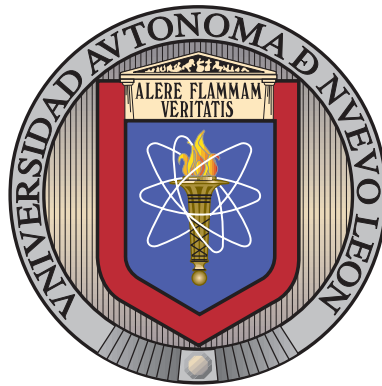


UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN

FACULTAD DE CIENCIAS FÍSICO-MATEMÁTICAS



EVALUACIÓN DE LA CALIDAD DE MÉTODOS DE  
OPTIMIZACIÓN EXACTA PARA MODELOS DE  
DISEÑO TERRITORIAL

POR:

NANCY SOLIS GARCÍA

COMO REQUISITO PARCIAL PARA OBTENER EL GRADO DE  
LICENCIADO EN MATEMÁTICAS

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN

FACULTAD DE CIENCIAS FÍSICO-MATEMÁTICAS



EVALUACIÓN DE LA CALIDAD DE MÉTODOS DE  
OPTIMIZACIÓN EXACTA PARA MODELOS DE  
DISEÑO TERRITORIAL

POR:

NANCY SOLIS GARCÍA

COMO REQUISITO PARCIAL PARA OBTENER EL GRADO DE  
LICENCIADO EN MATEMÁTICAS

**Universidad Autónoma de Nuevo León**  
**Facultad de Ciencias Físico-Matemáticas**

Los miembros del Comité de Tesis recomendamos que la Tesis “**Evaluación de la Calidad de Métodos de Optimización Exacta para Modelos de Diseño Territorial**”, realizada por la alumna Nancy Solis García, con número de matrícula **1147948**, sea aceptada para su defensa como opción al grado de Licenciado en Matemáticas.

El Comité de Tesis

---

Dr. Roger Z. Ríos Mercado

Asesor

---

Dra. Nataliya Kalashnykova

Revisor

---

Dra. Ada M. Álvarez Socarrás

Revisor

Vo. Bo.

---

M.A. Patricia Martínez Moreno

Directora

San Nicolás de los Garza, Nuevo León, Junio de 2008

*A mis padres y  
a los padres de mis padres*

# Agradecimientos

A **M.P.C.** por darme la oportunidad de venir a la Tierra, tener una familia y por proporcionarme cada día los medios necesarios para mejorar en lo espiritual, físico, cultural e intelectual.

A mis **abuelitos** por enseñar a mis padres y a su vez ellos a mi la importancia del trabajo y del amor por la familia.

A mis **padres** por estos 22 años que han dedicado amorosamente al progreso de mi vida y que con sacrificios me han dado la oportunidad de estudiar. Gracias a ellos soy lo que soy. Los amo.

A **Vivis, Sol y Panchito**, mis grandes amigos, por su apoyo, comprensión, cariño y sobre todo por confiar en mí.

A mi **chocolatito** por su amor, paciencia y apoyo en la revisión de mi tesis.

A mis **compañeros de la Facultad de Ciencias Físico-Matemáticas (FCFM)** por brindarme su amistad y por darme ánimo durante la realización de la tesis.

A mis **amigos SUD** por estar conmigo en las buenas y en las malas y por apoyarme en cada decisión que tomo.

A los **maestros y estudiantes del Programa de Posgrado en Ingeniería de Sistemas (PISIS) de la UANL** por compartir conmigo sus conocimientos y experiencias en el desarrollo de tesis, en especial a **Vane, Yajis, Efra y Miguel**, que sin duda me ayudaron a realizar mejor mi trabajo.

A los **maestros Alfredo, Lilia y Mirthala** de la FCFM, que con su ejemplo me han motivado a prepararme más profesionalmente, por cada clase que me impartieron, por su amistad y consejos.

Al **Dr. Roger** por su disposición en compartir sus conocimientos conmigo en los dos veranos científicos y en la realización de mi tesis. Por su paciencia y consejos para concluir este trabajo con éxito.

A la **Dra. Ada y a la Dra. Nataliya** por aceptar ser mis revisores, por sus consejos valiosos en como mejorar el desarrollo de la tesis y por compartir un poco de su conocimiento en cada clase que me impartieron.

