecto de los diferentes niveles de lealtad del cliente, los experimentos numéricos confirmaron que como era de esperarse, cuanto

---

menos leales son los clientes, mayor es la cuota de mercado capturada por la empresa más fuerte.

---

CAPÍTULO 3

EL PROBLEMA  $\overline{(r|p)}^{\mathcal{L}}$  – CENTROIDE

---

En este capítulo se describe el problema bi-nivel  $\overline{(r/p)}^{\mathcal{L}}$ –Centroide, primeramente se presenta la formulación matemática de un modelo de optimización tri-nivel y uno bi-nivel pero sin tomar en cuenta la lealtad de los clientes. Después se introduce un conjunto de restricciones válidas que garantizan que la asignación de los clientes a las instalaciones se realice considerando la lealtad. Luego del análisis pertinente, dichas restricciones nos ayudan a satisfacer las restrcciones de lealtad de los clientes. Para facilitar la comprensión de este capítulo, primero se hace una breve introducción a la programación multi-nivel, y se presentan algunos conceptos importantes relacionados con el tema principal de esta tesis, es decir, con la programación bi-nivel.

### 3.1 PROGRAMACIÓN MULTI-NIVEL

Hoy en día muchos componentes del proceso de toma de decisiones tienen que ser considerarse de manera jerarquizada. Incluso muchas de estas decisiones no están en nuestras manos, es decir, existe una jerarquía de mando en la toma de decisiones. Esto nos lleva a identificar que las decisiones se toman en diferentes niveles dentro de esta jerarquía. Una manera de abordar tales jerarquías es concentrarse en un nivel, pero incluyendo el comportamiento de los otros niveles como suposiciones. En Migdalas *et al.* (2013) se define la programación multi-jerárquica o también llamada optimización multi-nivel como el área que estudia y analiza los problemas

de toma de decisiones con estructura jerárquica. La programación multi-nivel se definió formalmente y se estudió por primera vez en Bracken & McGill (1973, 1974), en donde se presentó como una generalización de la programación matemática.

El problema general de optimización multi-nivel (**P**) se puede formular de la siguiente manera:

$$(\mathbf{P}_1) \quad \min_{x^1} f_1(x^1, x^2, \dots, x^k) \quad (3.1)$$

$$\text{Sujeto a:} \quad g_1(x^1, x^2, \dots, x^k) \leq 0 \quad (3.2)$$

donde  $x^2$  resuelve:

$$(\mathbf{P}_2) \quad \min_{x^2} f_2(x^1, x^2, \dots, x^k) \quad (3.3)$$

$$\text{Sujeto a:} \quad g_2(x^1, x^2, \dots, x^k) \leq 0 \quad (3.4)$$

(**P**)

.

.

.

donde  $x^k$  resuelve:

$$(\mathbf{P}_k) \quad \min_{x^k} f_k(x^1, x^2, \dots, x^k) \quad (3.5)$$

$$\text{Sujeto a:} \quad g_k(x^1, x^2, \dots, x^k) \leq 0 \quad (3.6)$$

(**P**<sub>1</sub>) es llamado el problema del primer nivel y corresponde al nivel más alto en la jerarquía. El responsable de la toma de decisiones en este nivel controla la variable de decisión  $x_1$  y su objetivo es minimizar la función  $f_1$ . Hay que tener en cuenta que las decisiones tomadas en los niveles de menor jerarquía tienen un impacto en la función objetivo y/o restricciones

del primer nivel. Análogamente,  $(\mathbf{P}_k)$  es llamado el problema del  $k$ -ésimo nivel y corresponde al nivel más bajo en la jerarquía.

Es importante mencionar que el problema multi-nivel  $(\mathbf{P})$  es muy difícil de resolver, lo menciona Migdalas *et al.* (2013). Por ejemplo, considere el caso más simple de un problema de optimización multi-nivel, es decir, considere un problema lineal con únicamente dos niveles. Es difícil de resolver ya que es NP-duro, se demuestra en Bard (1998). En la siguiente subsección presentaremos este caso con más detalle y se definen algunos conceptos importantes que nos ayudarán para plantear el método de solución propuesto.

### 3.1.1 PROGRAMACIÓN BI-NIVEL

Como se mencionó anteriormente, un caso particular de la programación multi-nivel es cuando se tienen solamente dos tomadores de decisiones, en este caso se llama programación de dos niveles, o simplemente, programación bi-nivel. Es Stackelberg *et al.* (1952) el primero en introducir la estrategia básica de líder y seguidor para un duopolio, asociando el problema del primer y segundo nivel al líder y al seguidor, respectivamente.

En Bard (1998) se describe el problema general de programación bi-nivel como sigue:

$$\min_x F(x, y) \quad (3.7)$$

$$\text{Sujeto a: } G(x, y) \leq 0 \quad (3.8)$$

donde  $y$  resuelve:

$$\min_y f(x, y) \quad (3.9)$$

$$\text{Sujeto a: } g(x, y) \leq 0 \quad (3.10)$$

$$x, y \geq 0 \quad (3.11)$$

Las restricciones (3.7) – (3.9) definen el problema del líder, donde (3.7) es la función objetivo del líder quien busca minimizar la función  $F(x, y)$ , sujeto a las restricciones (3.8) – (3.9). Note que (3.9) denota otro problema de programación matemática llamado problema del seguidor que está definido por las restricciones (3.9) – (3.11), donde (3.9) busca minimizar la función  $f(x, y)$  considerando las restricciones (3.10) y (3.11). Esta última restricción indica la no negatividad de las variables de decisión del problema de programación bi-nivel.

Ahora definiremos los conceptos básicos más importantes de la programación bi-nivel. Dichos conceptos son necesarios para comprender el algoritmo propuesto para resolver el problema estudiado en esta tesis:

1. Conjunto de restricciones del problema de programación bi-nivel.

$$\Omega = \{(x, y) \mid G(x, y) \leq 0, g(x, y) \leq 0\} \quad (3.12)$$

2. Conjunto factible del seguidor para cada  $x \in X$ .

$$\Omega(x) = \{y \in Y \mid g(x, y) \leq 0\} \quad (3.13)$$

3. Conjunto de reacción racional del seguidor para cada  $x \in \Omega(x)$ .

$$\Psi(x) = \{y \in Y \mid y \text{ resuelve : } \min\{f(x, y) \leq 0 \mid y \in \Omega(x)\}\} \quad (3.14)$$

4. Región inducible.

$$IR = \{(x, y) \mid (x, y) \in \Omega, y \in \Psi(x)\} \quad (3.15)$$

Como mencionamos anteriormente, el caso más simple de un problema bi-nivel es aquél donde el problema del líder es un problema lineal y el problema del seguidor también. Este caso se conoce como el problema de programación bi-nivel lineal (BLPP, por sus siglas en inglés). En Bard (1998) está demostrado que este problema pertenece a la clase de complejidad NP-duro.

## 3.2 DESCRIPCIÓN DEL PROBLEMA

Considere una situación en un mercado específico, en donde hay dos empresas que ofrecen un servicio requerido por un conjunto de clientes. Dichas empresas compiten entre sí y deben cerrar algunas de sus instalaciones existentes. Debido a la naturaleza de ambas empresas y al posicionamiento que tienen actualmente en el mercado, existe una jerarquía entre ellas. Al momento de tomar la decisión sobre cuáles instalaciones van a cerrar, deben tener en cuenta las posibles acciones de la empresa rival y la reacción de los clientes. Evidentemente, las instalaciones que cerrará cada una de las empresas deberán ser únicamente de las que le pertenecen. El objetivo que tiene cada empresa es conseguir la máxima captura de demanda de los clientes en el mercado. Entonces, este problema consiste en decidir las  $p$  y  $r$  instalaciones que van a cerrar ambas empresas, respectivamente. A este problema nosotros le llamaremos el problema  $(r|p)$  – *centroide negativo* el cual denotaremos como  $\overline{(r|p)}$  – *centroide*.

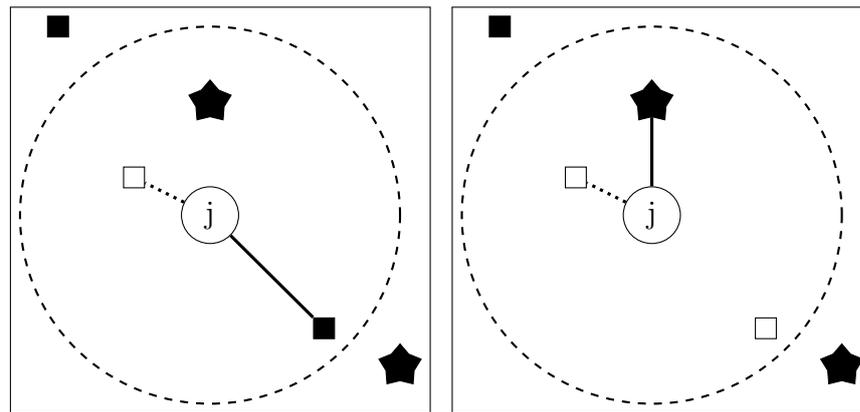
Una forma natural de modelar este problema es como un problema de programación tri-nivel, donde en el primer nivel se cierran  $p$  instalaciones de las que ya tiene operando en el mercado la empresa con mayor jerarquía, después en el segundo nivel se cierran  $r$  instalaciones de las que tiene operando en el mercado la otra empresa, y por último, en el tercer nivel los clientes seleccionan la instalación que más les convenga. Más adelante, en este capítulo se presenta el modelo de programación tri-nivel. Debido a

la complejidad del modelo tri-nivel, se busca la manera de tener un modelo más tratable. Para lograr esto, agregamos restricciones válidas sobre la asignación de los clientes a una de las instalaciones abiertas más cercanas, ya sea de su misma empresa o de la competencia, utilizando un conjunto de índices que definiremos más adelante. Estas restricciones permiten modelar el problema con un modelo de programación bi-nivel. Con estas restricciones se logra reducir el problema tri-nivel en un modelo equivalente de programación bi-nivel.

Como una de las novedades en esta tesis y con el fin de hacer más interesante nuestro problema, agregamos un supuesto importante para hacerlo más real. El supuesto recae en considerar que los clientes mantienen cierta lealtad hacia la empresa que le brinda servicio actualmente. Es decir, la reacción de los clientes que satisfacían su demanda en una de las instalaciones que cerró la empresa, será basada en una cierta lealtad. Para esto, consideramos un radio predeterminado para cada cliente con el fin de que éste no se asigne a la instalación más cercana, sino que permanezca con la misma empresa siempre y cuando exista otra instalación de la misma empresa dentro de un radio que está dispuesto a tolerar (ver figura 3.1). Es importante mencionar que los clientes serán leales a la empresa a la que estén asignados en un principio, es decir, con la empresa que le brinda el servicio requerido. A este problema lo llamaremos  $(r|p)$ –*centroide negativo* considerando lealtad de los clientes, el cual denotaremos simplemente como  $\overline{(r/p)}^{\mathcal{L}}$ –centroide.

### 3.3 MODELO MATEMÁTICO DE PROGRAMACIÓN TRI-NIVEL PARA EL PROBLEMA $\overline{(r|p)}$ –CENTROIDE

Como se mencionó anteriormente una manera natural de modelar nuestro problema es que en el primer nivel se cierren  $p$  instalaciones, luego



(a) Ejemplo de un cliente leal, permanece con la misma empresa. (b) Cliente capturado por la competencia.

■- instalación de la empresa 1 abierta ★- instalación de la empresa 2 abierta

Figura 3.1: Lealtad de un cliente  $j$  hacia la empresa a la que es leal sin importar la distancia pero dentro de un radio predefinido (círculo punteado). En blanco se denota una instalación cerrada.

en el segundo nivel se cierren  $r$  instalaciones y en un tercer nivel los clientes se asignen a la instalación más cercana.

Antes de presentar la formulación matemática, primero se van a definir propiamente los conjuntos, parámetros y variables de decisión involucrados en nuestro problema.

### Conjuntos.

$I^L$ : Conjunto de las instalaciones del primer nivel.

$I^F$ : Conjunto de las instalaciones del segundo nivel.

$I = I^L \cup I^F$ : Conjunto de las instalaciones.

$J$ : Conjunto de los clientes.

### Parámetros.

$p$ : Número de instalaciones a cerrar en el primer nivel.

$r$ : Número de instalaciones a cerrar en el segundo nivel.

$b_j$ : Demanda del cliente  $j \in J$ .

$d_{ij}$ : Distancia entre el cliente  $j \in J$  y la instalación  $i \in I$

### Variables de decisión.

Las variables de decisión binarias son las siguientes:

Del primer nivel:

$$x_i^L = \begin{cases} 0, & \text{Si se cierra la instalación } i \in I^L \\ 1, & \text{En otro caso} \end{cases}$$

Del segundo nivel:

$$x_i^F = \begin{cases} 0, & \text{Si se cierra la instalación } i \in I^F \\ 1, & \text{En otro caso} \end{cases}$$

Adicional a las variables de decisión definidas anteriormente, considere las siguientes variables auxiliares de asignación que nos van a ayudar a definir el tercer nivel del problema.

Variables auxiliares de asignación:

$$y_{ij}^L = \begin{cases} 1, & \text{Si el cliente } j \text{ es servido por la instalación } i \in I^L \\ 0, & \text{En otro caso} \end{cases}$$

$$y_{ij}^F = \begin{cases} 1, & \text{Si el cliente } j \text{ es servido por la instalación } i \in I^F \\ 0, & \text{En otro caso} \end{cases}$$

El modelo matemático de programación tri-nivel es como sigue:

$$\max_{x^L} \sum_{i \in I^L} \sum_{j \in J} b_j y_{ij}^L \quad (3.16)$$

$$\text{Sujeto a: } \sum_{i \in I^L} (1 - x_i^L) = p \quad (3.17)$$

$$x_i^L \in \{0, 1\} \quad \forall i \in I^L \quad (3.18)$$

donde  $x^F$  resuelve:

$$\max_{x^F} \sum_{i \in I^F} \sum_{j \in J} b_j y_{ij}^F \quad (3.19)$$

$$\text{Sujeto a: } \sum_{i \in I^F} (1 - x_i^F) = r \quad (3.20)$$

$$x_i^F \in \{0, 1\} \quad \forall i \in I^F \quad (3.21)$$

y además  $y_{ij}^L, y_{ij}^F$  resuelven:

$$\min_{y_{ij}^L, y_{ij}^F} \sum_{i \in I} \sum_{j \in J} d_{ij} (y_{ij}^L + y_{ij}^F) \quad (3.22)$$

$$\text{Sujeto a: } y_{ij}^L \leq x_i^L \quad \forall i \in I^L, \forall j \in J \quad (3.23)$$

$$y_{ij}^F \leq x_i^F \quad \forall i \in I^F, \forall j \in J \quad (3.24)$$

$$\sum_{i \in I} (y_{ij}^L + y_{ij}^F) = 1 \quad \forall j \in J \quad (3.25)$$

$$y_{ij}^L, y_{ij}^F \in \{0, 1\} \quad \forall i \in I, \forall j \in J \quad (3.26)$$

El problema del primer nivel está definido por (3.16) – (3.19), donde (3.16) es la función objetivo del primer nivel que busca maximizar la demanda capturada de clientes en el mercado, (3.17) indica que la empresa con más jerarquía debe cerrar  $p$  instalaciones, (3.18) establece que las variables de decisión del primer nivel deben ser binarias. La restricción (3.19) indica que las variables  $x^F$  son controladas por el segundo nivel que está definido por las restricciones (3.19) – (3.22), (3.19) es la función objetivo del segundo nivel que busca maximizar la demanda capturada de clientes en el mercado, en otras palabras robar la mayor cantidad de demanda de los clientes al primer nivel. La restricción (3.20) indica que la empresa con menos jerarquía debe cerrar  $r$  instalaciones, y (3.21) establece que las variables de decisión del segundo nivel deben ser binarias. Así mismo, la restricción (3.22) indica que las variables  $y_{ij}^L, y_{ij}^F$  son controladas por un tercer nivel definido por las restricciones (3.22) – (3.26). Las restricciones (3.22) indican que los clientes buscan minimizar la distancia hacia las instalaciones que les va a satisfacer la demanda. Las restricciones (3.23) y (3.24) indican que un cliente se asignará a una instalación que permanece abierta. Además, (3.25) indica que un cliente se asignará solo a una instalación, y (3.26) indican que las variables de asignación son binarias. Con esto, la formulación (3.16) – (3.26) da lugar al modelo de programación tri-nivel.

### 3.3.1 MODELO MATEMÁTICO DE PROGRAMACIÓN BI-NIVEL PARA EL PROBLEMA $\overline{(r|p)}$ –CENTROIDE

Como mencionamos anteriormente, debido a la complejidad de un modelo tri-nivel, buscamos reducir este modelo a uno más tratable, es decir reducir el modelo de programación tri-nivel en uno de programación bi-nivel.

Para el modelo de programación bi-nivel que vamos a presentar a continuación tome en cuenta los mismos parámetros y conjuntos definidos

anteriormente para el problema de programación tri-nivel. Así como el siguiente conjunto de índices:  $I_j(x^L) = \{i \in I^F : d_{ij} \leq \min_{k \in I^L | x_k^L=1} d_{kj}\}$  el cual nos ayuda a reducir el problema tri-nivel en uno bi-nivel, donde el primer nivel está asociado a la empresa del líder y el segundo nivel a la empresa del seguidor. Este conjunto guarda los índices de las instalaciones del seguidor que pueden capturar la demanda del cliente  $j \in J$  al cual se le cerró la instalación del líder a la cual estaba asignado.

Entonces las variables de decisión son las siguientes:

Del líder:

$$x_i^L = \begin{cases} 0, & \text{Si se cierra la instalación } i \in I^L \\ 1, & \text{En otro caso} \end{cases}$$

Del seguidor:

$$x_i^F = \begin{cases} 0, & \text{Si se cierra la instalación } i \in I^F \\ 1, & \text{En otro caso} \end{cases}$$

Considere las siguientes variables auxiliares de asignación para el problema de programación bi-nivel:

$$y_j = \begin{cases} 0, & \text{Si el cliente } j \text{ es servido por el líder} \\ 1, & \text{Si el cliente } j \text{ es servido por el seguidor} \end{cases}$$

El modelo de programación bi-nivel es como sigue:

$$\max_{x^L} \sum_{j \in J} b_j(1 - y_j) \quad (3.27)$$

$$\text{sujeto a: } \sum_{i \in I^L} (1 - x_i^L) = p \quad (3.28)$$

$$x_i^L \in \{0, 1\} \quad \forall i \in I^L \quad (3.29)$$

Donde  $x^F, y_j$  resuelven:

$$\text{máx} \quad \sum_{j \in J} b_j y_j \quad (3.30)$$

$$\text{subject to:} \quad \sum_{i \in I^F} (1 - x_i^F) = r \quad (3.31)$$

$$y_j \leq \sum_{i \in I_j(x^L)} x_i^F \quad \forall j \in J \quad (3.32)$$

$$x_i^F \in \{0, 1\}, y_j \in \{0, 1\} \quad \forall i \in I^F, \forall j \in J \quad (3.33)$$

La función objetivo del líder (3.27) busca capturar la demanda máxima. La ecuación (3.28) garantiza que el líder solo cerrará  $p$  instalaciones, (3.29) establece que las variables de decisión del primer nivel deben ser binarias. La ecuación (3.30) es lo que convierte al problema en uno de programación bi-nivel, ahí se indica que las variables del nivel inferior deben resolver a optimalidad el problema (3.30) – (3.33). En la función objetivo del seguidor, se expresa el objetivo de capturar la demanda máxima. La ecuación (3.31) garantiza que el seguidor solo cerrará  $r$  instalaciones. Las restricciones (3.32) determinan la demanda cedida por los tomadores de decisiones y (3.33) indican que las variables de decisión del seguidor así como de las variables auxiliares de asignación son binarias. Si el seguidor no tiene instalaciones en el conjunto  $I_j(x^L)$ , el cliente  $j$  es servido por el líder. De este modo, la formulación (3.27)-(3.33) es un problema bi-nivel entero.

Es importante hacer notar que en los modelo bi-nivel y tri-nivel presentados anteriormente, los clientes se asignan a la instalación más cercana una vez que se les cierra la instalación a la que estaba asignado en un principio, pero el problema central de esta tesis es el que se presentará a continuación, en donde tomamos en cuenta la lealtad de los clientes al agregar restricciones que nos ayudan a cumplir dicha característica.

## 3.4 EL PROBLEMA $\overline{(r|p)}^{\mathcal{L}}$ – CENTROIDE

### 3.4.1 SUPUESTOS

A continuación se enlistan los supuestos básicos considerados en este problema:

- Hay dos empresas operando actualmente en el mercado y existe una jerarquía entre ellas.
- Los clientes están asignados a la instalación más cercana, lo cual define a cuál empresa van a ser leales. Tenga en cuenta que consideramos clientes conservadores, es decir, si un cliente tiene a la misma distancia una instalación de cada empresa, este se asigna a la empresa con mayor jerarquía.
- Cada empresa ofrece el mismo servicio al mismo precio. Además, está prohibido que se oferten nuevos servicios en el mercado. Esto con el fin de evitar que los clientes asuman otra postura determinada por descuentos de las empresas, servicios de mejor calidad que los de la otra empresa, etc.
- El líder tiene que cerrar  $p$  instalaciones sabiendo que el seguidor también va a cerrar  $r$  instalaciones.
- Las instalaciones serán cerradas definitivamente, no se permite el cierre temporal ya sea por remodelación, cuestiones climatologías, por accidentes laborales, o por daños de cualquier índole en la instalación.
- No se permite la reubicación de una instalación, entiéndase deslocalización al cierre definitivo de ésta y no confundirse con el termino deslocalización utilizado en el sector económico.

- Un cliente puede cambiar de empresa siempre y cuando no exista otra instalación abierta de la empresa a la que es leal dentro de su radio de lealtad, en este caso, se asigna a la instalación más cercana.

### 3.4.2 RESTRICCIONES VÁLIDAS DE LEALTAD PARA LOS CLIENTES

En la actualidad, es muy común que los clientes no se asignen a la instalación abierta más cercana, hay diversos artículos en la literatura en los cuáles la asignación de los clientes se hace en base a una preferencia, por ejemplo en Hanjoul & Peeters (1987), Cánovas *et al.* (2007), Camacho-Vallejo *et al.* (2014) por mencionar algunos. Otras modelaciones del comportamiento de los clientes se pueden encontrar en Drezner & Eiselt (2002).

Como se mencionó anteriormente, los clientes están asignados a las instalaciones abiertas más cercanas. Esto da lugar a definir la lealtad de dicho cliente a alguna de las dos empresas. Note que, a final de cuentas, la lealtad viene a actuar como un criterio de preferencias de los clientes sobre alguna de las empresas. Por eso nosotros vamos a agregar restricciones válidas que nos ayuden a satisfacer las restrcciones de lealtad de los clientes.

Primeramente tome en cuenta los parámetros y conjuntos auxiliares de ordenamiento siguientes:

#### **Parámetros.**

$\rho_j$ : Radio de lealtad del cliente  $j \in J$ .

$S_k$ : Elemento de un conjunto cualesquiera  $S$  en la  $k$ -ésima posición.

#### **Conjuntos.**

$I^f(j)$ : Conjunto de las instalaciones ordenadas dentro del radio de lealtad del cliente  $j \in J$ . Donde  $I^f(j) = \{i \in I^f : d_{ij} \leq \rho_j\}$  con  $f \in \{L, F\}$  (Líder-Seguidor, según sea el caso). Los elementos de  $I^f(j)$  cumplen que  $d_{I^f(j)_{k-1},j} \leq d_{I^f(j)_k,j}$  Para  $k = 2, \dots, |I^f(j)|$ .

$I(j)$ : Conjunto ordenado de todas las instalaciones dentro del radio de lealtad para el cliente  $j \in J$ . Donde:

$$I(j) = \begin{cases} I^L(j) \cup I^F(j), & i^*(j) \in I^L \\ I^F(j) \cup I^L(j), & i^*(j) \in I^F \end{cases}$$

e  $i^*(j) = \operatorname{argmin}_{i \in I} \{d_{ij}\}$ .

$\bar{I}(j)$ : Conjunto ordenado de las instalaciones para el cliente  $j \in J$  fuera del radio de lealtad.  $\bar{I}(j) = \{i \in I^f : d_{ij} > \rho_j\}$  con  $f \in \{L, F\}$ . Los elementos de  $\bar{I}(j)$  satisfacen  $d_{\bar{I}(j)_{k-1}, j} \leq d_{\bar{I}(j)_k, j}$  Para  $k = 2, \dots, |\bar{I}(j)|$  y  $j \in J$ .

$O(j)$ : Conjunto de las instalaciones del cliente  $j \in J$  considerando lealtad, donde  $O(j) = \{I(j) \cup \bar{I}(j)\}$ .

$\tilde{I}^f(j)$ : Conjunto ordenado de las instalaciones de la empresa  $f \in \{L, F\}$  para el cliente  $j \in J$ . Es decir  $\tilde{I}^f(j) = I^f(j) \cup (\bar{I}(j) \cap I^f)$ ,  $\forall j \in J$  para  $f \in \{L, F\}$

Además defina la siguiente función auxiliar como:

### **Función Auxiliar**

$P_O(S_k)$ : Función inyectiva que regresa la posición del conjunto ordenado  $O(j)$  en donde se encuentra el  $k$ -ésimo elemento del conjunto  $S$ , o sea  $P_{O(S_k)} = h \Leftrightarrow O_h = S_k$

Por otro lado, el radio de lealtad  $\rho_j$  del cliente  $j \in J$ , es definido como sigue:

$$\rho_j = \begin{cases} \delta^L \times \min\{d_{ij}\}, & \text{Si el cliente } j \text{ es servido por la instalación } i \in I^L \\ \delta^S \times \min\{d_{ij}\}, & \text{Si el cliente } j \text{ es servido por la instalación } i \in I^F \end{cases}$$

donde  $\delta^L$  y  $\delta^S$  son constantes enteras utilizadas para tener una medida proporcional de la distancia mínima. Esto se hace con el fin de definir cuál

es la máxima distancia que está dispuesto a recorrer un cliente al que se le cerró la instalación a la cual estaba asignado. Es importante hacer notar que se toma la  $\min\{d_{ij}\} \neq 0$  con el fin de evitar radios de lealtad iguales a cero. Más adelante, en este capítulo se mostrará un ejemplo ilustrativo para tener un panorama más claro de cómo funcionan dichos parámetros al momento de resolver el problema considerando la lealtad.

Estos parámetros y conjuntos descritos anteriormente nos permiten agregar las restricciones (3.43) y (3.44), las cuales son presentadas en la siguiente sección. Hay que recalcar que dichas restricciones nos ayudan a tomar en cuenta la asignación de los clientes a las instalaciones considerando lealtad.

### 3.4.3 MODELO DE PROGRAMACIÓN BI-NIVEL PARA EL PROBLEMA $\overline{(r/p)}^{\mathcal{L}}$ – CENTROIDE

Para el modelo de programación bi-nivel que vamos a presentar a continuación tome en cuenta los mismos parámetros y conjuntos definidos anteriormente para el problema de programación tri-nivel presentado en la sección 3.3 así como los parámetros y conjuntos auxiliares considerados al definir la lealtad de los clientes.

Entonces las variables de decisión son las siguientes:

Del líder:

$$x_i^L = \begin{cases} 0, & \text{Si se cierra la instalación } i \in I^L \\ 1, & \text{En otro caso} \end{cases}$$

Del seguidor:

$$x_i^F = \begin{cases} 0, & \text{Si se cierra la instalación } i \in I^F \\ 1, & \text{En otro caso} \end{cases}$$

Considere las siguientes variables auxiliares de asignación para el problema de programación bi-nivel:

$$y_j = \begin{cases} 0, & \text{Si el cliente } j \text{ es servido por el líder} \\ 1, & \text{Si el cliente } j \text{ es servido por el seguidor} \end{cases}$$

$$z_{ij} = \begin{cases} 1, & \text{Si el cliente } j \text{ es servido por la instalación } i \\ 0, & \text{En otro caso} \end{cases}$$

El modelo resultante de programación bi-nivel es como sigue:

$$\max_{x^L} \sum_{j \in J} b_j(1 - y_j) \quad (3.34)$$

$$\text{subject to: } \sum_{i \in I^L} (1 - x_i^L) = p \quad (3.35)$$

$$x_i^L \in \{0, 1\} \quad \forall i \in I^L \quad (3.36)$$

donde  $x^F, y, z$  resuelven:

$$\max \sum_{j \in J} b_j y_j \quad (3.37)$$

$$\text{subject to: } \sum_{i \in I^F} (1 - x_i^F) = r \quad (3.38)$$

$$\sum_{i \in I} z_{ij} = 1 \quad \forall j \in J \quad (3.39)$$

$$z_{ij} \leq x_i^L \quad \forall i \in I^L, \forall j \in J \quad (3.40)$$

$$z_{ij} \leq x_i^F \quad \forall i \in I^F, \forall j \in J \quad (3.41)$$

$$\sum_{i \in I^L} z_{ij} \leq 1 - y_j \quad \forall j \in J \quad (3.42)$$

$$x_{\tilde{I}^L(j)_k}^L + \sum_{\ell=P_O(\tilde{I}^L(j)_k)+1}^{|\mathcal{O}(j)|} z_{\mathcal{O}(j)_\ell j} \leq 1 \quad \forall k = 1, \dots, |\tilde{I}^L(j)|, j \in J \quad (3.43)$$

$$x_{\tilde{I}^F(j)_k}^F + \sum_{\ell=P_O(\tilde{I}^F(j)_k)+1}^{|\mathcal{O}(j)|} z_{\mathcal{O}(j)_\ell j} \leq 1 \quad \forall k = 1, \dots, |\tilde{I}^F(j)|, j \in J \quad (3.44)$$

$$x_i^F \in \{0, 1\} \quad \forall i \in I^F \quad (3.45)$$

$$y_j \in \{0, 1\} \quad \forall j \in J \quad (3.46)$$

$$z_{ij} \in \{0, 1\} \quad \forall i \in I, \forall j \in J \quad (3.47)$$

La función objetivo del líder (3.34) busca capturar la demanda máxima. La ecuación (3.35) garantiza que el líder solo cerrará  $p$  instalaciones. La ecuación (3.37) es lo que convierte al problema en uno de programación bi-nivel, ahí se indica que las variables del nivel inferior deben resolver a optimalidad el problema (3.37) – (3.47). Así mismo, se expresa la función objetivo del seguidor (capturar la demanda máxima). La ecuación

(3.38) garantiza que el seguidor solo cerrará  $r$  instalaciones. Las restricciones (3.39) garantizan que el cliente  $j$  sea servido solamente por una instalación. Las restricciones (3.40) y (3.41) indican que los clientes solamente pueden ser asignados a instalaciones que permanecen abiertas ya sea del líder o seguidor. Las restricciones (3.42) indican que si el cliente  $j$  es servido por el líder entonces el cliente  $j$  no puede ser servido por el seguidor, esta restricción relaciona la variable auxiliar de asignación con la variable  $y$ . Las restricciones (3.43) y (3.44) garantizan que el cliente  $j$  no va a cambiar de empresa, siempre y cuando haya instalaciones abiertas de la empresa a la que es leal dentro de su radio de lealtad. Las restricciones (3.36), (3.45), (3.46) y (3.47) indican la naturaleza binaria de las variables de decisión, así como de las variables auxiliares de asignación. Con esto, la formulación (3.34) – (3.47) es un modelo de programación bi-nivel binaria.

### 3.5 RESTRICCIONES DE LEALTAD: UN EJEMPLO

#### ILUSTRATIVO.

Con el fin de ilustrar cómo funcionan los conjuntos de ordenamiento y las restricciones de lealtad de los clientes (3.43) y (3.44), vamos a presentar un ejemplo ilustrativo pequeño. Considere el caso en el cual se tiene  $|I^L| = |I^F| = 4$  y  $|J| = 10$ , es decir, se tienen actualmente 4 instalaciones abiertas del líder y 4 del seguidor, respectivamente. Además, hay 10 clientes asignados a ellas. Para este ejemplo, considere que el cliente  $j = 1$  está dispuesto a recorrer a lo mucho el doble de la distancia, en el caso de que se cierre la instalación que le brinda servicio. Es decir  $\delta^L = \delta^S = 2$ . En las tablas 3.1 y 3.2 se muestran las coordenadas de los clientes e instalaciones del líder y seguidor. Además, las distancias  $d_{ij}$  de ir de la instalación  $i \in I$  al cliente  $j \in J$  se muestran en la tabla 3.3. En azul se denota la mínima distancia entre cada cliente y la instalación que le presta el servicio, es decir,  $\forall j \in J, \min_{i \in I} \{d_{ij}\}$ , es decir, la distancia a la instalación más cercana para el cliente  $j \in J$ . Dichas distancias nos ayudan a identificar

la instalación  $i \in I$  a la que está asignado actualmente el cliente, y por consecuencia, se identifica la empresa a la va a ser leal. Por ejemplo, el cliente  $j = 1$  actualmente está asignado a la instalación  $i = 2 \in I^L$ , pues  $\min_{i \in I} \{d_{i1}\} = d_{21}$ . En ese caso  $d_{21} = 3.16$ , entonces dicho cliente va a ser leal al líder pues  $2 \in I^L$ . Además, dicha distancia nos va a ayudar a calcular el radio de lealtad como lo definimos en la subsección anterior.

En la tabla 3.4 se muestran los radios de lealtad de cada cliente, los cuales también nos ayudan a llenar nuestros conjuntos auxiliares. En este caso, como  $j = 1$  su  $i^*(1) = \operatorname{argmin}_{i \in I} \{d_{i1}\} = 2 \in I^L$ . Para  $I^f(j)$  con  $f \in \{L, F\}$  (Líder, Seguidor) tenemos que  $I^L(1) = \{i \in I^L : d_{i1} \leq \rho_1\}$ ,  $I^F(1) = \{i \in I^F : d_{i1} \leq \rho_1\}$ , además tenemos que  $I(1) = \{I^L(j) \cup I^F(j)\}$  para obtener el conjunto  $O(1) = \{I(1) \cup \bar{I}(j)\}$ . Por un análisis sencillo, podemos ver que  $d_{21} \leq d_{41} \leq \rho_1$  y  $d_{51} \leq d_{71} \leq \rho_1$  por lo que  $I^L(1) = \{2, 4\}$ ,  $I^F(1) = \{5, 7\}$  respectivamente, entonces  $I(1) = \{2, 4\} \cup \{5, 7\} = \{2, 4, 5, 7\}$ . Así tenemos que las instalaciones que están fuera del radio de lealtad  $\rho_1 < d_{31} \leq d_{11} \leq d_{61} \leq d_{81}$ ,  $\bar{I}(1) = \{3, 1, 6, 8\}$ . Entonces obtenemos nuestro conjunto  $O(1) = \{I(1) \cup \bar{I}(j)\} = \{\{2, 4, 5, 7\} \cup \{3, 1, 6, 8\}\} = \{2, 4, 5, 7, 3, 1, 6, 8\}$ , por último calculamos los conjuntos de ordenamiento por empresa para el cliente  $j = 1$ , es decir  $\tilde{I}^f(1)$  con  $f \in \{L, F\}$ , entonces  $\tilde{I}^L(1) = I^L(1) \cup (\bar{I}(1) \cap I^L) = \{2, 4\} \cup (\{3, 1, 6, 8\} \cap \{1, 2, 3, 4\}) = \{2, 4, 3, 1\}$ ,  $\tilde{I}^F(1) = I^F(1) \cup (\bar{I}(1) \cap I^F) = \{5, 7\} \cup (\{3, 1, 6, 8\} \cap \{5, 6, 7, 8\}) = \{5, 7, 6, 8\}$ , siguiendo un proceso análogo para los demás clientes. Al final se obtiene la tabla 3.5, donde la leyenda  $L$  significa que el cliente  $j$  es leal al líder o  $F$  si es leal al seguidor, es decir  $i^*(j) \in I^L$  ó  $i^*(j) \in I^F$  respectivamente.

<b>j-Cliente</b>	<b>x</b>	<b>y</b>
1	4	-1
2	-4	1
3	5	1
4	-7	7
5	-1	-5
6	6	7
7	1	7
8	4	-7
9	-10	2
10	7	-7

Tabla 3.1: Coordenadas de los clientes  $j \in J$ 

<b>I</b>	<b>x</b>	<b>y</b>
1	7	7
2	3	-4
3	-1	4
4	-1	-4
5	8	-4
6	-6	7
7	7	4
8	-10	4

Tabla 3.2: Coordenadas de las instalaciones  $i \in I$ 

<b><math>d_{ij}</math></b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7</b>	<b>8</b>	<b>9</b>	<b>10</b>
1	8.54	12.53	6.32	14	14.42	<b>1</b>	6	14.32	17.72	14
2	<b>3.16</b>	8.60	5.39	14.87	4.12	11.40	11.18	<b>3.16</b>	14.31	5
3	7.07	<b>4.24</b>	6.71	6.71	9.00	<b>7.62</b>	<b>3.61</b>	12.08	9.21	13.60
4	5.83	5.83	7.81	12.53	<b>1</b>	13.04	11.18	5.83	10.82	8.54
5	5	13	5.83	18.60	9.05	11.18	13.04	5	18.97	<b>3.16</b>
6	12.80	6.32	12.53	<b>1</b>	13	12	7	17.20	6.40	19.10
7	5.83	11.40	<b>3.60</b>	14.32	12.04	3.16	6.71	11.40	17.12	11
8	14.87	6.71	15.30	4.24	12.73	16.28	11.40	17.80	<b>2</b>	20.25

Tabla 3.3: Distancia de ir del cliente  $j \in J$  a la instalación  $i \in I$

$J$	$\rho_j$
1	$2 \times d_{21} = 6.32$
2	$2 \times d_{32} = 8.48$
3	$2 \times d_{73} = 7.2$
4	$2 \times d_{64} = 2$
5	$2 \times d_{45} = 2$
6	$2 \times d_{16} = 2$
7	$2 \times d_{37} = 7.22$
8	$2 \times d_{28} = 6.32$
9	$2 \times d_{89} = 4$
10	$2 \times d_{510} = 6.32$

Tabla 3.4: Radios de lealtad

$J$	Leal A:	$I^L(j)$	$I^F(j)$	$I(j)$	$\bar{I}(j)$	$\tilde{I}^L(j)$	$\tilde{I}^F(j)$	$O(j)$
1	L	{2, 4}	{5, 7}	{2, 4, 5, 7}	{3, 1, 6, 8}	{2, 4, 3, 1}	{5, 7, 6, 8}	{2, 4, 5, 7, 3, 1, 6, 8}
2	L	{3, 4}	{6, 8}	{3, 4, 6, 8}	{2, 7, 1, 5}	{3, 4, 2, 1}	{6, 8, 7, 5}	{3, 4, 6, 8, 2, 7, 1, 5}
3	F	{2, 1, 3}	{7, 5}	{7, 5, 2, 1, 3}	{4, 6, 8}	{2, 1, 3, 4}	{7, 5, 6, 8}	{7, 5, 2, 1, 3, 4, 6, 8}
4	F	$\emptyset$	{6}	{6}	{8, 3, 4, 1, 7, 2, 5}	{3, 4, 1, 2}	{6, 8, 7, 5}	{6, 8, 3, 4, 1, 7, 2, 5}
5	L	{4}	$\emptyset$	{4}	{2, 3, 5, 7, 8, 6, 1}	{4, 2, 3, 1}	{5, 7, 8, 6}	{4, 2, 3, 5, 7, 8, 6, 1}
6	L	{1}	$\emptyset$	{1}	{7, 3, 5, 2, 6, 4, 8}	{1, 3, 2, 4}	{7, 5, 6, 8}	{1, 7, 3, 5, 2, 6, 4, 8}
7	L	{3, 1}	{7, 6}	{3, 1, 7, 6}	{4, 2, 8, 5}	{3, 1, 4, 2}	{7, 6, 8, 5}	{3, 1, 7, 6, 4, 2, 8, 5}
8	L	{2, 4}	{5}	{2, 4, 5}	{7, 3, 1, 6, 8}	{2, 4, 3, 1}	{5, 7, 6, 8}	{2, 4, 5, 7, 3, 1, 6, 8}
9	F	$\emptyset$	{8}	{8}	{6, 3, 4, 2, 7, 1, 5}	{3, 4, 2, 1}	{8, 6, 7, 5}	{8, 6, 3, 4, 2, 7, 1, 5}
10	F	{2}	{5}	{5, 2}	{4, 7, 3, 1, 6, 8}	{2, 4, 3, 1}	{5, 7, 6, 8}	{5, 2, 4, 7, 3, 1, 6, 8}