Al **PISIS** por permitirme usar instalaciones de su edificio y sus Laboratorios de Cómputo de Alto Desempeño para realizar los experimentos. Sin duda es un lugar muy confortable.

Al **equipo de investigación de diseño territorial del PISIS** por su apoyo y comentarios para que esta tesis se mejorara. Por proporcionarme las herramientas necesarias para la realización de los experimentos.

Al **Consejo Nacional de Ciencia y Tecnología (CONACYT)** por el apoyo económico brindado como asistente de investigación de su proyecto SEP-CONACYT 48499-Y.

A la **UANL** por el apoyo otorgado mediante su Programa de Verano de Investigación Científica y Tecnológica (PROVERICYT) en 2006 y 2007. Este tipo de programas son muy valiosos en el fomento de la investigación a estudiantes de licenciatura y preparatoria. También agradezco el apoyo otorgado por el proyecto UANL-PAICYT CA1478-07.

# Resumen

Este estudio trata sobre un problema de diseño de territorios de atención comercial, el cual puede ser visto como un problema de agrupar pequeñas áreas o unidades básicas dentro de grandes áreas geográficas. Los territorios deben satisfacer ciertas características de planeación por parte de la empresa. Este es un problema de optimización combinatoria clasificado como NP-duro, es decir, que no se puede resolver de manera exacta en tiempo polinomial (razonable) para instancias relativamente grandes, por lo cual representa todo un reto desde la perspectiva de investigación. Desde la perspectiva práctica, este problema es muy importante ya que un diseño eficiente ayudaría a las empresas a distribuir su gran área de trabajo en territorios más pequeños de la mejor manera de acuerdo a sus criterios de planeación.

En este trabajo se presentan cuatro modelos relacionados con el problema en estudio. Estos modelos se clasifican como programas enteros mixtos lineales ya que las variables de decisión son enteras y tanto las restricciones como la función objetivo son lineales. El algoritmo exacto que se utiliza para su resolución es el Método de Ramificación y Acotamiento (MRA), el cual es uno de los métodos más populares en la solución exacta de problemas de programación entera.

En este trabajo se incluye una evaluación empírica del desempeño del MRA al aplicarlo en la resolución de algunas instancias de los modelos mencionados. Esto implica un estudio y evaluación de diversos parámetros algorítmicos que dependen de la estructura del problema y que afectan el comportamiento del método. En la evaluación interesan principalmente la calidad de las soluciones encontradas y el tiempo de cómputo empleado por el algoritmo de solución. En otro experimento se pretende conocer el alcance de los actuales métodos exactos en función del tamaño

de las instancias que pueden tratarse con dichas técnicas. Por último, se estudia la sensibilidad del algoritmo del método con respecto al número de territorios.



# Índice

<b>1. Introducción</b>	<b>1</b>
1.1. Descripción del Problema . . . . .	4
1.2. Naturaleza del Problema . . . . .	6
1.3. Relevancia e Importancia del Trabajo . . . . .	7
1.4. Objetivos . . . . .	8
<b>2. Antecedentes Científicos</b>	<b>9</b>
2.1. Programación Entera . . . . .	10
2.2. Heurísticas . . . . .	13
2.3. Trabajos Relacionados con el Problema de Diseño Territorial de Atención Comercial . . . . .	14
2.4. Otros Trabajos Relacionados con el Problema de Diseño Territorial . . . . .	18
<b>3. Planteamiento del Problema</b>	<b>21</b>
3.1. Descripción del Problema . . . . .	21
3.2. Representación del Modelo en un Grafo . . . . .	22
3.3. Formulación Entera Mixta Lineal (PEML) . . . . .	24
3.4. Modelos Relacionados . . . . .	27
<b>4. Metodología de Solución</b>	<b>31</b>
4.1. Método de Ramificación y Acotamiento . . . . .	31
4.2. Fortalecer el Modelo $A_R$ . . . . .	40

---

<b>5. Estudio Computacional</b>	<b>42</b>
5.1. Objetivos y Datos de Prueba . . . . .	43
5.2. Diseño de Experimentos . . . . .	44
5.3. Resultados . . . . .	47
<b>6. Conclusiones</b>	<b>79</b>
6.1. Conclusiones de los Experimentos . . . . .	79
6.2. Conclusiones de la Tesis . . . . .	80
6.3. Aportaciones del Trabajo . . . . .	81
<b>A. GAMS</b>	<b>82</b>
<b>B. CPLEX</b>	<b>87</b>