 Tabla 3.5: Tabla conjuntos de ordenamiento para cada cliente  $j \in J$ 

Vamos a escribir las restricciones de lealtad de forma explícita para el cliente  $j = 1$ :

Para la restricción (3.43) tenemos que:

$$x_{\tilde{I}^L(1)_k}^L + \sum_{\ell=P_O(\tilde{I}^L(1)_k)+1}^{|O(1)|} z_{O(1)_\ell 1} \leq 1 \quad \forall k = 1, \dots, |\tilde{I}^L(1)|$$

- Si  $k = 1$  tenemos que:  $\tilde{I}^L(1)_1 = 2$  y  $P_O(\tilde{I}^L(1)_1) = P_O(2) = 1$  entonces:

$$x_{\tilde{I}^L(1)_1}^L + \sum_{\ell=P_O(\tilde{I}^L(1)_1)+1}^{|O(1)|} z_{O(1)_\ell 1} \leq 1$$

$$\begin{aligned}
x_2^L + \sum_{\ell=P_O(2)+1}^{|O(1)|} z_{O(1)\ell 1} &\leq 1 \\
x_2^L + z_{O(1)21} + z_{O(1)31} + z_{O(1)41} + z_{O(1)51} + z_{O(1)61} + z_{O(1)71} + z_{O(1)81} &\leq 1 \\
x_2^L + z_{41} + z_{51} + z_{71} + z_{31} + z_{11} + z_{61} + z_{81} &\leq 1 \quad (3.48)
\end{aligned}$$

- Si  $k = 2$  tenemos que:  $\tilde{I}^L(1)_2 = 4$  y  $P_O(\tilde{I}^L(1)_2) = P_O(4) = 2$  entonces:

$$\begin{aligned}
x_{\tilde{I}^L(1)_2}^L + \sum_{\ell=P_O(\tilde{I}^L(1)_2)+1}^{|O(1)|} z_{O(1)\ell 1} &\leq 1 \\
x_4^L + \sum_{\ell=P_O(2)+1}^{|O(1)|} z_{O(1)\ell 1} &\leq 1 \\
x_4^L + z_{O(1)31} + z_{O(1)41} + z_{O(1)51} + z_{O(1)61} + z_{O(1)71} + z_{O(1)81} &\leq 1 \\
x_4^L + z_{51} + z_{71} + z_{31} + z_{11} + z_{61} + z_{81} &\leq 1 \quad (3.49)
\end{aligned}$$

- Si  $k = 3$  tenemos que:  $\tilde{I}^L(1)_3 = 3$  y  $P_O(\tilde{I}^L(1)_3) = P_O(3) = 5$  entonces:

$$\begin{aligned}
x_{\tilde{I}^L(1)_3}^L + \sum_{\ell=P_O(\tilde{I}^L(1)_3)+1}^{|O(1)|} z_{O(1)\ell 1} &\leq 1 \\
x_3^L + \sum_{\ell=P_O(3)+1}^{|O(1)|} z_{O(1)\ell 1} &\leq 1 \\
x_3^L + z_{O(1)61} + z_{O(1)71} + z_{O(1)81} &\leq 1 \\
x_3^L + z_{11} + z_{61} + z_{81} &\leq 1 \quad (3.50)
\end{aligned}$$

- Si  $k = 4$  tenemos que:  $\tilde{I}^L(1)_4 = 1$  y  $P_O(\tilde{I}^L(1)_4) = P_O(1) = 6$  entonces:

$$x_{\tilde{I}^L(1)_4}^L + \sum_{\ell=P_O(\tilde{I}^L(1)_4)+1}^{|O(1)|} z_{O(1)\ell 1} \leq 1$$

$$\begin{aligned}
x_1^L + \sum_{\ell=P_O(1)+1}^{|O(1)|} z_{O(1)\ell 1} &\leq 1 \\
x_1^L + z_{O(1)71} + z_{O(1)81} &\leq 1 \\
x_1^L + z_{61} + z_{81} &\leq 1
\end{aligned} \tag{3.51}$$

Para la restriccion (3.44) tenemos que:

$$x_{\tilde{I}^F(1)_k}^F + \sum_{\ell=P_O(\tilde{I}^F(1)_k)+1}^{|O(1)|} z_{O(1)\ell 1} \leq 1 \quad \forall k = 1, \dots, |\tilde{I}^F(1)|$$

- Si  $k = 1$  tenemos que:  $\tilde{I}^F(1)_1 = 5$  y  $P_O(\tilde{I}^F(1)_1) = P_O(5) = 3$  entonces:

$$\begin{aligned}
x_{\tilde{I}^F(1)_1}^L + \sum_{\ell=P_O(\tilde{I}^F(1)_1)+1}^{|O(1)|} z_{O(1)\ell 1} &\leq 1 \\
x_5^F + \sum_{\ell=P_O(5)+1}^{|O(1)|} z_{O(1)\ell 1} &\leq 1 \\
x_5^F + z_{O(1)41} + z_{O(1)51} + z_{O(1)61} + z_{O(1)71} + z_{O(1)81} &\leq 1 \\
x_5^F + z_{71} + z_{31} + z_{11} + z_{61} + z_{81} &\leq 1
\end{aligned} \tag{3.52}$$

- Si  $k = 2$  tenemos que:  $\tilde{I}^F(1)_2 = 7$  y  $P_O(\tilde{I}^F(1)_2) = P_O(7) = 4$  entonces:

$$\begin{aligned}
x_{\tilde{I}^F(1)_2}^F + \sum_{\ell=P_O(\tilde{I}^F(1)_2)+1}^{|O(1)|} z_{O(1)\ell 1} &\leq 1 \\
x_7^F + \sum_{\ell=P_O(7)+1}^{|O(1)|} z_{O(1)\ell 1} &\leq 1 \\
x_7^F + z_{O(1)51} + z_{O(1)61} + z_{O(1)71} + z_{O(1)81} &\leq 1 \\
x_7^F + z_{31} + z_{11} + z_{61} + z_{81} &\leq 1
\end{aligned} \tag{3.53}$$

- Si  $k = 3$  tenemos que:  $\tilde{I}^F(1)_3 = 6$  y  $P_O(\tilde{I}^F(1)_3) = P_O(6) = 7$  entonces:

$$\begin{aligned}
x_{\tilde{I}^F(1)_3}^F + \sum_{\ell=P_O(\tilde{I}^F(1)_3)+1}^{|O(1)|} z_{O(1)\ell 1} &\leq 1 \\
x_6^F + \sum_{\ell=P_O(6)+1}^{|O(1)|} z_{O(1)\ell 1} &\leq 1 \\
x_6^F + z_{O(1)8 1} &\leq 1 \\
x_6^F + z_{8 1} &\leq 1
\end{aligned} \tag{3.54}$$

- Si  $k = 4$  tenemos que:  $\tilde{I}^F(1)_4 = 8$  y  $P_O(\tilde{I}^F(1)_4) = P_O(8) = 8$  entonces:

$$\begin{aligned}
x_{\tilde{I}^F(1)_4}^F + \sum_{\ell=P_O(\tilde{I}^F(1)_4)+1}^{|O(1)|} z_{O(1)\ell 1} &\leq 1 \\
x_8^F + \sum_{\ell=P_O(8)+1}^{|O(1)|} z_{O(1)\ell 1} &\leq 1 \\
x_8^F &\leq 1
\end{aligned} \tag{3.55}$$

Tome en cuenta que por propiedades de sumatorias se cumple lo siguiente:

$$\sum_{\ell=8+1}^8 z_{O(1)\ell 1} = 0$$

Ahora vamos a proceder a explicar el funcionamiento de las restricciones (3.43) y (3.44), siga considerando al cliente  $j = 1$  para el cual se tiene que  $O(1) = \{2, 4, 5, 7, 3, 1, 6, 8\}$ . Suponga que el líder cierra la instalación  $i = 2$  y mantiene abierta la  $i = 4$ . Es decir,  $x_2^L = 0$  y  $x_4^L = 1$  entonces de la restricción (3.48) tenemos que:

$$z_{41} + z_{51} + z_{71} + z_{31} + z_{11} + z_{61} + z_{81} \leq 1 \tag{3.56}$$

y de la restricción (3.49) se sabe que  $z_{51} = z_{71} = z_{31} = z_{11} = z_{61} = z_{81} = 0$ . Además, como  $x_2^L = 0$ , por la restricción (3.40) tenemos que  $z_{2j} \leq x_2^L$ . En particular, para el cliente 1 tendremos que  $z_{21} \leq 0$ , entonces  $z_{21} = 0$ . Esto último es obvio, pues si se cerró la instalación 2 ningún cliente puede estar asignado a ella. Luego, de (3.56) tenemos que  $z_{41} \leq 1$  y por la restricción (3.39) para el cliente 1,  $z_{11} + z_{21} + z_{31} + z_{41} + z_{51} + z_{61} + z_{71} + z_{81} = 1$  tenemos que  $z_{41} = 1$ , es decir, a pesar de que se cerró la instalación 2, el cliente 1 se mantiene leal al líder al asignarse a la instalación 4 dentro de su radio de lealtad. Esto ocurre a pesar de que la instalación 5, la cual pertenece al seguidor, está más cerca que esta pues  $d_{51} \leq d_{41} < \rho_1$ . Por último, de la restricción (3.42) tenemos que  $z_{41} \leq 1 - y_1$ , entonces  $y_1 = 0$ , lo cual significa que el cliente 1 es atendido por el líder.

Estas restricciones se escriben de manera similar para los clientes restantes y adoptan el mismo comportamiento ayudando a satisfacer las restricciones de lealtad de los clientes.

ALGORITMO DE RAMIFICACIÓN Y  
ACOTAMIENTO PARA RESOLVER EL  
 $\overline{(r|p)}^{\mathcal{L}}$  –CENTROIDE

---

En este capítulo explicaremos a detalle el algoritmo de ramificación y acotamiento que se utilizó para resolver el problema  $\overline{(r|p)}$ -Centroide. El algoritmo implementado es una adaptación de uno que se encuentra en la literatura para resolver problemas de programación bi-nivel binaria. Por último, presentamos un ejemplo pequeño con el fin de ilustrar el funcionamiento y/o comportamiento de nuestro algoritmo.

#### 4.1 ALGORITMO DE RAMIFICACIÓN Y ACOTAMIENTO ADAPTADO PARA RESOLVER EL $\overline{(r|p)}^{\mathcal{L}}$ –CENTROIDE.

Para resolver nuestro problema  $\overline{(r|p)}^{\mathcal{L}}$  –Centroide, consideramos el algoritmo propuesto en Bard & Moore (1992). Ese algoritmo fue propuesto para resolver problemas de programación bi-nivel binarios, por lo que nosotros proponemos una versión adaptada de dicho algoritmo pero considerando las características de nuestro problema.

La idea es enumerar los nodos de un árbol binario asociado al problema del líder pero teniendo en cuenta la reacción óptima del seguidor. Es decir, el algoritmo encuentra la mejor decisión del líder entre las soluciones factibles de la región inducible  $IR$ .

Para iniciar el algoritmo, se busca tener una buena solución del líder. Esto se logra resolviendo el siguiente problema auxiliar de un solo nivel:

$$\text{máx} \quad \sum_{j \in J} b_j y_j \quad (4.1)$$

$$\text{subject to:} \quad \sum_{i \in I^L} (1 - x_i^L) = p \quad (4.2)$$

$$x_i^L \in \{0, 1\} \quad \forall i \in I^L \quad (4.3)$$

$$\sum_{i \in I^F} (1 - x_i^F) = r \quad (4.4)$$

$$\sum_{i \in I} z_{ij} = 1 \quad \forall j \in J \quad (4.5)$$

$$z_{ij} \leq x_i^L \quad \forall i \in I^L, \forall j \in J \quad (4.6)$$

$$z_{ij} \leq x_i^F \quad \forall i \in I^F, \forall j \in J \quad (4.7)$$

$$\sum_{i \in I^L} z_{ij} \leq 1 - y_j \quad \forall j \in J \quad (4.8)$$

$$x_{\tilde{I}^L(j)_k}^L + \sum_{\ell=P_O(\tilde{I}^L(j)_k)+1}^{|\mathcal{O}(j)|} z_{\mathcal{O}(j)_\ell j} \leq 1 \quad \forall k = 1, \dots, |\tilde{I}^L(j)|, j \in J \quad (4.9)$$

$$x_{\tilde{I}^F(j)_k}^F + \sum_{\ell=P_O(\tilde{I}^F(j)_k)+1}^{|O(j)|} z_{O(j)\ell j} \leq 1 \quad \forall k = 1, \dots, |\tilde{I}^F(j)|, j \in J \quad (4.10)$$

$$x_i^F \in \{0, 1\} \quad \forall i \in I^F \quad (4.11)$$

$$y_j \in \{0, 1\} \quad \forall j \in J \quad (4.12)$$

$$z_{ij} \in \{0, 1\} \quad \forall i \in I, \forall j \in J \quad (4.13)$$

$$\sum_{j \in J} b_j(1 - y_j) \geq \alpha \quad (4.14)$$

Es fácil ver que solamente agregamos las restricciones del líder al problema del segundo nivel. Además, la función objetivo del líder pasa a ser la restricción (4.14), la cual busca establecer una cota para su valor de la función objetivo, donde  $\alpha$  es un parámetro del modelo que se va determinando conforme avanza el algoritmo.

Antes de describir nuestro algoritmo, necesitamos definir algunos conjuntos y parámetros necesarios para su desarrollo. Además, para mejorar la lectura de la tesis, denote una solución del problema bi-nivel como un punto de la forma  $XYZ = (x^L, x^F, y, z)$  y la función objetivo del líder definida como en (3.34) será denotada por  $F = \sum_{j \in J} b_j(1 - y_j)$ .

### Conjuntos y Parámetros a utilizar en el algoritmo:

$W$ : Conjunto de índices asociados a las variables.  $W = \{1, \dots, |I^L|\}$

$P_k$ : El camino explorado en la iteación  $k$ .

$W_k$ : Conjunto de índices de las variables exploradas en la iteración  $k$ .

$S_k^+$ : Conjunto de índices de las variables exploradas asociadas a las instalaciones que se mantienen abiertas en la iteración  $k$ .  $S_k^+ = \{i : i \in W_k \text{ y } x_i^L = 1\}$

$S_k^-$ : Conjunto de índices de las variables exploradas asociadas a las instalaciones cerradas en la iteración  $k$ .  $S_k^- = \{i : i \in W_k \text{ y } x_i^L = 0\}$

$S_k^0$ : Conjunto de índices de las variables que no han sido exploradas en la iteración  $k$ .  $S_k^0 = \{i \in W : i \notin W_k\}$

$IR^k$ : Conjunto de puntos explorados en la región inducible

$\underline{F}$ : Cota inferior asociada al valor de la función objetivo del líder.  $\underline{F} = \text{máx}[F : XYZ \in IR^k]$ .

Es importante mencionar que solamente se va a ramificar en las variables de decisión del líder, esto es,  $x^L$ . Note que el algoritmo está diseñado para mejorar la decisión del líder pero considerando la reacción óptima del seguidor, esto es,  $x^F$ . Además, dado que la asignación de las variables  $z$  y  $y$  se hace basada en la lealtad preestablecida de los clientes hacia las instalaciones, estas variables toman menos importancia en el diseño del algoritmo. Al inicio del algoritmo se inicializan los parámetros de la siguiente manera:  $\alpha = -\infty$  y  $\underline{F} = -\infty$ .

El algoritmo propuesto queda como sigue:

Paso 1. Inicialización. Haga  $k := 0$ ,  $S_k^+ := \emptyset$ ,  $S_k^- := \emptyset$ ,  $S_k^0 := \{1, \dots, |I^L|\}$ ,  $\alpha := -\infty$  y  $\underline{F} := -\infty$ . Esto crea el nodo raíz del árbol de búsqueda.

Paso 2. Iteración general. Haga  $x_i^L = 0$  para  $i \in S_k^-$  y  $x_i^L = 1$  para  $i \in S_k^+$ . Con el valor actual de  $\alpha$ , intente encontrar una solución factible al problema auxiliar (4.1)–(4.14). Si se encuentra solución, haga  $k := k + 1$ , guarde

la solución  $XYZ^k$  y vaya al paso 3; en caso contrario, corte el nodo actual y vaya al paso 7.

Paso 3. Acotamiento. Haga  $x^L = (x_i^L)^k, \forall i \in I^L$  y resuelva el problema del nivel inferior (3.37) – (3.47) para obtener un punto

$\widehat{XYZ}^k = ((x^L)^k, (\hat{x}^F)^k, (\hat{y}), (\hat{z})^k) \in IR$ . Calcule  $F(\widehat{XYZ}^k)$  y haga  $\underline{F} := \max[\underline{F}, F(\widehat{XYZ}^k)]$ . Si  $F(\widehat{XYZ}^k) = F(\widehat{XYZ}^{k-1})$  vaya al paso 7.

Paso 4. Haga  $I = \{i \in S_{k-1}^0 : (x_i^L)^k = 0\}$ . Si  $I = \emptyset$ , haga  $S_k^+ := S_{k-1}^+$ ,  $S_k^- := S_{k-1}^-$ ,  $S_k^0 := S_{k-1}^0$ ,  $P_k := P_{k-1}$  y vaya al paso 6; en caso contrario, vaya al paso 5.

Paso 5. Ramificación. Crear  $|I|$  nuevos nodos como sigue: agregue  $i \in I$  al camino  $P_{k-1}$  en orden ascendente, uno por uno, para obtener nuevos nodos y el camino actualizado  $P_k$ ; haga  $S_k^+ := S_{k-1}^+$ ,  $S_k^0 := S_{k-1}^0 \setminus I$  y  $S_k^- := S_{k-1}^- \cup I$ .

Paso 6. Haga  $\alpha := \underline{F} + 1$ . Regrese al paso 2.

Paso 7. Retroceder. Si no existen nodos sin explorar, vaya al paso 8; en caso contrario, retroceda desde el nodo actual (esto es, al nodo sin explorar agregado más recientemente). Defina la variable correspondiente como  $i'$ , ramifique sobre su complemento al hacer  $(x_{i'}^L)^k = 1$ . Actualice los conjuntos  $S_k^+$ ,  $S_k^-$ ,  $S_k^0$ , y  $P_k$  de manera apropiada. Regrese al paso 2.

Paso 8. Terminación. Si  $\underline{F} = -\infty$ , entonces no hay solución factible para el problema bi-nivel definido por (3.34) – (3.47); en caso contrario, considere el punto factible asociado con  $\underline{F}$  como una solución óptima.

#### 4.1.1 EJEMPLO ILUSTRATIVO.

Con el fin de ilustrar el funcionamiento del algoritmo propuesto, vamos a presentar un ejemplo pequeño. Considere el caso en el cual se tiene que  $|I^L| = |I^F| = 10$  y  $|J| = 40$ , es decir, se tienen actualmente 10 instalaciones abiertas del líder y 10 del seguidor, respectivamente. Además,

hay 40 clientes asignados a dichas plantas. Para este ejemplo considere que  $p = r = 3$ , es decir, el líder va a cerrar 3 de sus 10 instalaciones sabiendo que el seguidor también va a cerrar 3 de sus 10 instalaciones.

Con el fin de agilizar la explicación del algoritmo, al momento de escribir la solución bi-nivel  $XYZ = (x^L, x^F, y, z)$  solamente mostraremos los valores correspondientes a  $x^L$  y  $x^F$ , es decir,  $XYZ = (x^L, x^F)$ .