# Índice de Figuras

4.1. Ilustración del subproblema $P_k$ . . . . .	33
4.2. Ejemplo de aplicar los conjuntos ordenados especiales. . . . .	37
5.1. Utilizar o no las prioridades en instancias de DS05/(60,4) . . . . .	48
5.2. Utilizar o no las prioridades, en instancias de DS05/(100,5) . . . . .	49
5.3. Usar conjuntos ordenados especiales o no, en instancias DS05/(100,5), modelo $A_R$ . . . . .	51
5.4. Evaluación de sondeos de implicaciones lógicas, en DS05/(100,5) del modelo $A_R$ . . . . .	53
5.5. Regla para seleccionar la variable a ramificar, en instancias DS05/(100,5) en el modelo $A_R$ . . . . .	55
5.6. Nodos en cada territorio formado para instancias DS05/(100,5) . . . . .	62
5.7. Nodos en cada territorio formado para instancias DS10/(100,5) . . . . .	62
5.8. Restricciones válidas en el modelo $A_R$ , en instancias DS05/(100,5) . . . . .	66
5.9. Tiempo de ejecución en 5 instancias DS05/(60,4) del Modelo $A_R$ va- riando $p$ . . . . .	68
5.10. Tiempo de ejecución en 5 instancias DS05/(100,5) del Modelo $A_R$ variando $p$ . . . . .	69
5.11. Tiempo de ejecución en 5 instancias DS05/(60,4) del Modelo $B_R$ va- riando $p$ . . . . .	69
5.12. Tiempo de ejecución en 5 instancias DS05/(100,5) del Modelo $B_R$ variando $p$ . . . . .	70

---

5.13. Tiempo de ejecución en 5 instancias DS05/(60,4) del Modelo $C_R$ variando $p$ . . . . .	70
5.14. Tiempo de ejecución en 5 instancias DS05/(100,5) del Modelo $C_R$ variando $p$ . . . . .	71

# Índice de Tablas

3.1. Número de restricciones de conexidad . . . . .	26
4.1. Implicaciones lógicas . . . . .	38
5.1. Tiempo de cómputo promedio, variando las prioridades para el modelo $A_R$ . . . . .	48
5.2. Tiempo de cómputo promedio, utilizando o no conjuntos ordenados especiales para $A_R$ . . . . .	50
5.3. Tiempo de cómputo promedio, utilizando o no conjuntos ordenados especiales para $B_R$ . . . . .	50
5.4. Tiempo de cómputo promedio, utilizando o no conjuntos ordenados especiales para $C_R$ . . . . .	51
5.5. Tiempo de cómputo promedio, realizando o no sondeos de implicaciones lógicas en el modelo $A_R$ . . . . .	53
5.6. Tiempo de cómputo promedio, regla para seleccionar la variable a ramificar en el modelo $A_R$ . . . . .	55
5.7. Tiempo de cómputo promedio, regla para seleccionar la variable a ramificar en el modelo $B_R$ . . . . .	56
5.8. Tiempo de cómputo promedio, regla para seleccionar la variable a ramificar en el modelo $C_R$ . . . . .	56
5.9. Hipótesis alternativa para las instancias DS05/(60,4) del Modelo $A_R$ .	57
5.10. Hipótesis alternativa para las instancias DS05/(100,5) del Modelo $A_R$ .	57
5.11. Hipótesis alternativa para las instancias DS10/(100,5) del Modelo $A_R$ .	58

5.12. Hipótesis alternativa para las instancias DS05/(60,4) del Modelo $C_R$ .	59
5.13. Hipótesis alternativa para las instancias DS10/(60,4) del Modelo $C_R$ .	59
5.14. Hipótesis alternativa para las instancias DS05/(100,5) del Modelo $C_R$ .	60
5.15. Hipótesis alternativa para las instancias DS10/(100,5) del Modelo $C_R$ .	60
5.16. Tiempo de cómputo promedio, fortalecer el modelo $A_R$ .	63
5.17. Hipótesis alternativa para las instancias DS05/(60,4), para fortalecer $A_R$ .	63
5.18. Hipótesis alternativa para las instancias DS10/(60,4), para fortalecer $A_R$ .	64
5.19. Hipótesis alternativa para las instancias DS05/(100,5), para fortalecer $A_R$ .	64
5.20. Hipótesis alternativa para las instancias DS10/(100,5), para fortalecer $A_R$ .	65
5.21. Tiempo de ejecución en 10 instancias DS05 del Modelo $A_R$ variando $p$ .	67
5.22. Tiempo de ejecución en 10 instancias DS05 del Modelo $B_R$ variando $p$ .	67
5.23. Tiempo de ejecución en 10 instancias DS05 del Modelo $C_R$ variando $p$ .	68
5.24. Hipótesis alternativa para las 5 instancias DS05/(60,4), $p$ -diferente en $B_R$ .	71
5.25. Hipótesis alternativa para las 5 instancias DS05/(100,5), $p$ -diferente en $B_R$ .	72
5.26. Alcance del método para el modelo $A_R$ en la instancia DU150-01 con $n=150$ y $p=10$ .	73
5.27. Alcance del método para el modelo $A_R$ para cuatro instancias con $n=150$ , $p=10$ y COR=0.01.	74
5.28. Alcance del método para el modelo $B_R$ en instancias de diferentes tamaños con $\tau=0.05$ y $p=10$ .	75
5.29. Alcance del método para el modelo $C_R$ en la instancia DU150-01 con $n=150$ y $p=10$ .	76
5.30. Alcance del método para el modelo $C_R$ en cuatro instancias con $n=150$ , $p=10$ y COR=0.01.	77
5.31. Alcance del método para varias instancias el modelo $D_R$ con $\tau=0.05$ .	77

A.1. Modelo en GAMS para una instancia con $n = 5$ , $p = 2$ , tipo DS10 y modelo $A_R$ . . . . .	86
B.1. Algunas opciones de CPLEX . . . . .	90
B.2. Archivo cplex.opt . . . . .	92

# Capítulo 1

## Introducción

El mundo ha sido testigo de un crecimiento dramático en el tamaño y complejidad de las organizaciones en los años posteriores a la Revolución Industrial. Hubo muchos beneficios a partir de este cambio revolucionario, no obstante, surgieron nuevos problemas que ocurren hasta la fecha en muchas empresas. Uno de estos problemas es la tendencia de muchos de los componentes de una organización a convertirse en autónomos, con sus propias metas y valores, dejando a un lado los objetivos de toda la organización. Por ejemplo, lo que es mejor para un componente, puede ir en deterioro de otro. Un problema relacionado con esto es que, a medida que la complejidad aumenta, se vuelve más difícil la asignación de los recursos disponibles a las diferentes actividades de la mejor manera para la organización. Debido a la necesidad de encontrar solución a este tipo de problemas surge la Investigación de Operaciones (IO) [10].

Se dice que el origen de la IO se encuentra en los servicios militares prestados a principios de la Segunda Guerra Mundial. Debido a los esfuerzos bélicos, existía una necesidad urgente de asignar recursos escasos a las distintas operaciones militares y a las actividades dentro de cada operación de la forma más efectiva. Como la explosión industrial seguía su curso al terminar la guerra, un gran número de personas comenzaron a ver que los problemas que se estaban presentando en la industria eran básicamente los mismos problemas a los que se había enfrentado la milicia, pero en un contexto diferente. De esta manera, la investigación de operaciones comenzó a



introducirse en la industria, los negocios y el gobierno.

Actualmente los profesionales de la investigación de operaciones colaboran con los responsables de las empresas en el diseño y mejora de las operaciones y decisiones, resuelven problemas y ayudan en las funciones de gestión, planificación, predicción, aportan conocimiento y ayudan en la toma de decisiones.

El presente estudio trata sobre un problema de diseño de territorios de atención comercial, que surge en una compañía distribuidora de bebidas embotelladas, el cual puede ser visto como un problema de agrupar pequeñas áreas o unidades básicas dentro de grandes áreas geográficas. Los territorios deben satisfacer ciertas características de planeación por parte de la empresa. Este es un problema de optimización combinatoria clasificado como NP-duro [16], es decir, difícil de resolver. En términos generales, un problema de optimización es catalogado como difícil cuando existe una demostración teórica que muestra que no existe ningún algoritmo que garantice encontrar la mejor solución posible (solución óptima) en un tiempo razonable [13]. Esto representa todo un reto desde la perspectiva de investigación. Desde la perspectiva práctica, este problema es muy importante ya que una buena solución ayudaría a las empresas a distribuir su gran área de trabajo en territorios más pequeños de la mejor manera de acuerdo a sus criterios de planeación.