A continuación se escriben de manera explícita algunos pasos del algoritmo:

Paso 1. (Inicialización) Hacer:

$$k := 0.$$

$$W := \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$$

$$W_0 := \emptyset$$

$$S_0^+ := \emptyset$$

$$S_0^- := \emptyset$$

$$S_0^0 := \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$$

$$\alpha := -\infty$$

$$\underline{F} := -\infty$$

Note que los conjuntos  $W_0$ ,  $S_0^+$  y  $S_0^-$  son iguales al conjunto vacío, esto se debe a que en un principio el algoritmo no ha explorado alguna variable. Luego, vamos al paso 2.

Paso 2. (Iteración general) Haga  $x_i^L = 0$  para  $i \in S_0^-$  y  $x_i^L = 1$  para  $i \in S_0^+$ , como estos conjuntos están vacíos en este momento (paso 1 anterior), no se agregan restricciones que aseguren que dichas instalaciones se mantengan abiertas y/o cerradas. Resolvemos el problema auxiliar con el valor actual de  $\alpha$  y se obtiene la siguiente solución:

$$XYZ = \begin{pmatrix} x^L \\ x^F \end{pmatrix} = \begin{pmatrix} (0, 0, 1, 1, 1, 1, 1, 0, 1, 1) \\ (1, 0, 0, 0, 1, 1, 1, 1, 1, 1) \end{pmatrix}$$

La cual indica que el líder va a cerrar las instalaciones 1, 2 y 8 (ver figura 4.1), es decir,  $x_1^L = x_2^L = x_8^L = 0$  y el seguidor va a cerrar las instalaciones 2, 3 y 4, es decir,  $x_2^F = x_3^F = x_4^F = 0$ .

Como se encontró una solución al problema auxiliar, incrementamos el contador  $k := k + 1 = 0 + 1 = 1$  y guardamos la solución de la siguiente manera:

$$\begin{pmatrix} x^L \\ x^F \end{pmatrix}^1 = \begin{pmatrix} (0, 0, 1, 1, 1, 1, 1, 0, 1, 1) \\ (1, 0, 0, 0, 1, 1, 1, 1, 1, 1) \end{pmatrix}^1$$

Como se encontró una solución, vamos al paso 3.

Paso 3. (Acotamiento) Hacer  $x^L = (x_i^L)^1 = (0, 0, 1, 1, 1, 1, 1, 0, 1, 1)^1$  y para ésta solución del líder resolvemos el problema del nivel inferior (3.37)–(3.47), y se obtiene un  $\widehat{XYZ}^1 = ((x^L)^1, (\hat{x}^F)^1, (\hat{y}), (\hat{z})^1) \in IR$ . Y obtenemos lo siguiente:

$$\widehat{XYZ}^1 = \begin{pmatrix} x^L \\ x^F \end{pmatrix}^1 = \begin{pmatrix} (0, 0, 1, 1, 1, 1, 1, 0, 1, 1) \\ (1, 1, 0, 0, 1, 0, 1, 1, 1, 1) \end{pmatrix}^1$$

Calculando  $F(\widehat{XYZ}^1)$  se obtiene que  $F(\widehat{XYZ}^1) = 7,793$  y hacemos  $\underline{F} := \max\{\underline{F}, F(\widehat{XYZ}^1)\} = \max\{-\infty, 7,793\} = 7,793$ . Luego de esto, vamos al paso 4 pues podemos ver que  $F(\widehat{XYZ}^1) \neq F(\widehat{XYZ}^0)$ .

Note que en este paso se obtiene la mejor solución bi-nivel para la la solución del líder  $(x_i^L)^1$ , además la solución del seguidor cambia a  $x_4^F = x_5^F = x_6^F = 0$ .

Paso 4. Haga  $I = \{i \in S_0^0 : (x_i^L)^1 = 0\} = \{1, 2, 8\}$ .

Si  $I = \emptyset$ , deje  $S_1^+ := S_{1-1}^+$ ,  $S_1^- := S_{1-1}^-$ ,  $S_1^0 := S_{1-1}^0$ ,  $P_1 := P_{1-1}$  y vaya al paso 6 (cosa que no pasa, pues el conjunto  $I$  es distinto de vacío), entonces vamos al paso 5.

Paso 5. (Ramificación) Crear  $|I|$  nuevos nodos como sigue: agregue  $i \in I$  al camino  $P_0$  en orden ascendente, uno por uno, para obtener nuevos

nodos y el camino actualizado  $P_1$ ; haga:

$$S_1^+ := S_0^+ = \emptyset$$

$$S_1^0 := S_0^0 \setminus I = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\} \setminus \{1, 2, 8\} = \{3, 4, 5, 6, 7, 9, 10\}$$

y

$$S_1^- := S_0^- \cup I = \emptyset \cup \{1, 2, 8\} = \{1, 2, 8\}$$

$$P_1 := P_0 = (1, 2, 8)$$

$$W_1 := \{1, 2, 8\}$$

Luego vamos al paso 6.

Paso 6. Haga  $\alpha := \underline{F} + 1 = 7,793 + 1 = 7,794$ . Regrese al paso 2.

Paso 2. (Iteración general). Mantenga  $x_i^L = 0$  para  $i \in S_1^-$  y  $x_i^L = 1$  para  $i \in S_1^+$ .

- Como  $S_1^+ = S_0^+ = \emptyset$  no necesitamos agregar la restricción  $x_i^L = 1$  para  $i \in S_1^+$ .
- Para  $x_i^L = 0$  con  $i \in S_1^- = \{1, 2, 8\}$ , tenemos que obligar al modelo auxiliar a mantener cerradas las instalaciones 1, 2 y 8 es decir agregar las siguientes restricciones  $x_1^L = x_2^L = x_8^L = 0$ . Pero como ya se tienen  $p = 3$  instalaciones cerradas, ya no es necesario explorar este nodo. Se corta el nodo actual y vamos al paso 7.

Paso 7. Como existen nodos  $P_1 = P_0 = (1, 2, 8)$ . Retroceder desde el nodo actual, es decir, el nodo aún no explorado creado más recientemente. Definimos la variable correspondiente como  $i' = 8$ , ramificamos sobre su complemento, es decir  $x_8^L = 1$  y actualizar  $S_1^+$ ,  $S_1^-$ ,  $S_1^0$ ,  $P_1$ , de manera apropiada:

$$S_1^+ := \{8\}$$

$$S_1^- := \{1, 2\}$$

$$S_1^0 := \{3, 4, 5, 6, 7, 9, 10\}$$

$$P_1 := (1, 2, \underline{8})$$

$$W_1 := \{1, 2, 8\}$$

Tome en cuenta que en el camino  $P_1$ , un número subrayado indica que este ya no es un nodo sin explorar. Mejor dicho, es el nodo al cual ya se ramificó en su complemento. Por ejemplo,  $i' = 8$ , es decir,  $\underline{8}$ . Adoptaremos este formato a lo largo del algoritmo.

Regrese al paso 2.

Paso 2. (Iteración general) Mantenga  $x_i^L = 0$  para  $i \in S_1^-$  y  $x_i^L = 1$  para  $i \in S_1^+$ . Es decir:

- Para los  $i \in S_1^+$  mantener  $x_i^L = 1$ , entonces  $x_{\underline{8}}^L = 1$ .
- Para los  $i \in S_1^-$  mantener  $x_i^L = 0$ , entonces  $x_1^L = x_2^L = 0$

Y con el valor actual de  $\alpha = 7,794$  intentamos encontrar una solución a la formulación auxiliar y obtenemos que:

$$XYZ = \begin{pmatrix} x^L \\ x^F \end{pmatrix} = \begin{pmatrix} (0, 0, 1, 1, 1, 0, 1, 1, 1, 1) \\ (1, 1, 0, 1, 1, 0, 1, 1, 1, 0) \end{pmatrix}$$

Y como se encontró una solución incrementamos el contador  $k := k + 1 = 1 + 1 = 2$  y guardamos la solución como:

$$\begin{pmatrix} x^L \\ x^F \end{pmatrix}^2 = \begin{pmatrix} (0, 0, 1, 1, 1, 0, 1, 1, 1, 1) \\ (1, 1, 0, 1, 1, 0, 1, 1, 1, 0) \end{pmatrix}^2$$

Al encontrarse solución, vamos al paso 3.

Paso 3. (Acotamiento) Hacer  $x^L = (x_i^L)^2 = (0, 0, 1, 1, 1, 0, 1, 1, 1, 1)$  y para esta solución del líder resolvemos el problema del nivel inferior (3.37) – (3.47), para obtener un  $\widehat{XYZ}^2 = ((x^L)^2, (\hat{x}^F)^2, (\hat{y}), (\hat{z})^2) \in IR$ . Y obtenemos lo siguiente:

$$\widehat{XYZ}^2 = \begin{pmatrix} x^L \\ x^F \end{pmatrix}^2 = \begin{pmatrix} (0, 0, 1, 1, 1, 0, 1, 1, 1, 1) \\ (0, 1, 1, 1, 1, 0, 1, 1, 1, 0) \end{pmatrix}^2$$

Entonces se calcula  $F(\widehat{XYZ}^2) = 8,072$  y hacemos  $\underline{F} := \max\{\underline{F}, F(\widehat{XYZ}^2)\} = \max\{7,793, 8,072\} = 8,072$ . Luego de esto, vamos al paso 4 ya que podemos apreciar que  $F(\widehat{XYZ}^2) \neq F(\widehat{XYZ}^1)$ .

Paso 4. Haga  $I = \{i \in S_1^0 : (x_i^L)^2 = 0\} = \{6\}$ .

Si  $I = \emptyset$ , deje  $S_2^+ := S_{2-1}^+$ ,  $S_2^- := S_{2-1}^-$ ,  $S_2^0 := S_{2-1}^0$ ,  $P_2 := P_{2-1}$  y vaya al paso 6. Pero como  $I$  es distinto del vacío, hay que ir al paso 5.

Paso 5. (Ramificación) Crear  $|I|$  nuevos nodos como sigue: agregue  $i \in I$  al camino  $P_1$  en orden ascendente, uno por uno, para obtener nuevos nodos y el camino actualizado  $P_2$ ; haga:

$$S_2^+ := S_1^+ = \{8\}$$

$$S_2^0 := S_1^0 \setminus I = \{3, 4, 5, 6, 7, 9, 10\} \setminus \{6\} = \{3, 4, 5, 7, 9, 10\} \text{ y}$$

$$S_2^- := S_1^- \cup I = \{1, 2\} \cup \{6\} = \{1, 2, 6\}$$

$$P_2 := (1, 2, \underline{8}, 6)$$

$$W_2 := \{8, 1, 2, 6\} = \{1, 2, 6, 8\}$$

Luego vamos al paso 6.

Paso 6. Haga  $\alpha := \underline{F} + 1 = 8,072 + 1 = 8,073$ .

Paso 2. (Iteración general) Mantenga  $x_i^L = 0$  para  $i \in S_2^-$  y  $x_i^L = 1$  para  $i \in S_2^+$ .

- Para  $x_i^L = 1$  con  $i \in S_2^+$ , agregar al modelo auxiliar  $x_8^L = 1$ .
- Para  $x_i^L = 0$  con  $i \in S_2^- = \{1, 2, 6\}$ , se tiene que garantizar que en el modelo auxiliar se cumpla lo siguiente:  $x_1^L = x_2^L = x_6^L = 0$ . Pero como ya se tienen  $p = 3$  instalaciones cerradas, ya no es necesario explorar este nodo. Cortamos el nodo actual y vamos al paso 7.

Paso 7. Como existen nodos  $P_2 = (1, 2, \underline{8}, 6)$ . Retroceder desde el nodo actual (nodo aún no explorado creado más recientemente). Definimos

la variable correspondiente como  $i' = 6$ , ramificamos sobre su complemento, es decir  $x_6^L = 1$  y actualizar  $S_2^+$ ,  $S_2^-$ ,  $S_2^0$ ,  $P_2$ , de manera apropiada:

$$S_2^+ := \{8, 6\}$$

$$S_2^- := \{1, 2\}$$

$$S_2^0 := \{3, 4, 5, 7, 9, 10\}$$

$$P_2 := (1, 2, \underline{8}, \underline{6})$$

$$W_2 := \{1, 2, 8, 6\} = \{1, 2, 6, 8\}$$

Regrese al paso 2.

Paso 2. (Iteración general) Se mantiene  $x_i^L = 0$  para  $i \in S_2^-$  y  $x_i^L = 1$  para  $i \in S_2^+$ .

- Para  $x_i^L = 1$  con  $i \in S_2^+ = \{8, 6\}$ , agregar al modelo auxiliar  $x_8^L = x_6^L = 1$ .
- Para  $x_i^L = 0$  con  $i \in S_2^- = \{1, 2\}$ , agregar  $x_1^L = x_2^L = 0$ .

Y con el valor actual de  $\alpha = 8,073$  intentamos encontrar una solución a la formulación auxiliar y obtenemos que:

$$XYZ = \begin{pmatrix} x^L \\ x^F \end{pmatrix} = \begin{pmatrix} (0, 0, 1, 0, 1, 1, 1, 1, 1, 1) \\ (1, 1, 0, 1, 0, 1, 1, 1, 1, 0) \end{pmatrix}$$

Y como se encontró una solución incrementamos el contador  $k := k + 1 = 2 + 1 = 3$  y guardamos la solución como:

$$\begin{pmatrix} x^L \\ x^F \end{pmatrix}^3 = \begin{pmatrix} (0, 0, 1, 0, 1, 1, 1, 1, 1, 1) \\ (1, 1, 0, 1, 0, 1, 1, 1, 1, 0) \end{pmatrix}^3$$

Al encontrarse una solución, vamos al paso 3.

Paso 3. (Acotamiento) Hacer  $x^L = (x_i^L)^3 = (0, 0, 1, 0, 1, 1, 1, 1, 1, 1)$  y para esta solución del líder se resuelve el problema del nivel inferior (3.37) –

(3.47), para obtener un  $\widehat{XYZ}^3 = ((x^L)^3, (\hat{x}^F)^3, (\hat{y}), (\hat{z})^3) \in IR$ . Y obtenemos lo siguiente:

$$\widehat{XYZ}^3 = \begin{pmatrix} x^L \\ x^F \end{pmatrix}^3 = \begin{pmatrix} (0, 0, 1, 0, 1, 1, 1, 1, 1, 1) \\ (1, 1, 0, 0, 0, 1, 1, 1, 1, 1) \end{pmatrix}^3$$

Entonces se calcula  $F(\widehat{XYZ}^3) = 8,678$  y se hace

$\underline{F} := \max\{\underline{F}, F(\widehat{XYZ}^3)\} = \max\{8,072, 8,678\} = 8,678$ . Luego de esto, vamos al paso 4, pues  $F(\widehat{XYZ}^3) \neq F(\widehat{XYZ}^2)$ .

Siguiendo este comportamiento del algoritmo se completa la primera rama del árbol presentada en la figura 4.1 donde se puede ver que el criterio de corte se cumple de la iteración 5 a la 6, pues  $F(\widehat{XYZ}^6) = F(\widehat{XYZ}^5)$  por lo que se corta la rama (línea punteada) y regresamos al nodo sin explorar más reciente que en este caso es 2. Entonces, ramificamos en su complemento para así iniciar nuevamente el proceso del algoritmo y explorar una nueva rama. En la figura 4.2 se puede observar el comportamiento del algoritmo si omitimos el criterio de corte presentado en el paso 3 como  $F(\widehat{XYZ}^k) = F(\widehat{XYZ}^{k-1})$  al tener una exploración completa de la rama, donde la leyenda INFACTIBILIDAD indica que no es posible seguir explorando ese camino ya que no hay nodos que explorar, es decir no hay instalaciones que cerrar.