El problema de diseño territorial (PDT) que motiva esta tesis tuvo sus orígenes en los trabajos de Vargas Suárez [20] y Vargas-Suárez, Ríos-Mercado y López [19]. En estos trabajos se busca definir agrupaciones de puntos de ventas que satisfagan ciertos requerimientos impuestos por la empresa que presenta este problema. Para dar solución a su problema, Vargas Suárez primero presenta la modelación matemática del problema y después desarrolla una heurística GRASP con la cual obtiene resultados favorables para su estudio.

Posteriormente Ríos-Mercado y Fernández [16] presentan una extensión al problema de Vargas Suárez en el cual permiten considerar criterios adicionales impuestos

por la empresa. Ellos desarrollan una heurística GRASP reactiva en su estudio. Algunas variaciones del problema desde la perspectiva heurística han sido tratadas en Caballero Hernández, Ríos-Mercado y López [8] y Segura-Ramiro et al. [18]. En todos estos trabajos el principal objetivo es desarrollar una herramienta computacional que dé buenos resultados al problema que se estudia en el menor tiempo posible, es decir se diseñan heurísticas (véase Sección 2.2). Después se realiza la experimentación implementando cada uno de estos algoritmos diseñados. Sin embargo no existe un estudio que aborde el problema o sus variaciones desde la perspectiva exacta. Es por eso que esta tesis conforma una extensión de estas investigaciones donde la principal contribución es proporcionar información relevante que permita medir la calidad de las heurísticas desarrolladas para problemas con estructura similar. Otra contribución importante es que al modificar de una manera inteligente el algoritmo del método exacto, éste converge más pronto al valor óptimo.

Específicamente, en esta tesis se lleva a cabo un estudio de cuatro modelos relacionados con el PDT. Estos modelos varían en sus restricciones y la función objetivo. Los modelos en estudio tienen la forma de Programas Enteros Mixtos Lineales (PEML). Un PEML es un modelo en el que tanto las restricciones como la función objetivo son funciones lineales y las variables de decisión son algunas enteras y otras continuas. En particular, se trabaja con relajaciones lineales del modelo original. La relajación de un problema consiste en eliminar alguna restricción del modelo, lo cual permite ampliar la región factible. En este caso, esta relajación obedece al hecho de que en el modelo original está presente un conjunto de restricciones de orden exponencial el cual es imposible escribir explícitamente. Lo anterior, tiene la ventaja que es relativamente más fácil de resolver y que una solución a la relajación proporciona una cota inferior al óptimo del problema original [10]. El obtener esta cota es importantísimo ya que con esta cota se puede medir la calidad de las heurísticas, además asegura que la solución óptima por lo menos obtiene el valor de esta cota, es decir el óptimo está por encima de la cota inferior (en el caso de minimización). Lo anterior ayuda a ir descartando soluciones que estén por debajo de esa cota.

Además, se lleva a cabo una evaluación empírica del desempeño del método de optimización exacta (de ramificación y acotamiento) al aplicarlo en la resolución de algunas instancias de los diferentes modelos. Esto implica un estudio y evaluación de diversos parámetros algorítmicos que dependen de la estructura del modelo y que afectan el comportamiento del método. Se estudia el alcance del método exacto para estos modelos y la sensibilidad del mismo al variar el número de territorios. En la evaluación interesan principalmente la calidad de las soluciones encontradas y el tiempo de cómputo empleado por el algoritmo de solución.

## 1.1. Descripción del Problema

El análisis del problema a estudiar en este trabajo de investigación busca dar solución a una necesidad real, la cual es resolver la definición de distritos o territorios comerciales para una red metropolitana de distribución de bebidas embotelladas de la localidad de Monterrey. Monterrey es una de las ciudades con mayor consumo de refrescos a nivel mundial [20], por consiguiente, la red metropolitana de distribución del producto está formada por una gran cantidad de “manzanas”. Esta empresa embotelladora de refrescos ha considerado seriamente dividir estas manzanas en grupos más pequeños de la mejor manera satisfaciendo ciertos criterios económicos y geográficos definidos por dicha empresa. Esto ayudará a tener una buena administración y además a dar un buen servicio a los clientes ubicados en cada grupo o territorio.