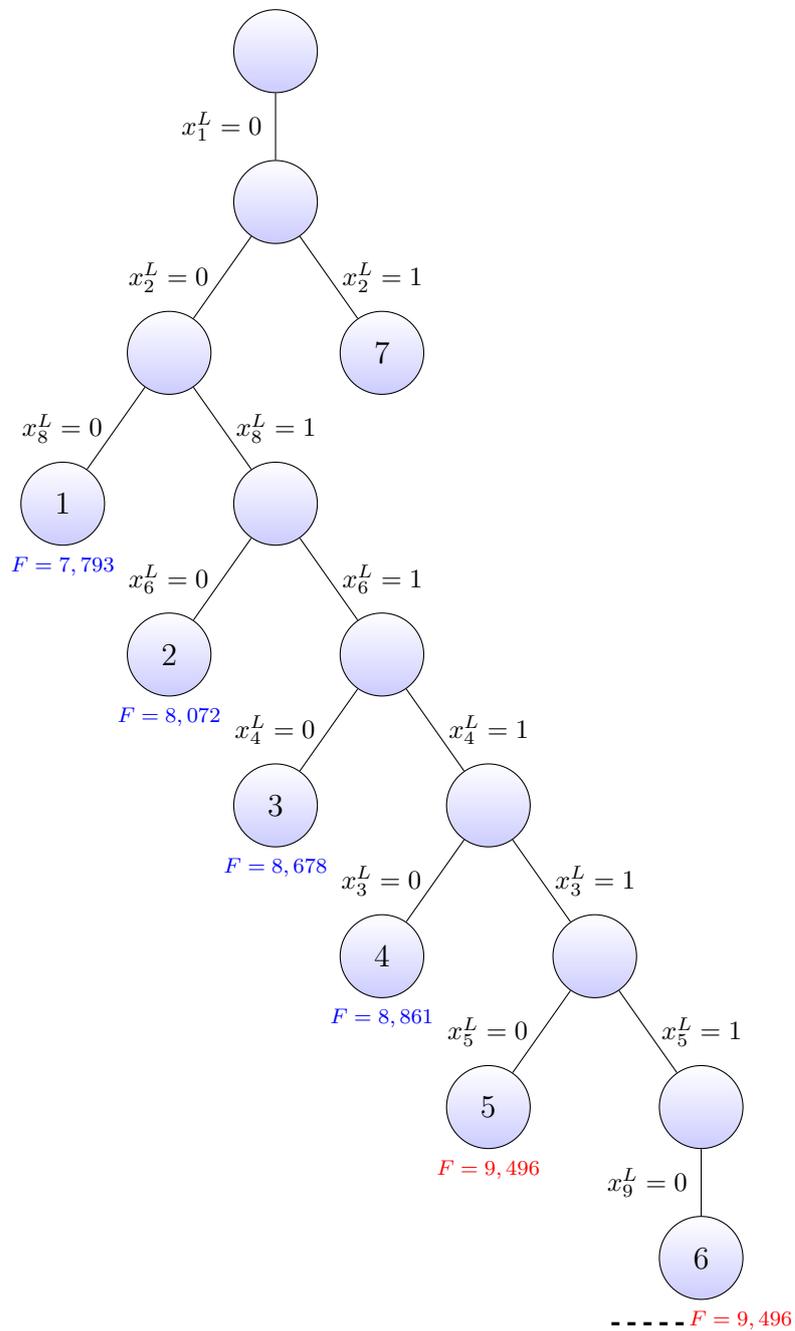


Figura 4.1: Rama con corte

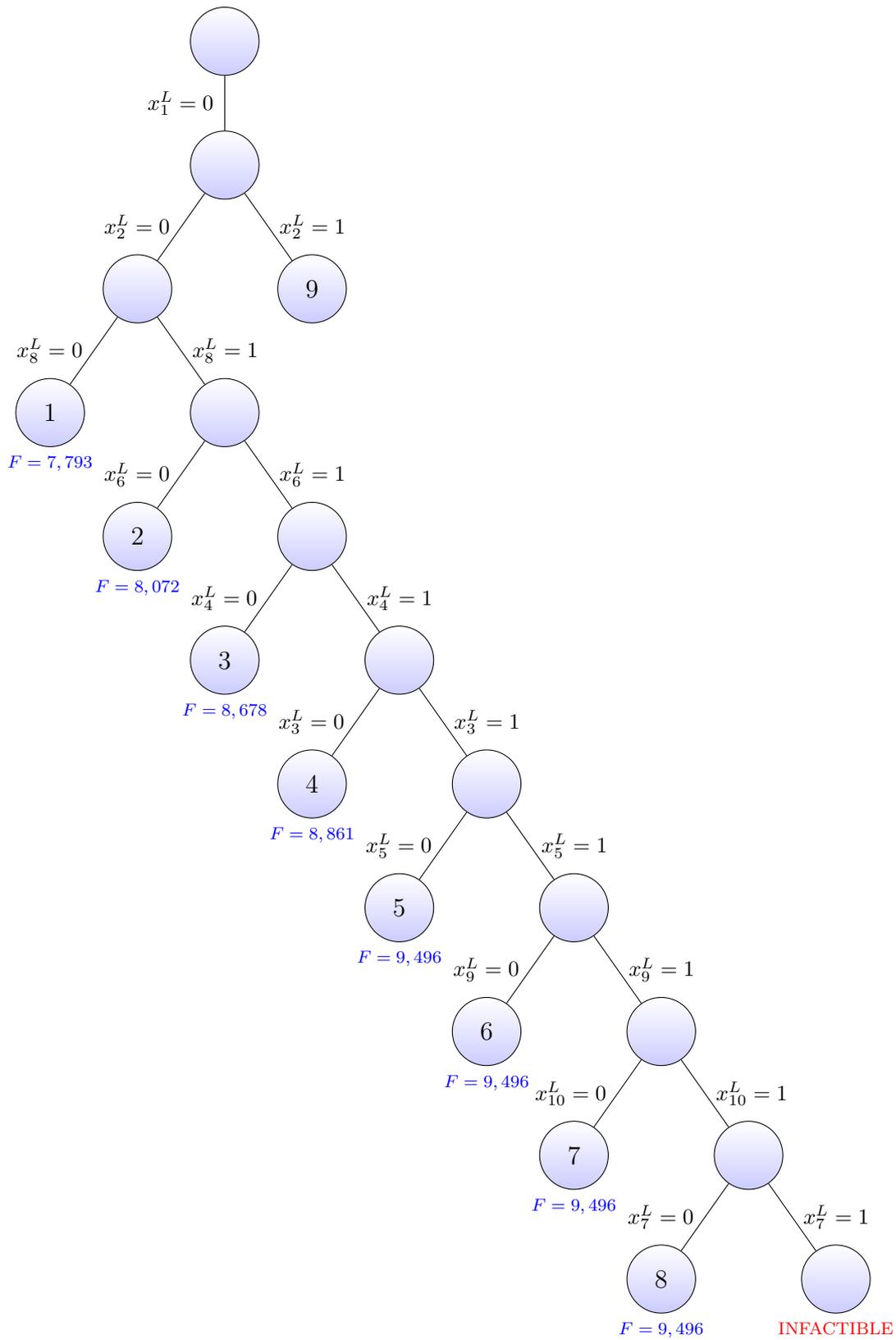


Figura 4.2: Rama completa

## EXPERIMENTACIÓN COMPUTACIONAL

---

En esta sección se muestran los resultados obtenidos al resolver cuatro conjuntos de instancias para el problema  $\overline{(r|p)}^{\mathcal{L}}$ -Centroide. Los resultados obtenidos se comparan con los obtenidos por un algoritmo exhaustivo. Con esto se muestra la efectividad del algoritmo propuesto en esta tesis. Este capítulo empieza con la descripción de las instancias y la forma en que fueron creadas. Después, se presentan los resultados obtenidos al considerar la lealtad por parte de los clientes.

### 5.1 GENERACIÓN DE INSTANCIAS DE PRUEBA

Como mencionamos en la introducción de esta tesis, el problema aquí planteado es una variante del problema  $(r|p)$ -centroide al considerar el cierre de instalaciones. Después de hacer una extensa revisión de literatura sobre trabajos que abordan el problema  $(r|p)$ -centroide, se concluyó que la mayoría de los trabajos usan las instancias de la librería de problemas de localización discretos, las cuales se pueden encontrar en la siguiente liga: [http://math.nsc.ru/AP/benchmarks/Competitive/p\\_med\\_comp\\_eng.html](http://math.nsc.ru/AP/benchmarks/Competitive/p_med_comp_eng.html). En la mayoría de estas instancias se tienen 100 nodos de demanda generados al azar con una distribución uniforme en el cuadrado  $7000 \times 7000$ , y las demandas pueden ser unitarias o aleatorias en un intervalo de  $[1,200]$ . Los valores utilizados de  $p$  y  $r$  son 5, 7, 10, 15, 20, 25 y 30. Estas instancias se usaron en Alekseeva *et al.* (2010); Alekseeva & Kochetov (2013); Alekseeva *et al.* (2015); Carrizosa *et al.* (2012); Davydov

*et al.* (2014); Biesinger *et al.* (2015). En Campos-Rodríguez *et al.* (2012) se generan nodos de demanda aleatorios en el cuadro unitario  $[0, 1] \times [0, 1]$  y se resuelven para valores de  $p$  y  $r$  de 2, 3, 5 y 8. Las demandas de los clientes las generan aleatoriamente en el intervalo  $[0, 1]$ . En Roboredo & Pessoa (2013) se utilizan dos grupos de instancias adicionales, unas aleatorias donde el número de instalaciones usado es de 20 y 50 para cada tomador de decisiones y el número de clientes es de 30 y 100, respectivamente. Los valores utilizados de  $p$  y  $r$  son 2 y 4, con demandas aleatorias en el intervalo  $[1, 20]$ .

Después de haber hecho la intensa revisión de literatura de trabajos que abordan el  $(r|p)$ -centroide identificamos los tamaños de instancias que se han utilizado, y se muestran en la tabla 5.1.

Instalaciones	Clientes	$r$ y $p$
100	100	10
20, 50	30, 100	2, 4
100	100	5, 10, 15, 20
50	50	10
100	100	7, 25, 30
25, 100	25, 100	2, 3, 5, 8

Tabla 5.1: Tamaños de las instancias en la literatura

Basados en esas instancias, las cuales son para un problema relacionado con el aquí estudiado, se decidió generar instancias de los tamaños mostrados en la tabla 5.2.

Instalaciones	Clientes	$r$ y $p$
10	40	2, 3, 5
9, 12, 15, 20, 25	55	2,3,4,5
20	80	3, 5, 8
10, 12, 14	100	2, 3, 4, 5
9,12,15,20,30,40, 50	150	2, 3, 4, 6

Tabla 5.2: Tamaños de las instancias generadas

Para obtener las instancias, se utilizaron dos generadores distintos y unas redes de nodos de la literatura. El primer generador crea instancias de manera aleatoria (conjunto A) y el otro crea las instancias de manera ordenada mediante un esquema que evita la aleatoriedad (conjunto B). Por último, se utilizó la bien conocida red de Swain Swain (1974) (conjunto C), y una versión extendida de dicha red (conjunto D), la cual fue utilizada en Ruiz-Hernández *et al.* (2017).

En ambos generadores, las coordenadas de las instalaciones y clientes están en el mallado  $[-1, 1] \times [-1, 1]$  y las demandas de cada cliente se obtienen aleatoriamente de los enteros en el intervalo  $(0, 200]$ . Como ya se mencionó, se utilizaron los datos de la red de Swain y la versión extendida de esta para introducirlos al segundo generador y obtener instancias para el problema aquí estudiado. A continuación vamos a describir a detalle la manera en la que funcionan ambos generadores.

### 5.1.1 GENERADOR ALEATORIO (CONJUNTO A)

En este generador se crean instancias de manera aleatoria. Es decir, se obtuvieron las coordenadas de las instalaciones del líder y seguidor de manera aleatoria dentro de un cuadrante predefinido. Análogamente, se generaron las coordenadas de los clientes de manera aleatoria. Es importante mencionar que no se permitía la repetición en dicha generación, es decir, un cliente y una instalación no pueden tener las mismas coordenadas. Dado que inicialmente los clientes están asignados a las instalaciones más cercanas, se validó que todas las instalaciones tuvieran al menos un cliente asignado.

Primeramente, se localizan  $2p$  instalaciones del líder en el mallado mencionado anteriormente, luego en el mismo mallado se localizan  $2q$  instalaciones del seguidor. Por último, se crean  $|J|$  nodos de demanda (clientes), que en un principio se asignan a la instalación más cercana. Luego de

esto, identificamos las instalaciones ya sea del líder o seguidor que no tienen cliente asignado para ser eliminadas, o bien hasta reducir la instancia a  $p$  y  $q$  instalaciones del líder y seguidor, respectivamente. Si el resultado no era efectivo, es decir, tener menos de  $p$  y  $q$  instalaciones con al menos un cliente asignado, se volvía a realizar este proceso hasta conseguir la instancia de tamaño deseado.

El código generador de este tipo de instancias se implementó en C++ utilizando Visual Studio 2017.

### 5.1.2 GENERADOR ORDENADO (CONJUNTO B)

Con el fin de tener instancias apegadas a la realidad en algún determinado mercado y además simular una situación real en un mercado competitivo, se decidió realizar este generador con el fin de evitar la aleatoriedad al localizar las instalaciones del líder y seguidor. En este generador, primero se creó aleatoriamente un conjunto de  $N$  nodos de demanda (clientes) sin repetición en el mallado mencionado anteriormente. Luego, para localizar las  $p$  instalaciones existentes del líder se resuelve un problema de la  $p$ -mediana, lo que permite minimizar la distancia media entre las instalaciones del líder y los nodos que serán los clientes. Después, se resuelve un problema  $(r|X_p)$ -medianoide para localizar las  $r$  instalaciones del seguidor considerando los  $|N| - p$  nodos restantes. De esta forma, el seguidor busca maximizar la demanda capturada de clientes tomando en cuenta las  $p$  instalaciones que ya tiene localizadas el líder. Note que en estas instancias se da lugar a tener clientes localizados en la misma posición de algunas instalaciones, por lo que tendríamos radios de lealtad iguales a cero, entonces para evitar esa situación se tomó la distancia mínima diferente de cero de ir de ese cliente-instalación  $j \in N$  a otro cliente  $i \in N$ , es decir,  $\min\{d_{ij}\} \neq 0$ , sin embargo, los clientes localizados en la misma posición de una instalación se asignan a ella.

El código generador de este tipo de instancias se implementó en C++ utilizando Visual Studio 2017 y CPLEX 12.7.1.

A continuación se presentan brevemente los modelos que se resolvieron con el fin de tener instancias más estructuradas y apegadas a la realidad en un mercado competitivo.

### MODELO DE LA $p$ -MEDIANA

El problema de la  $p$ -mediana se abordó por primera vez en ReVelle & Swain (1970), el cual consiste en localizar  $p$  instalaciones seleccionadas de un conjunto  $N$  de nodos de demanda de tal manera que se minimice la distancia media entre las instalaciones a localizar y los clientes asignados a ellas.

Para formular tal problema tome en cuenta lo siguiente:

#### **Conjuntos.**

$N$ : Nodos de demanda.

#### **Parámetros.**

$p$ : Número de instalaciones a localizar.

$h_i$ : Demanda del nodo  $i \in N$ .

$d_{ij}$ : Distancia de ir del nodo  $i \in N$  al nodo  $j \in N$ .

Variables de decisión:

$$x_j = \begin{cases} 1, & \text{Si se abre la instalación } j \in N \\ 0, & \text{En otro caso} \end{cases}$$

$$y_{ij} = \begin{cases} 1, & \text{Si la demanda del nodo } i \in N \text{ es servida por el nodo } j \in N \\ 0, & \text{En otro caso} \end{cases}$$

Por lo que el modelo de programación para el problema de localización  $p$ –mediana queda como sigue:

$$\text{mín} \sum_{i \in N} \sum_{j \in N} h_i d_{ij} y_{ij} \quad (5.1)$$

$$\text{Sujeto a:} \quad \sum_{j \in J} y_{ij} = 1 \quad \forall i \in N \quad (5.2)$$

$$\sum_{j \in J} x_j = p \quad (5.3)$$

$$y_{ij} \leq x_j \quad \forall i \in N, j \in N \quad (5.4)$$

$$x_j, y_{ij} \in \{0, 1\} \quad \forall i \in N, \forall j \in N \quad (5.5)$$

#### MODELO $(r|X_p)$ –MEDIANOIDE

El modelo  $(r|X_p)$ –medianoide fue presentado por primera vez en Hakimi (1983) el cual consiste en localizar  $r$  instalaciones de tal manera que capture la mayor demanda posible en un determinado mercado, considerando que una empresa ya se encuentra operando en dicho mercado al tener localizadas  $|X_p|$  instalaciones.

Para formular tal problema tome en cuenta los siguientes conjuntos y parámetros:

$I$ : Conjunto de los nodos de demanda.

$J^L$ : Conjunto de las instalaciones que ya abrió el líder mediante el modelo de la  $p$ -mediana.

$J^S$ : Conjunto de las instalaciones del seguidor.

$\bar{d}_i^L$ : Distancia a la instalación más cercana del líder, es decir,  $\bar{d}_i^L = \min_{j \in J^L} \{d_{ij}\}$

$N_i^S$ : Conjunto de índices de los sitios potenciales más cercanos que una instalación del líder, donde  $N_i^S = \{j : d_{ij} < \bar{d}_i^L\}, \forall i \in I$

$a_i$ : Demanda del nodo  $i \in I$

Variables de decisión:

$$x_j^S = \begin{cases} 1, & \text{Si el seguidor abre la instalación } j \in I \\ 0, & \text{En otro caso} \end{cases}$$

$$y_i^S = \begin{cases} 1, & \text{Si el seguidor captura la demanda del nodo } i \in I \\ 0, & \text{En otro caso} \end{cases}$$

Por lo que el modelo de programación para el problema de localización  $(r|X_p)$ -medianoide queda como sigue:

$$\text{máx} \sum_{i \in I} a_i y_i^S \quad (5.6)$$

$$\text{Sujeto a: } y_i^S \leq \sum_{j \in N_i^S} x_j \quad \forall i \in I \quad (5.7)$$

$$\sum_{j=1}^N x_j^S = r \quad \forall i \in I \quad (5.8)$$

$$y_i^S \in \{0, 1\} \quad \forall i \in I \quad (5.9)$$

$$x_j^S \in \{0, 1\} \quad \forall j \in J \quad (5.10)$$

### 5.1.3 GENERADOR BASADO EN LA RED DE SWAIN (CONJUNTO C) Y EN LA RED DE SWAIN EXTENDIDA (CONJUNTO D)

Como se mencionó anteriormente, se utilizó una red creada de manera manual por Ralph W. Swain, utilizada en Swain (1974), la cual consiste de 55 nodos de demanda bien distribuidos. La red se muestra en la figura 5.1. Además, también se usó una versión extendida de esta red creada en Ruiz-Hernández *et al.* (2017) para tener 150 nodos de demanda en total. La red extendida se muestra en la figura 5.2. Cabe recalcar que estas redes se dieron como dato de entrada al generador ordenado recién descrito para tener instancias para el problema aquí estudiado.

Note que en estas instancias se da lugar a tener clientes localizados en la misma posición de algunas instalaciones, por lo que tendríamos radios de lealtad iguales a cero, entonces para evitar esa situación se tomó la distancia mínima diferente de cero de ir de un cliente  $j \in J$  a una instalación  $i \in J$ , es decir,  $\min\{d_{ij}\} \neq 0$ , sin embargo, los clientes localizados en la misma posición de una instalación se asignan a ella. Hay que tener claro que al tratarse de una red de nodos, las distancias que se toman son las distancias en red, es decir, la distancia del camino mínimo de ir de un nodo a otro obtenidas mediante el algoritmo de Dijkstra y no la distancia euclidiana.

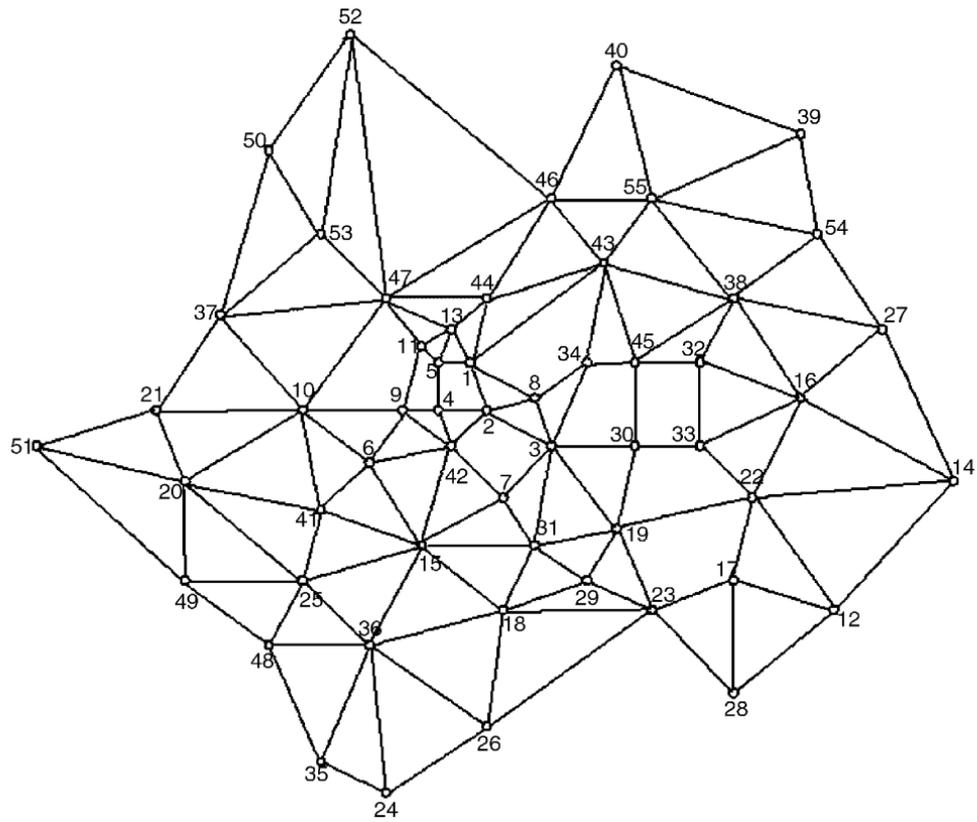


Figura 5.1: Red de Swain.

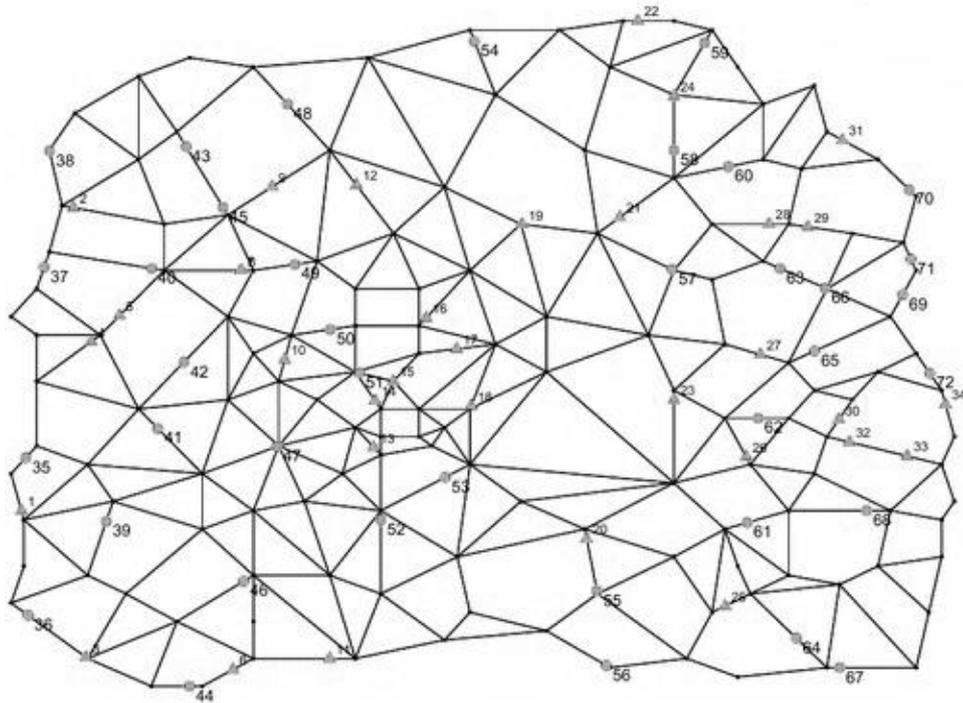


Figura 5.2: Red de Swain Extendida.

## 5.2 AMBIENTE COMPUTACIONAL

El algoritmo propuesto en la sección 4 se implementó en FICO Xpress optimization suite 8.4. Para llevar a cabo la experimentación computacional se utilizó una estación de trabajo con un procesador Intel (R) Core de 3.0 GHz con 32 GB de memoria RAM y Windows 8 Professional. Para validar la efectividad del algoritmo propuesto, se comparan los resultados con los resultados obtenidos por un algoritmo de enumeración explícita. En este último, se verifican todas las posibles soluciones del líder; lo cual es posible solo para instancias de tamaño limitado. En ambos algoritmos, el problema del nivel inferior se resuelve a optimalidad para cada solución del líder.

### 5.3 RESULTADOS NUMÉRICOS

En las siguientes tablas presentamos el nombre de la instancia de la siguiente manera: número de instalaciones del líder y seguidor, número de clientes, instalaciones a cerrar por el líder y seguidor, y el número de la instancia. Sin perder generalidad, se consideró que  $|I^L| = |I^F|$  y  $p = r$ . También se muestra el valor de la función objetivo del líder  $F(x, y, z)$ , el cual representa la demanda capturada una vez que cerró  $p$  instalaciones. Además, para cada esquema de solución, se presenta el número de soluciones exploradas y el tiempo de ejecución requerido (en segundos) y se muestra el ahorro de tiempo correspondiente. Por último, se muestra el porcentaje de ahorro relativo en soluciones exploradas. Dicho ahorro relativo se calcula como sigue:

$$\% \text{ Ahorro relativo} = \frac{100 * (\text{Sol. posibles} - \text{Sol. exploradas})}{\text{Sol. posibles}}$$

Con el fin de facilitar la lectura de aquí en adelante nos referiremos al ahorro relativo en soluciones exploradas simplemente como *ahorro*.

En las tablas 5.3 y 5.4 se muestran los resultados obtenidos de resolver las instancias del conjunto A. En la tabla 5.5, 5.6 y 5.7 se muestran los resultados de las instancias del conjunto B. En las tablas 5.8 y 5.9 se muestran los resultados obtenidos de las instancias de los conjuntos C y D, respectivamente.

Instancia	Valor Óptimo	B&B propuesto		Enumerativo explícito		Ahorro tiempo (seg)	% Ahorro relativo
		Soluciones exploradas	Tiempo (seg)	Soluciones exploradas	Tiempo (seg)		
10.40.2-1	1510	24	3.116	45	4.759	1.643	46.67
10.40.2-2	1863	40	3.993	45	4.257	0.264	11.11
10.40.2-3	1078	23	2.939	45	4.43	1.491	48.89
10.40.2-4	1629	19	2.518	45	4.301	1.783	57.78
10.40.2-5	1969	23	2.838	45	4.467	1.629	48.89
10.40.3-1	1510	21	2.765	120	14.42	11.655	82.50
10.40.3-2	1962	106	13.261	120	13.746	0.485	11.67
10.40.3-3	1078	81	12.033	120	14.313	2.28	32.50
10.40.3-4	1629	67	9.606	120	13.37	3.764	44.17
10.40.3-5	1927	85	12.841	120	15.398	2.557	29.17
10.40.5-1	1520	237	33.439	252	43.591	10.152	5.95
10.40.5-2	2122	239	30.321	252	30.405	0.084	5.16
10.40.5-3	1316	237	50.623	252	50.92	0.297	5.95
10.40.5-4	1672	239	43.915	252	44.055	0.14	5.16
10.40.5-5	1843	234	45.003	252	45.032	0.029	7.14

Tabla 5.3: Resultados instancias de tamaño  $I^L = I^F = 10$ ,  $J = 40$  y  $p = r \in \{2, 3, 5\}$ 

Instancia	Valor Óptimo	B&B propuesto		Enumerativo explícito		Ahorro tiempo (seg)	% Ahorro relativo
		Soluciones exploradas	Tiempo (seg)	Soluciones exploradas	Tiempo (seg)		
20.80.3-1	4282	401	199.159	1140	302.996	103.837	64.82
20.80.3-2	3360	444	206.882	1140	407.298	200.416	61.05
20.80.3-3	3985	208	106.569	1140	309.517	202.948	81.75
20.80.3-4	3921	393	198.23	1140	291.966	93.736	65.53
20.80.3-5	4087	375	178.548	1140	318.092	139.544	67.11
20.80.5-1	4282	8759	3154.13	15504	4884.19	1730.06	43.50
20.80.5-2	3360	5400	3149.17	15504	4455.4	1306.23	65.17
20.80.5-3	3985	5816	2675.75	15504	5172.4	2496.65	62.49
20.80.5-4	3921	7760	3555.47	15504	4865.17	1309.7	49.95
20.80.5-5	4087	8507	4608.04	15504	4652.96	44.92	45.13
20.80.8-1	4222	103085	50366.5	125970	51489.3	1122.8	18.17

Tabla 5.4: Resultados instancias de tamaño  $I^L = I^F = 20$ ,  $J = 80$  y  $p = r \in \{3, 5, 8\}$ 

Instancia	Valor Óptimo	B&B propuesto		Enumerativo explícito		Ahorro tiempo (seg)	% Ahorro relativo
		Soluciones exploradas	Tiempo (seg)	Soluciones exploradas	Tiempo (seg)		
10.100.2-1	4595	42	11.391	45	11.459	0.068	6.67
10.100.2-2	4533	37	10.435	45	11.88	1.445	17.78
10.100.2-3	4516	38	10.623	45	11.608	0.985	15.56
10.100.2-4	5320	34	8.832	45	9.312	0.48	24.44
10.100.2-5	3959	38	10.333	45	11.439	1.106	15.56
10.100.3-1	4533	116	28.321	120	39.168	10.847	3.33
10.100.3-2	4412	117	26.651	120	37.992	11.341	2.50
10.100.3-3	4426	114	27.519	120	38.788	11.269	5
10.100.3-4	5375	89	30.674	120	35.02	4.346	25.83
10.100.3-5	3884	107	36.693	120	38.68	1.987	10.83
10.100.4-1	4419	202	62.733	210	87.304	24.571	3.81
10.100.4-2	4347	207	62.631	210	88.345	25.714	1.43
10.100.4-3	4200	208	64.449	210	88.854	24.405	0.95
10.100.4-4	5167	190	60.523	210	85.799	25.276	9.52
10.100.4-5	3796	183	58.118	210	87.234	29.116	12.86
10.100.5-1	4231	246	117.142	252	117.723	0.581	2.38
10.100.5-2	4423	248	113.221	252	113.534	0.313	1.59
10.100.5-3	4011	250	125.892	252	126.134	0.242	0.79
10.100.5-4	4880	227	118.389	252	121.4	3.011	9.92
10.100.5-5	3719	221	111.113	252	117.696	6.583	12.30

Tabla 5.5: Resultados instancias de tamaño  $I^L = I^F = 10$ ,  $J = 100$  y  $p = r \in \{2, 3, 4, 5\}$

Instancia	Valor Óptimo	B&B propuesto		Enumerativo explícito		Ahorro tiempo (seg)	% Ahorro relativo
		Soluciones exploradas	Tiempo (seg)	Soluciones exploradas	Tiempo (seg)		
12.100.2-1	4561	58	12.439	66	13.095	0.656	12.12
12.100.2-2	4448	66	13.906	66	20.328	6.422	0
12.100.2-3	4271	51	10.939	66	12.783	1.844	22.73
12.100.2-4	5030	58	12.985	66	13.782	0.797	12.12
12.100.2-5	4058	65	13.189	66	13.25	0.061	1.52
12.100.3-1	4553	210	54.387	220	54.425	0.038	4.55
12.100.3-2	4425	219	84.367	220	84.647	0.28	0.45
12.100.3-3	4408	214	56.703	220	57.514	0.811	2.73
12.100.3-4	5192	188	52.804	220	55.801	2.997	14.55
12.100.3-5	3926	182	50.17	220	54.685	4.515	17.27
12.100.4-1	4510	476	157.159	495	161.002	3.843	3.84
12.100.4-2	4418	495	158.098	495	158.102	0.004	0
12.100.4-3	4329	479	143.166	495	146.966	3.8	3.23
12.100.4-4	5299	459	137.378	495	143.146	5.768	7.27
12.100.4-5	3819	431	139.169	495	145.405	6.236	12.93
12.100.5-1	4358	792	293.204	792	293.832	0.628	0
12.100.5-2	4378	792	379.97	792	381.67	1.7	0
12.100.5-3	4161	790	272.884	792	275.545	2.661	0.25
12.100.5-4	5334	753	264.627	792	273.425	8.798	4.92
12.100.5-5	3709	701	260.109	792	277.501	17.392	11.49
12.100.6-1	4257	918	558.453	924	566.364	7.911	0.65
12.100.6-2	4409	924	512.957	924	518.565	5.608	0
12.100.6-3	4203	915	511.991	924	526.935	14.944	0.97
12.100.6-4	5348	901	487.329	924	508.851	21.522	2.49
12.100.6-5	3664	894	471.327	924	483.636	12.309	3.25

Tabla 5.6: Resultados instancias de tamaño  $I^L = I^F = 12$ ,  $J = 100$  y  $p = r \in \{2, 3, 4, 5, 6\}$ 

Instancia	Valor Óptimo	B&B propuesto		Enumerativo explícito		Ahorro tiempo (seg)	% Ahorro relativo
		Soluciones exploradas	Tiempo (seg)	Soluciones exploradas	Tiempo (seg)		
14.100.2-1	4801	69	25.176	91	30.368	5.192	24.18
14.100.2-2	4830	71	27.34	91	32.003	4.663	21.98
14.100.2-3	4070	88	32.049	91	33.491	1.442	3.30
14.100.2-4	5002	71	25.905	91	31.563	5.658	21.98
14.100.2-5	4267	68	23.714	91	32.068	8.354	25.27
14.100.3-1	4700	330	135.765	364	149.296	13.531	9.34
14.100.3-2	5000	316	140.126	364	155.087	14.961	13.19
14.100.3-3	4043	347	150.381	364	160.772	10.391	4.67
14.100.3-4	4972	304	132.554	364	152.256	19.702	16.48
14.100.3-5	4162	339	141.987	364	150.967	8.98	6.87
14.100.4-1	4597	943	442.166	1001	466.315	24.149	5.79
14.100.4-2	5036	949	403.764	1001	476.62	72.856	5.19
14.100.4-3	4059	940	450.486	1001	466.153	15.667	6.09
14.100.4-4	4876	924	424.246	1001	466.473	42.227	7.69
14.100.4-5	4168	991	441.315	1001	463.441	22.126	1
14.100.5-1	4491	1957	716.025	2002	1036.73	320.705	2.25
14.100.5-2	5103	1978	722.862	2002	741.79	18.928	1.20
14.100.5-3	4034	1946	691.379	2002	1011.7	320.321	2.80
14.100.5-4	4871	1902	678.694	2002	998.977	320.283	5
14.100.5-5	4208	2002	677.3	2002	697.338	20.038	0
14.100.6-1	4418	2939	1674.4	3003	1697.11	22.71	2.13
14.100.6-2	5159	2964	1742.09	3003	1747.57	5.48	1.30
14.100.6-3	3993	2970	1191.24	3003	1693.95	502.71	1.10
14.100.6-4	4892	2929	1308.56	3003	1656.41	347.85	2.46
14.100.6-5	4108	3001	1154.5	3003	1672.95	518.45	0.07

Tabla 5.7: Resultados instancias de tamaño  $I^L = I^F = 14$ ,  $J = 100$  y  $p = r \in \{2, 3, 4, 5, 6\}$

Instancia	Valor Óptimo	B&B propuesto		Enumerativo explícito		Ahorro tiempo (seg)	% Ahorro relativo
		Soluciones exploradas	Tiempo (seg)	Soluciones exploradas	Tiempo (seg)		
9.55.2-1	8355	35	5.157	36	5.268	0.111	2.78
9.55.3-2	8650	82	15.461	84	15.699	0.238	2.38
9.55.4-3	8805	122	28.834	126	29.472	0.638	3.17
12.55.2-1	9568	50	10.069	66	11.508	1.439	24.24
12.55.3-2	9568	166	39.776	220	45.321	5.545	24.55
12.55.4-3	9330	449	132.534	495	136.288	3.754	9.29
12.55.5-4	9139	736	235.184	792	239.425	4.241	7.07
15.55.2-1	9024	40	9.938	105	19.019	9.081	61.90
15.55.3-2	9024	156	53.372	455	102.834	49.462	65.71
15.55.4-3	9024	442	190.394	1365	380.181	189.787	67.62
15.55.5-4	9024	1211	543.454	3003	935.113	391.659	59.67
20.55.2-1	9766	25	8.724	190	42.954	34.23	86.84
20.55.3-2	9766	157	70.788	1140	290.458	219.67	86.23
20.55.4-3	9766	864	440.251	4845	1380.68	940.429	82.17
20.55.5-4	9766	2972	1626.39	15504	3447.79	1821.4	80.83
25.55.2-1	10166	17	7.723	300	80.595	72.872	94.33
25.55.3-2	10166	93	46.718	2300	633.586	586.868	95.96
25.55.4-3	10166	671	523.065	12650	2986.23	2463.165	94.70
25.55.5-4	10166	2446	5746.05	53130	18906.7	13160.65	95.40

Tabla 5.8: Resultados instancias Red de Swain,  $\delta^L = \delta^S = 2$ 

Instancia	Valor Óptimo	B&B propuesto		Enumerativo explícito		Ahorro tiempo (seg)	% Ahorro relativo
		Soluciones exploradas	Tiempo (seg)	Soluciones exploradas	Tiempo (seg)		
9.150.2-1	62231	36	13.566	36	13.912	0.346	0
9.150.4-2	62455	126	73.527	126	74.539	1.012	0
12.150.2-1	61098	57	26.496	66	30.295	3.799	13.64
12.150.4-2	60888	495	323.009	495	324.286	1.277	0
12.150.6-3	59486	57	26.819	924	29.189	2.37	93.83
15.150.2-1	62566	105	56.35	105	56.431	0.081	0
15.150.4-2	63867	1365	1003.03	1365	1003.11	0.08	0
15.150.6-3	62398	5004	3579.39	5005	4104.64	525.25	0.02
20.150.2-1	61500	190	89.422	190	90.22	0.798	0
20.150.4-2	60581	4843	2819.8	4845	2944.86	125.06	0.04
20.150.6-3	59379	38733	25078.5	38760	25612	533.5	0.07
30.150.2-1	64413	296	344.691	435	482.319	137.628	31.95
30.150.4-2	65473	23130	18571.2	27405	20947.6	2376.4	15.60
40.150.2-1	66717	259	518.091	780	1251.14	733.049	66.79
40.150.3-2	66717	2948	5652.2	9880	12420.4	6768.2	70.16
50.150.2-1	71466	334	593.627	1225	2484.57	1890.943	72.73
50.150.3-2	71466	4138	8167.51	19600	31811	23643.49	78.89

Tabla 5.9: Resultados instancias Red extendida de Swain (dándola como dato de entrada el segundo generador)

Para el conjunto A de instancias podemos observar de la tabla 5.3 que en las instancias 10.40.2 – 2, 10.40.3 – 2, 10.40.5 – 1, 10.40.5 – 2, 10.40.5 – 3, 10.40.5 – 4 y 10.40.5 – 5, el ahorro está por debajo del 15 % del total de nodos. Además, en 10.40.2 – 1, 10.40.2 – 3, 10.40.2 – 5, 10.40.3 – 3, 10.40.3 – 4, 10.40.3 – 5 se obtuvo un ahorro de casi el 50 %. Más aún, en 10.40.2 – 4 y 10.40.3 – 1 se logró un ahorro mayor a la mitad del total de nodos, es decir, en esas dos instancias se exploraron menos de la mitad de los nodos para resolver el problema. En la tabla 5.4, los resultados son mejores ya que en 10 de las 11 instancias se tiene un ahorro por encima del 40 %, siendo 20.80.3 – 3 la que presenta el mejor ahorro con 81.75 %, es decir se ahorró más de la tercera parte del total de nodos para obtener el valor óptimo. Note que la instancia 20.80.8 – 1 es la que presenta el ahorro más bajo con 18.17 %.

Para las instancias del conjunto B, donde se tienen 10 instalaciones potenciales y 100 clientes, de la tabla 5.5 podemos ver que ninguna instancia pudo superar el 30 % de ahorro. Las instancias que presentaron más ahorro son 10.100.2 – 4 y 10.100.3 – 4 con 24.44 % y 25.83 %, respectivamente. Las instancias 1, 2, 3 y 5 con  $p = r = 2$ , 10.100.3 – 5, las instancias 4 y 5 con  $p = r \in \{4, 5\}$ , presentan un ahorro entre el 5 % y 20 %, respectivamente. Únicamente 9 de estas instancias presentan muy poco ahorro, pues las instancias 1, 2 y 3 con  $p = r \in \{3, 4, 5\}$  no superan el 5 %.

En la tabla 5.6 donde se tienen 12 instalaciones potenciales y 100 clientes, la instancia que presenta el mayor ahorro es 12.100.2 – 3 con el 22.73 %. Además, podemos ver que las instancias 1 y 4 con  $p = r = 2$ , las instancias 4 y 5 para  $p = r \in \{3, 4\}$  y 12.100.5 – 5, respectivamente, presentan un ahorro entre el 5 % y 20 % del total de nodos. Sin embargo, 16 de las 25 instancias presentan poco ahorro al estar muy por debajo del 5 %. Lamentablemente podemos observar que en cinco de estas instancias no se presenta ahorro alguno.

En la tabla 5.7 podemos ver que las instancias 14.100.2–1, 14.100.2–2, 14.100.2–4, 14.100.2–5 y 14.100.3–4 presentan el mejor ahorro en nodos oscilando entre el 16.48 % y 25.27 %, pero 20 instancias del total presentan un ahorro por debajo del 15 %. Solamente la instancia 14.100.5 – 5 no presentó ahorro.

Para las instancias del conjunto C podemos observar de la tabla 5.8 que los resultados son muy alentadores, pues en 12 de las 19 instancias presentan un ahorro por encima del 50 %, en las instancias 1, 2, 3 y 4 con 25 instalaciones potenciales y 55 clientes con  $p = r \in \{2, 3, 4, 5\}$  presentan el mejor ahorro pues están por encima del 90 %. Solamente las instancias 1, 2 y 3, donde se tienen 9 instalaciones potenciales y las instancias 1, 2, 3 y 4, donde se tienen 12 instalaciones potenciales con  $p = r \in \{2, 3, 4, 5\}$ , respectivamente, presentan un ahorro menor al 30 %.

Para las instancias del conjunto D podemos observar de la tabla 5.9 que solamente 11 de las 17 instancias presentan un ahorro en nodos, siendo las instancias 40.150.3 – 2, 50.150.2 – 1 y 50.150.3 – 2 las que presentan el mayor ahorro pues superan el 70 %. Las instancias que menos presentan ahorro son 15.150.6 – 3, 20.150.4 – 2 y 20.150.6 – 3 con 0.02 %, 0.04 % y 0.07 %, respectivamente. Sin embargo, seis del total de instancias no presentan ahorro en nodos.

De manera general podemos decir que los resultados más alentadores los obtuvimos con las instancias del tipo A, B y C. Sin embargo, hay que tomar en cuenta que en cinco instancias del tipo B, cuando se tenían 12 clientes no se presentó ahorro en nodos. Siendo las instancias tipo C las que presenta los ahorros más alentadores, pues en todas ellas se presentan ahorros considerables. De las instancias del conjunto D podemos apreciar que el ahorro comienza a percibirse conforme aumenta el tamaño de la instancia. Sin embargo, en muchas de esas instancias no se presenta ahorro alguno.

### 5.3.1 COMPARANDO EL TIEMPO COMPUTACIONAL REQUERIDO

Como era de esperarse, el algoritmo propuesto parece ser una buena opción para resolver el problema planteado en esta tesis. A continuación se muestran los gráficos correspondientes a las tablas presentadas anteriormente con respecto al tiempo computacional requerido en segundos, en el eje  $X$  se presenta el número de instancia y en el eje  $Y$  el tiempo en escala logarítmica base 10. La barra de la izquierda (para cada instancia) indica el tiempo que utilizó el algoritmo basado en ramificación y acotamiento propuesto en esta tesis y la barra de la derecha muestra el tiempo que utilizó el algoritmo de enumeración exhaustiva.

Podemos observar en todos los gráficos que el tiempo de ejecución del algoritmo propuesto en esta tesis siempre está por debajo del algoritmo de enumeración completa. Esto puede ser obvio de ver, pues claro está que nuestro algoritmo reduce el esfuerzo enumerativo al explorar menos nodos en la mayoría de los casos. Sin embargo, hay que hacer notar que incluso aunque no haya ahorro en nodos (como pudimos observar en algunas instancias tipo B y D), los tiempos de ejecución de nuestro algoritmo y los del algoritmo de enumeración exhaustiva, respectivamente, son comparables.

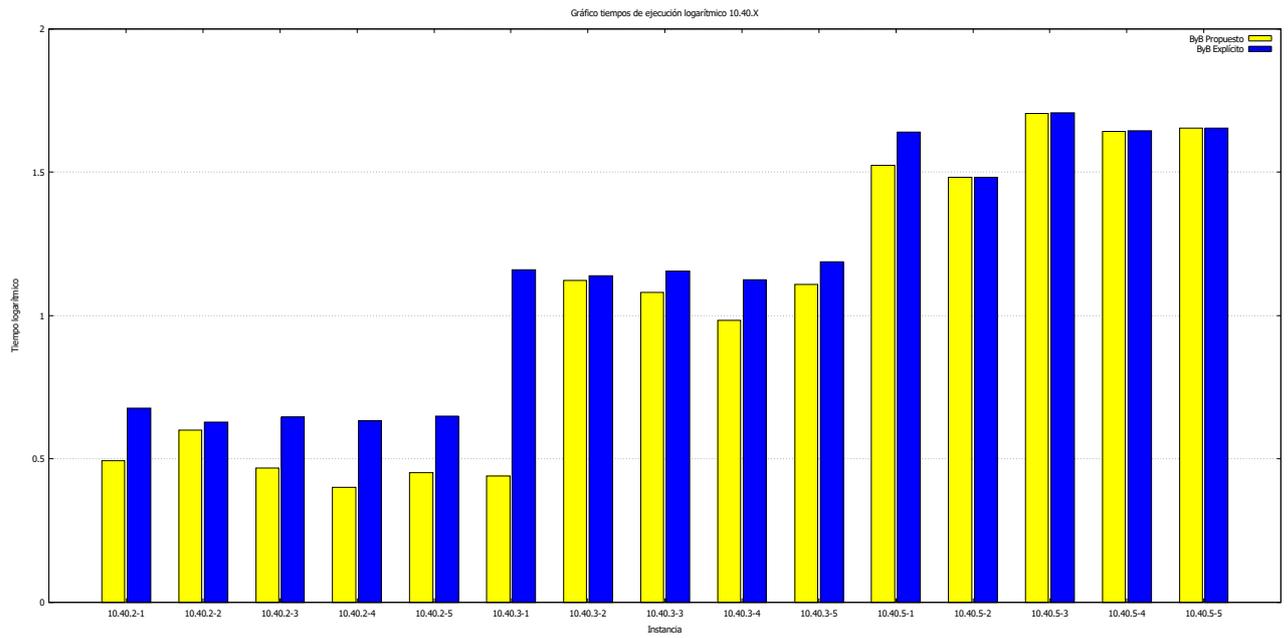


Figura 5.3: Comparación de tiempos de la tabla 5.3.

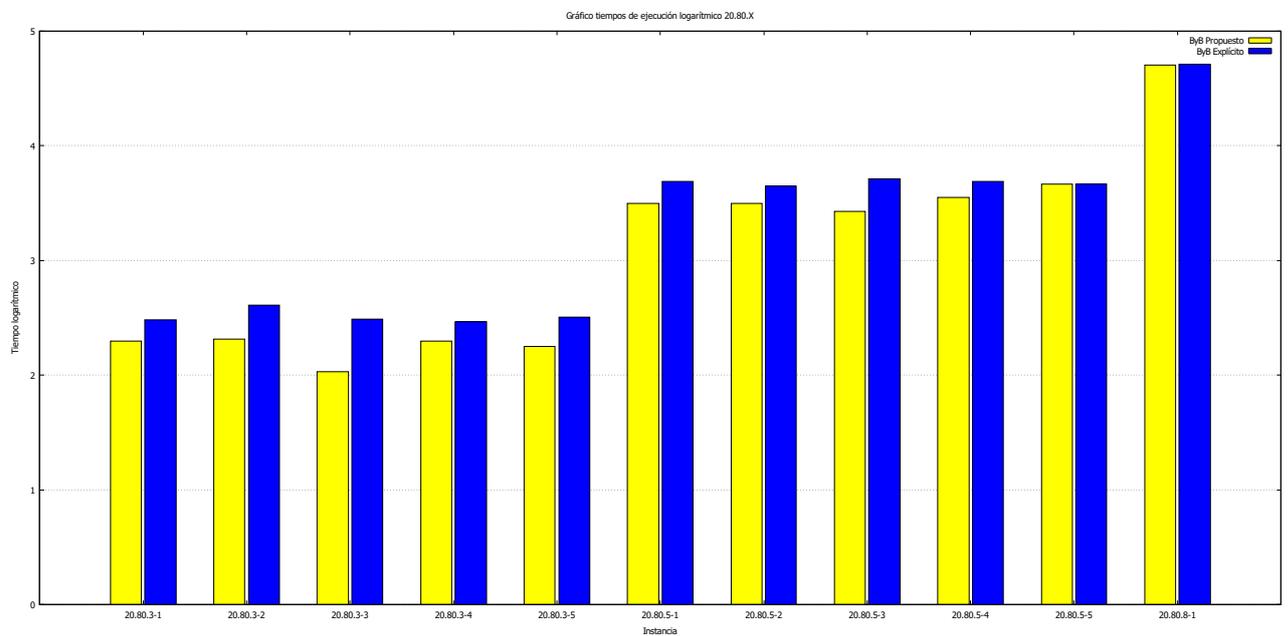


Figura 5.4: Comparación de tiempos de la tabla 5.4.

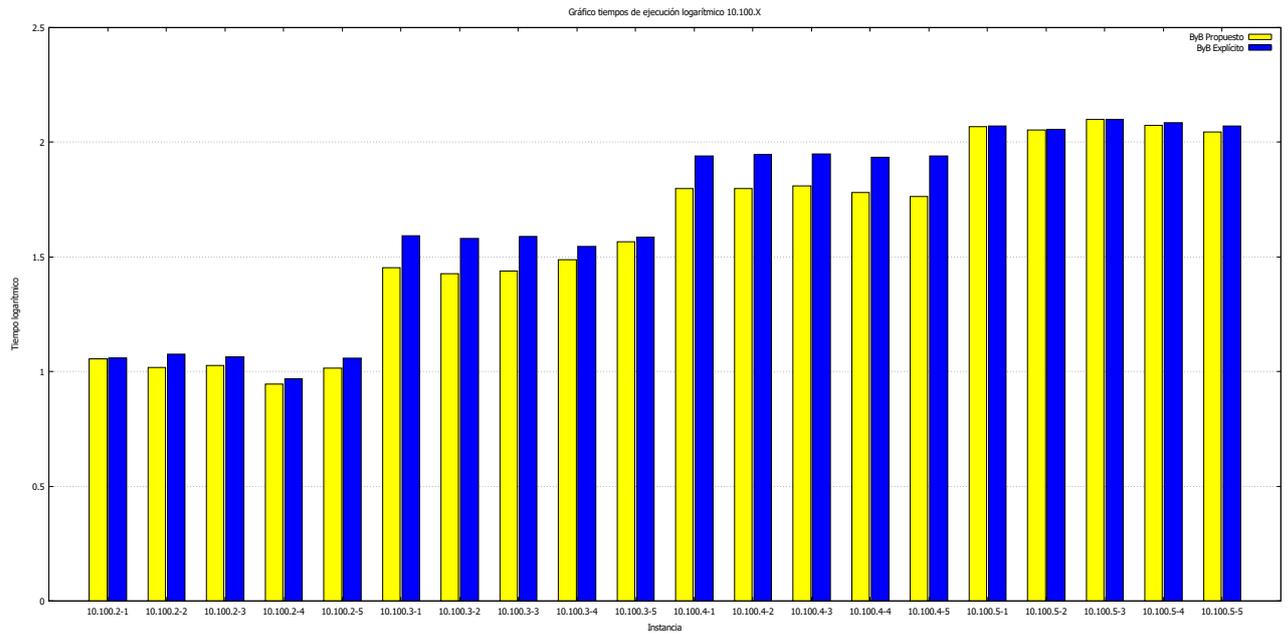


Figura 5.5: Comparación de tiempos de la tabla 5.5.

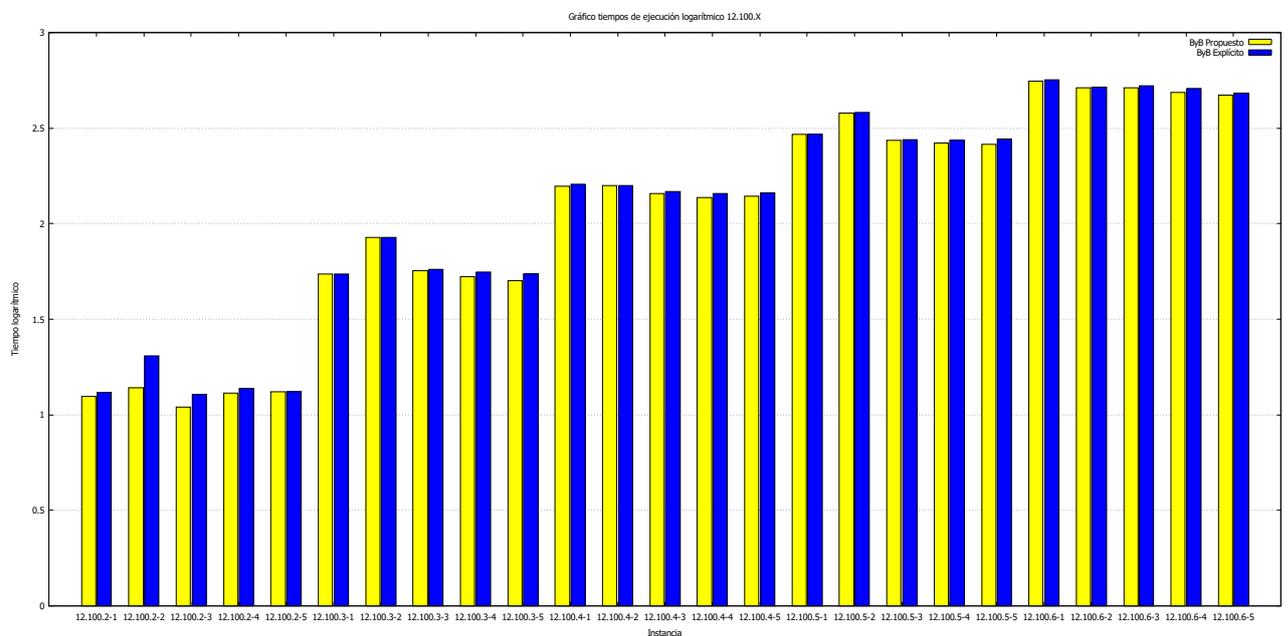


Figura 5.6: Comparación de tiempos de la tabla 5.6.

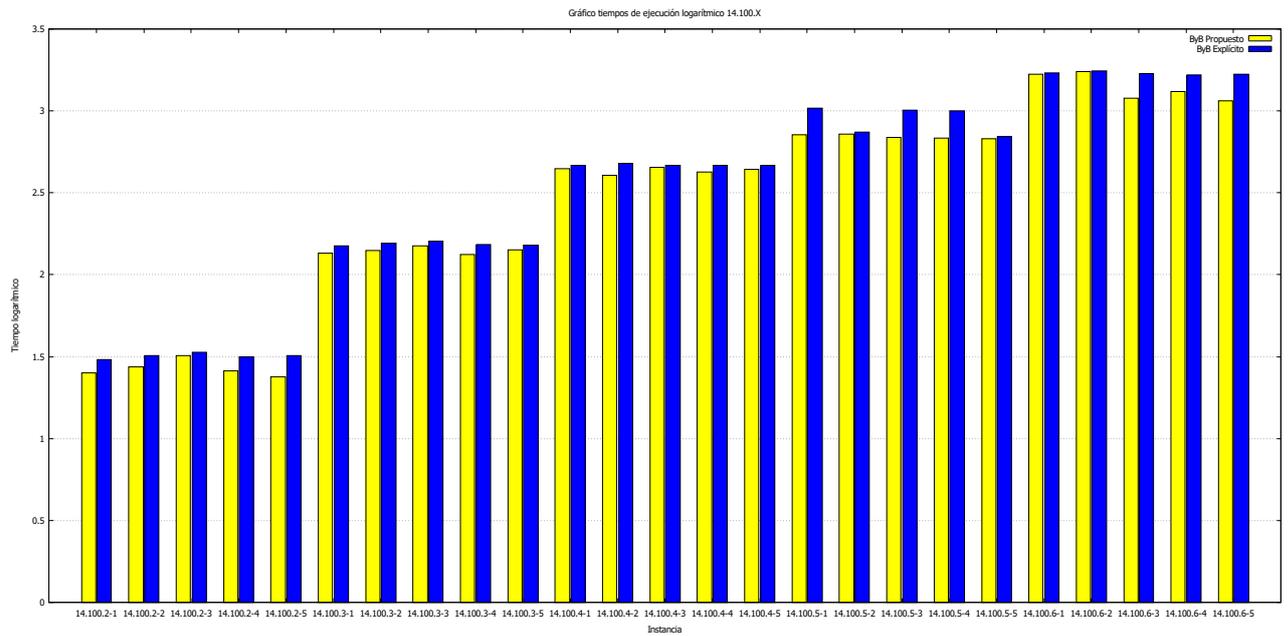


Figura 5.7: Comparación de tiempos de la tabla 5.7.

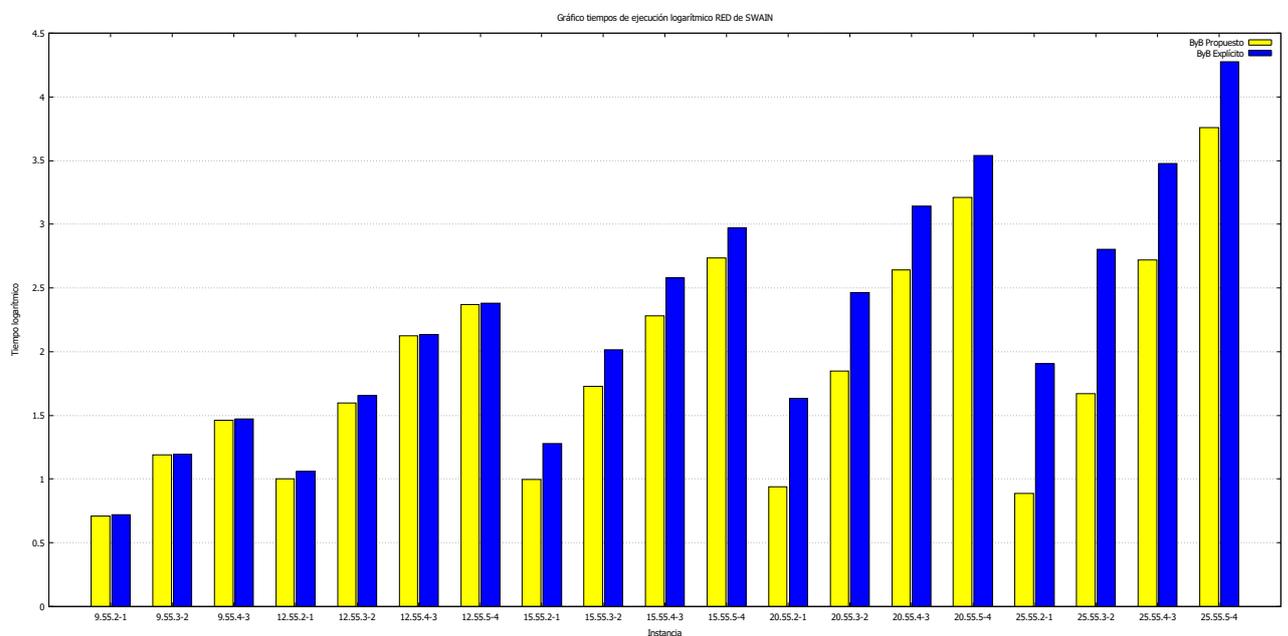


Figura 5.8: Comparación de tiempos de la tabla 5.8.

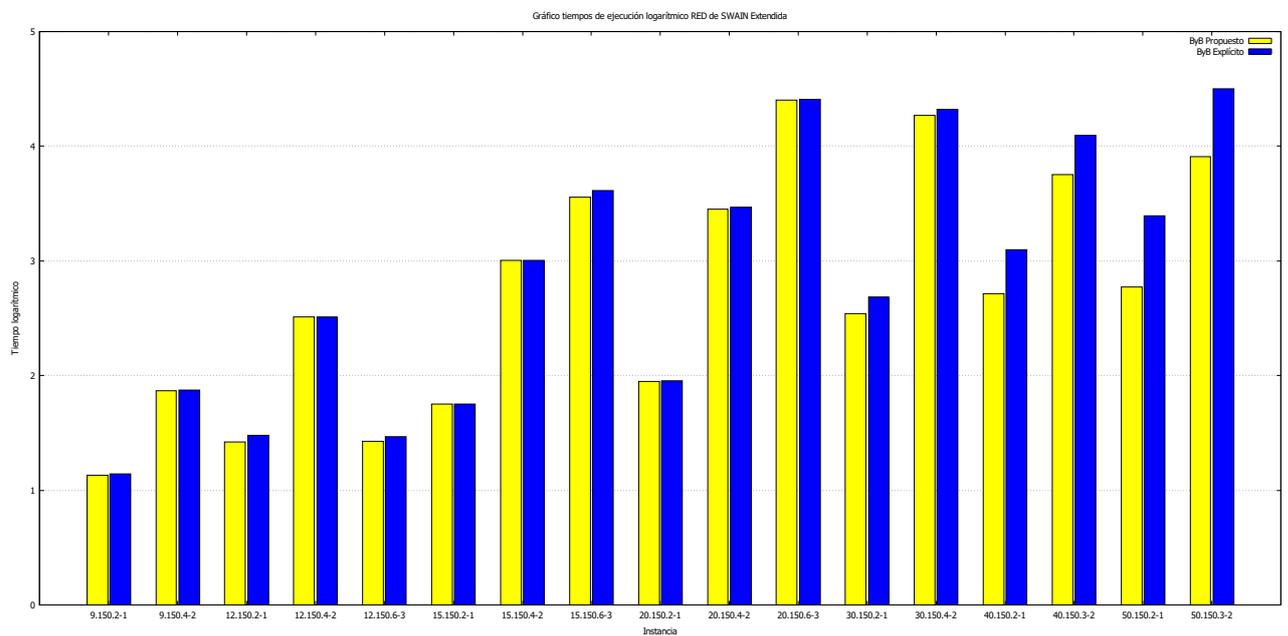


Figura 5.9: Comparación de tiempos de la tabla 5.9.

## CONCLUSIONES Y TRABAJO A FUTURO

---

En este capítulo se presentan las conclusiones a las que llegamos después de estudiar detalladamente el problema bi-nivel  $\overline{(r|p)}^{\mathcal{L}}$  –Centroide con lealtad de los clientes y de analizar los resultados arrojados en la experimentación computacional mediante el algoritmo propuesto. Además, se presentan algunas posibles líneas de investigación futura que se pueden desarrollar a partir del trabajo presentado en esta tesis.

### 6.1 CONCLUSIONES

En esta tesis consideramos un problema bi-nivel de cierre de instalaciones competitivas considerando lealtad de los clientes, nosotros denominamos a este problema el  $\overline{(r|p)}^{\mathcal{L}}$  –Centroide. En este problema de programación bi-nivel se consideró que tanto el líder como el seguidor buscan maximizar la demanda capturada. Es importante mencionar que en nivel inferior se agregaron restricciones válidas para satisfacer las restricciones de lealtad de los clientes. En el problema se tienen que cerrar un número predefinido de instalaciones del líder y del seguidor de manera jerárquica, de tal manera que los tomadores de decisiones maximicen la demanda capturada después de haber cerrado sus instalaciones.

Para resolver el problema propusimos un algoritmo basado en ramificación y acotamiento. Este algoritmo se adaptó de uno existente en la literatura para resolver problemas generales de programación bi-nivel binaria.

En base a los resultados obtenidos, se puede asegurar que el algoritmo propuesto es eficiente, ya que en la mayoría de las instancias utilizadas se presenta ahorro en nodos. Sin embargo, es evidente que dicho ahorro es muy variante. Por ejemplo, para el primer conjunto de instancias (A) es más notable el ahorro que para las instancias del segundo conjunto (B). Suponemos que esto se debe a que las instancias del conjunto B están conformadas por redes de nodos más estructuradas y apegadas a la realidad que al tomarlas de forma aleatoria. También se puede observar que en algunos casos el ahorro es poco o nulo, como trabajo a futuro pensamos en desarrollar un paso de pre-procesamiento durante la ejecución del algoritmo propuesto con el fin de evitar explorar soluciones innecesarias, es decir no cerrar ciertas instalaciones muy prometedoras del líder. Un primer acercamiento a esto es identificar cuales clientes no van a cambiar de compañía al cerrar la instalación al que esté asignado y evitar explorar dicha solución. Con esto se estaría reduciendo considerablemente la región factible del problema.

En cuanto a tiempos de ejecución para resolver nuestro problema podemos observar que el tiempo requerido al utilizar el algoritmo basado en ramificación y acotamiento propuesto es menor que el tiempo invertido para hacer la enumeración completa. Esto no es tan obvio, ya que dentro del algoritmo propuesto se resuelven dos problemas de programación matemática, un problema auxiliar y el nivel inferior. En cambio, en la enumeración completa solo se resuelve un problema, esto es, el nivel inferior para cada solución del líder. A pesar de esto, el tiempo no es muy importante para una decisión estratégica de este tipo.

## 6.2 TRABAJO FUTURO

En esta tesis se resolvió el problema  $\overline{(r|p)}^{\mathcal{L}}$  –Centroide de manera exacta, nosotros propusimos un algoritmo basado en ramificación y acotamien-

to, el algoritmo mostró ser una buena opción para resolver el problema. Sin embargo, solo pudimos resolver instancias de tamaño limitado debido a las limitantes de capacidad computacional o bien, al tiempo requerido para resolverlo.

Es natural pensar en otros enfoques de solución, por ejemplo, se puede desarrollar una metaheurística para resolver el problema. De hecho, la mayoría de los problemas de localización de instalaciones competitivas (CFL, por sus siglas en inglés) -principalmente el bien conocido problema  $(r|p)$ -Centroide- se abordan mediante programación binivel y se proponen metaheurísticas para encontrar soluciones de buena calidad. Se pueden observar los trabajos de Alekseeva *et al.* (2010), Alekseeva & Kochetov (2013), Davydov *et al.* (2012) Davydov *et al.* (2014) por mencionar algunos. Es claro que con ello no aseguramos obtener la solución óptima, pero si tener soluciones de buena calidad a bajo costo computacional.

Además, mencionaremos dos líneas de investigación directas muy interesantes para estudiar extensiones del problema presentado en esta tesis.

### 6.2.1 EL PROBLEMA $\overline{(r|p)}^{\mathcal{L}}$ -CENTROIDE ESTOCÁSTICO.

Es importante mencionar que el problema  $\overline{(r|p)}^{\mathcal{L}}$ -Centroide presentado en esta tesis, fue estudiado desde el punto de vista determinista. Es decir, los parámetros iniciales del problema siempre son conocidos.

Otra manera en la que se puede abordar el problema es mediante programación estocástica, esto es, si se consideran demandas desconocidas o bajo incertidumbre por parte de los clientes. Al hacer esto, en lugar de considerar demandas conocidas se considerarían inciertas, y el modelo de programación bi-nivel cambiaría drásticamente, así como la metodología de solución.

Primero, se agregarían variables aleatorias para considerar la demanda y por ende cambiarían las funciones objetivo de nuestro problema bi-nivel.

Además, el impacto de la incorporación de la incertidumbre en el modelo de programación matemática afectaría la factibilidad y optimalidad del problema en estudio. En este contexto la connotación que tiene el término factibilidad en programación estocástica es más extensa que en el caso determinista, debido a que no se puede garantizar que la solución del modelo estocástico sea factible para todas las posibles opciones de la variable aleatoria.

### 6.2.2 EL PROBLEMA $\overline{(r|p)}^{\mathcal{L}}$ –CENTROIDE EN VARIAS DIMENSIONES.

El siguiente enfoque que se puede abordar consiste en una aplicación directa y tangible del  $\overline{(r|p)}^{\mathcal{L}}$  –Centroide.

Uno de los problemas que actualmente se está trabajando en la literatura, es el estudio del problema de la proliferación de productos, canales y mercados en la cadena de suministro, vea el trabajo de Menezes *et al.* (2016). Hoy en día las empresas se dedican a producir una gran cantidad de variedades de un mismo producto para satisfacer el gusto de los clientes. Por ejemplo, en la industria del yogurt existen una gran variedad de yogurts como el natural, natural azucarado, con frutas, aromatizados, entre muchos otros. Debido a esto, las empresas se ven en la necesidad de aumentar el número de líneas de producción y con esto, tienen una mayor inversión económica.

Por lo que, una de las cuestiones que están abiertas es la racionalización de la cartera de productos de las empresas. En pocas palabras, cómo reducir el número de SKUs (Stock keeping unit, en inglés) o productos finales producidos y distribuidos por la empresa con la finalidad de tener la menor pérdida de consumidores o clientes.

Considere las diferentes variedades de un mismo producto, no es difícil ver que éstas pueden ser representadas en un espacio multidimensional de atributos o “ingredientes” (pensando en yogurts, con los ejes representando

los contenidos de azúcar, grasa, fruta, lactosa, saborizantes, etc), entonces las preferencias de los clientes serían también puntos en ese espacio multi-dimensional. Así como las variedades ofertadas por la empresa competidora también serían atributos.

De esta manera, encontrar los productos y especificaciones a eliminar de la línea de producción de manera óptima es un  $\overline{(r|p)}^{\mathcal{L}}$  –Centroide en varias dimensiones.

# ÍNDICE DE TABLAS

---

3.1. Coordenadas de los clientes $j \in J$ . . . . .	41
3.2. Coordenadas de las instalaciones $i \in I$ . . . . .	41
3.3. Distancia de ir del cliente $j \in J$ a la instalación $i \in I$ . . . . .	41
3.4. Radios de lealtad . . . . .	42
3.5. Tabla conjuntos de ordenamiento para cada cliente $j \in J$ . . . . .	42
5.1. Tamaños de las instancias en la literatura . . . . .	62
5.2. Tamaños de las instancias generadas . . . . .	62
5.3. Resultados instancias de tamaño $I^L = I^F = 10$ , $J = 40$ y $p = r \in \{2, 3, 5\}$ . . . . .	72
5.4. Resultados instancias de tamaño $I^L = I^F = 20$ , $J = 80$ y $p = r \in \{3, 5, 8\}$ . . . . .	72
5.5. Resultados instancias de tamaño $I^L = I^F = 10$ , $J = 100$ y $p = r \in \{2, 3, 4, 5\}$ . . . . .	72
5.6. Resultados instancias de tamaño $I^L = I^F = 12$ , $J = 100$ y $p = r \in \{2, 3, 4, 5, 6\}$ . . . . .	73
5.7. Resultados instancias de tamaño $I^L = I^F = 14$ , $J = 100$ y $p = r \in \{2, 3, 4, 5, 6\}$ . . . . .	73
5.8. Resultados instancias Red de Swain, $\delta^L = \delta^S = 2$ . . . . .	74

---

5.9. Resultados instancias Red extendida de Swain (dándola como dato de entrada el segundo generador) . . . . .	74
---	----

# ÍNDICE DE FIGURAS

---

3.1. Lealtad de un cliente $j$ hacia la empresa a la que es leal sin importar la distancia pero dentro de un radio predefinido (círculo punteado). En blanco se denota una instalación cerrada. . . . .	27
4.1. Rama con corte . . . . .	59
4.2. Rama completa . . . . .	60
5.1. Red de Swain. . . . .	69
5.2. Red de Swain Extendida. . . . .	70
5.3. Comparación de tiempos de la tabla 5.3. . . . .	78
5.4. Comparación de tiempos de la tabla 5.4. . . . .	78
5.5. Comparación de tiempos de la tabla 5.5. . . . .	79
5.6. Comparación de tiempos de la tabla 5.6. . . . .	79
5.7. Comparación de tiempos de la tabla 5.7. . . . .	80
5.8. Comparación de tiempos de la tabla 5.8. . . . .	80
5.9. Comparación de tiempos de la tabla 5.9. . . . .	81

---

APÉNDICE A

# AUTOBIOGRAFÍA

---

**Juan Carlos García Vélez**

CANDIDATO PARA EL GRADO DE MAESTRÍA EN CIENCIAS  
CON ORIENTACIÓN EN MATEMÁTICAS

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN  
FACULTAD DE CIENCIAS FÍSICO-MATEMÁTICAS

TESIS:

EL PROBLEMA  $(r|p)$ –*Centroide Negativo*  
CONSIDERANDO LEALTAD DE LOS CLIENTES

## A.1 ESTANCIAS DE INVESTIGACIÓN

1. *Colegio Universitario de Estudios Financieros, Madrid, España.*

Mayo-Agosto 2018.

Dr. Diego Rafael Ruiz Hernández.

2. *University of Sheffield, Sheffield, Inglaterra.*

Septiembre-October 2018.

Dr. Diego Rafael Ruiz Hernández.

## A.2 CONFERENCIAS

1. *El problema  $(r/p)$ -centroide negativo.* XXVI Escuela Nacional de Optimización y Análisis Numérico, Universidad Juárez Autónoma de Tabasco (3-7 Julio 2017), Tabasco, México.

2. *El problema  $(r/p)$ -centroide negativo.* VI Congreso de la Sociedad Mexicana de Investigación de Operaciones, Universidad Panamericana (4-6 Octubre 2017), Jalisco, México.

3. *The negative  $(r,p)$ -centroid problem considering loyalty of the customers.* 2nd International Workshop on Bilevel Programming (IWOBIP '18) (18-22 Junio 2018), Lille, Francia.

4. *The negative  $(r,p)$ -centroid problem considering loyalty of the customers.* Joint EURO/ALIO International Conference 2018 on Applied Combinatorial Optimization (25-27 Junio 2018), Bologna, Italia.

## BIBLIOGRAFÍA

---

- ALEKSEEVA, E. & KOCHETOV, Y. (2013). Matheuristics and exact methods for the discrete  $(r|p)$ -centroid problem. In: *Metaheuristics for Bi-level Optimization*. Springer, pp. 189–219.
- ALEKSEEVA, E., KOCHETOV, Y. & PLYASUNOV, A. (2015). An exact method for the discrete  $(r|p)$ -centroid problem. *Journal of Global Optimization* **63**(3), 445–460.
- ALEKSEEVA, E., KOCHETOVA, N., KOCHETOV, Y. & PLYASUNOV, A. (2010). Heuristic and exact methods for the discrete  $(r|p)$ -centroid problem. In: *European Conference on Evolutionary Computation in Combinatorial Optimization*. Springer.
- BARD, J. F. (1998). *Practical Bilevel Optimization: Algorithms and Applications*. Dordrecht, Kluwer Academic Publishers.
- BARD, J. F. & MOORE, J. T. (1992). An algorithm for the discrete bilevel programming problem. *Naval Research Logistics (NRL)* **39**(3), 419–435.
- BENATI, S. & LAPORTE, G. (1994). Tabu search algorithms for the  $(r/xp)$ -medianoid and  $(r/p)$ -centroid problems. *CENTRE DE RECHERCHE SUR LES TRANSPORTS PUBLICATION* (941).
- BHADURY, J., EISELT, H. A. & JARAMILLO, J. H. (2003). An alternating heuristic for medianoid and centroid problems in the plane. *Computers & Operations Research* **30**(4), 553–565.

- BHAUMIK, P. K. (2010). Optimal shrinking of the distribution chain: the facilities delocation decision. *International Journal of Systems Science* **41**(3), 271–280.
- BIESINGER, B., HU, B. & RAIDL, G. (2014). An evolutionary algorithm for the leader-follower facility location problem with proportional customer behavior. In: *International Conference on Learning and Intelligent Optimization*. Springer.
- BIESINGER, B., HU, B. & RAIDL, G. (2015). A hybrid genetic algorithm with solution archive for the discrete  $(r|p)$ -centroid problem. *Journal of Heuristics* **21**(3), 391–431.
- BRACKEN, J. & MCGILL, J. T. (1973). Mathematical programs with optimization problems in the constraints. *Operations Research* **21**(1), 37–44.
- BRACKEN, J. & MCGILL, J. T. (1974). A method for solving mathematical programs with nonlinear programs in the constraints. *Operations Research* **22**(5), 1097–1101.
- BRUNO, J. E. & ANDERSEN, P. W. (1982). Analytical methods for planning educational facilities in an era of declining enrollments. *Socio-Economic Planning Sciences* **16**(3), 121–131.
- CAMACHO-VALLEJO, J.-F., CORDERO-FRANCO, Á. E. & GONZÁLEZ-RAMÍREZ, R. G. (2014). Solving the bilevel facility location problem under preferences by a stackelberg-evolutionary algorithm. *Mathematical Problems in Engineering* **2014**.
- CAMPOS-RODRÍGUEZ, C., MORENO-PÉREZ, J. A. & SANTOS-PEÑATE, D. R. (2012). Particle swarm optimization with two swarms for the discrete  $(r|p)$ -centroid problem. In: *International Conference on Computer Aided Systems Theory*. Springer.

- CÁNOVAS, L., GARCÍA, S., LABBÉ, M. & MARÍN, A. (2007). A strengthened formulation for the simple plant location problem with order. *Operations Research Letters* **35**(2), 141–150.
- CARRIZOSA, E., DAVYDOV, I. & KOCHETOV, Y. (2012). A new alternating heuristic for the  $(r|p)$ -centroid problem on the plane. In: *Operations Research Proceedings 2011*. Springer, pp. 275–280.
- CHURCH, R. L. & MURRAY, A. T. (1993). Modeling school utilization and consolidation. *Journal of Urban Planning and Development* **119**(1), 23–38.
- DAVYDOV, I., KOCHETOV, Y. & CARRIZOSA, E. (2012). Vns heuristic for the  $(r|p)$ -centroid problem on the plane. *Electronic Notes in Discrete Mathematics* **39**, 5–12.
- DAVYDOV, I. A., KOCHETOV, Y. A., MLADENOVIC, N. & UROSEVIC, D. (2014). Fast metaheuristics for the discrete  $(r|p)$ -centroid problem. *Automation and Remote Control* **75**(4), 677–687.
- DIAMOND, J. T. & WRIGHT, J. R. (1987). Multiobjective analysis of public school consolidation. *Journal of Urban Planning and Development* **113**(1), 1–18.
- DREZNER, T. & EISELT, H. A. (2002). Consumers in competitive location models. *Facility location: Applications and theory* , 151–178.
- DREZNER, Z. & HAMACHER, H. W. (2001). *Facility location: applications and theory*. Springer Science & Business Media.
- HAKIMI, S. L. (1983). On locating new facilities in a competitive environment. *European Journal of Operational Research* **12**(1), 29–35.
- HANJOUL, P. & PEETERS, D. (1987). A facility location problem with clients' preference orderings. *Regional Science and Urban Economics* **17**(3), 451–473.

- HOTELLING, H. (1929). Stability in competition. *The Economic Journal* **39**(153), 41–57.
- KOCHETOV, Y., KOCHETOVA, N. & PLYASUNOV, A. (2013). A matheuristic for the leader-follower facility location and design problem. In: *Proceedings of the 10th Metaheuristics International Conference (MIC 2013)*, vol. 32. Citeseer.
- KRESS, D. & PESCH, E. (2012a).  $(r|p)$ -centroid problems on networks with vertex and edge demand. *Computers & Operations Research* **39**(12), 2954–2967.
- KRESS, D. & PESCH, E. (2012b). Sequential competitive location on networks. *European Journal of Operational Research* **217**(3), 483–499.
- MENEZES, M. B., RUIZ-HERNÁNDEZ, D. & GUIMARAES, R. (2016). The component commonality problem in a real multidimensional space: An algorithmic approach. *European Journal of Operational Research* **249**(1), 105–116.
- MIGDALAS, A., PARDALOS, P. M. & VÁRBRAND, P. (2013). *Multilevel optimization: algorithms and applications*, vol. 20. Springer Science & Business Media.
- MLADENović, N., BRIMBERG, J., HANSEN, P. & MORENO-PÉREZ, J. A. (2007). The  $p$ -median problem: A survey of metaheuristic approaches. *European Journal of Operational Research* **179**(3), 927–939.
- MURRAY, A. T. & WU, X. (2003). Accessibility tradeoffs in public transit planning. *Journal of Geographical Systems* **5**(1), 93–107.
- REVELLE, C., MURRAY, A. T. & SERRA, D. (2007). Location models for ceding market share and shrinking services. *Omega* **35**(5), 533–540.
- REVELLE, C. S. & SWAIN, R. W. (1970). Central facilities location. *Geographical analysis* **2**(1), 30–42.

- ROBOREDO, M. C. & PESSOA, A. A. (2013). A branch-and-cut algorithm for the discrete  $(r|p)$ -centroid problem. *European Journal of Operational Research* **224**(1), 101–109.
- RODRÍGUEZ, C. M. C., PÉREZ, J. A. M., NOLTEMEIER, H. & PEÑATE, D. R. S. (2009). Two-swarm pso for competitive location problems. In: *Nature Inspired Cooperative Strategies for Optimization (NICSO 2008)*. Springer, pp. 115–126.
- RUIZ-HERNÁNDEZ, D. & DELGADO-GÓMEZ, D. (2016). The stochastic capacitated branch restructuring problem. *Annals of Operations Research* **246**(1-2), 77–100.
- RUIZ-HERNÁNDEZ, D., DELGADO-GÓMEZ, D. & LÓPEZ-PASCUAL, J. (2015). Restructuring bank networks after mergers and acquisitions: A capacitated delocation model for closing and resizing branches. *Computers & Operations Research* **62**, 316–324.
- RUIZ-HERNÁNDEZ, D., ELIZALDE, J. & DELGADO-GÓMEZ, D. (2017). Cournot–stackelberg games in competitive delocation. *Annals of Operations Research* **256**(1), 149–170.
- SERRA, D. & REVELLE, C. (1994). Competitive location in discrete space.
- STACKELBERG, H. v. *et al.* (1952). Theory of the market economy.
- SWAIN, R. W. (1974). A parametric decomposition approach for the solution of uncapacitated location problems. *Management Science* **21**(2), 189–198.
- WANG, Q., BATA, R., BHADURY, J. & RUMP, C. M. (2003). Budget constrained location problem with opening and closing of facilities. *Computers & Operations Research* **30**(13), 2047–2069.

---

WESTON JR, F. C. (1982). Optimal configuration of telephone answering sites in a service industry. *European Journal of Operational Research* **10**(4), 395–405.