Cada una de las manzanas contiene puntos de venta. Cada manzana tiene asociada varias medidas de desempeño, las cuales serán simplemente la suma aritmética de las medidas correspondientes de los puntos de venta que la conforman.

En el resto del trabajo, la palabra grupo, distrito y territorio se usan indistintamente y se denota por unidad básica a cada manzana.

Las actividades establecidas y medidas en cada manzana son:

- El número de clientes atendidos (puntos de venta).

- El volumen de ventas logrado en cada unidad básica.

Otra manera de ver el segundo parámetro (volumen de ventas) es como la demanda del producto, medida por cantidad de cajas. La medida de cada actividad del territorio es la suma aritmética de las correspondientes a las unidades básicas que lo integran. Lo ideal es que la carga total de cada territorio formado sea lo más parecido posible con respecto a cada una de estas dos actividades definidas. Sin embargo, dado que es muy difícil lograr un balance simultáneo para las dos actividades definidas, se establece cierto parámetro de tolerancia para cada actividad.

Las unidades básicas son, por definición, áreas geográficas con una ubicación específica dentro de una región. Los territorios formados deben respetar esta ubicación natural y es requisito indispensable que el territorio se forme únicamente con unidades básicas que sean contiguas, es decir, que para poder llegar a cada unidad básica del territorio deben visitarse solamente unidades básicas que pertenezcan al mismo territorio. No es válido cruzar la frontera imaginaria que existe entre cada territorio a fin de dar atención a alguna unidad básica. Esta característica se debe a que existen barreras naturales o creadas por el hombre, las cuales hacen evidente la discontinuidad, como por ejemplo: ríos o arroyos, accidentes topográficos, cadenas montañosas, cruces de ferrocarril, etc. [20]. Lo anterior es importante para la empresa ya que si un repartidor asignado a surtir producto en cierto territorio pasa cerca de una unidad básica a la cual no le corresponde dar servicio, ésta se puede disgustar y puede dar una mala imagen de la empresa.

Por otro lado, la empresa considera un último requisito para realizar el diseño de territorios. Al mismo tiempo que se considera la contigüidad y el balance de los territorios, se debe respetar que cada territorio sea lo más compacto posible, es decir, que las unidades básicas en cada territorio estén lo más cerca posible. Esta restricción es importante puesto que beneficia económicamente a la empresa ya que la vida útil de los camiones se alarga, se ahorra el combustible y tiempo de los repartidores al momento de trasladarse de una unidad básica a otra.

Resumiendo lo anterior se tiene que la empresa desea encontrar el mejor agrupamiento de sus unidades básicas en territorios bien definidos que satisfagan las siguientes restricciones:

- Que cada territorio esté balanceado con respecto a las dos actividades medidas.
- Que las unidades básicas dentro de cada territorio sean contiguas.
- Que cada territorio sea lo más compacto posible.

## 1.2. Naturaleza del Problema

Este problema de diseño de territorios es un problema de optimización combinatoria, es decir aquel problema que, implícita o explícitamente pretende encontrar un objeto combinatorio tal como una permutación, una combinación o un subconjunto que satisface ciertas restricciones y posee una medida de desempeño deseada (por ejemplo: maximizar un valor o minimizar un costo). El conjunto posible de alternativas entre las que se busca el objeto combinatorio deseado es ciertamente finito; sin embargo, éste número finito puede ser astronómicamente grande.

En general, la mayoría de los problemas combinatorios son difíciles de resolver. Esto debido a los siguientes factores:

- El número de objetos combinatorios crece exponencialmente dependiendo del tamaño del problema, alcanzando magnitudes inimaginables, aún para instancias de tamaño moderado.
- El óptimo no está bien caracterizado.
- Para la mayoría de este tipo de problemas no se conocen aún algoritmos capaces de resolverlos de manera exacta en un tiempo razonable.

El problema en estudio es uno de estos problemas combinatorios difíciles de resolver, y si bien puede formularse como un problema lineal entero mixto, el tamaño

de las instancias reales impide la obtención de soluciones exactas mediante los métodos exactos de optimización.

Más técnicamente, el problema que se estudia es un problema de optimización combinatoria NP-duro, es decir, no existen algoritmos exactos que resuelvan instancias grandes (reales) del problema en un tiempo razonable. Por ende, en este trabajo se desea analizar las soluciones a este problema para instancias relativamente pequeñas con un método exacto y así determinar el alcance de este método en función del tamaño de las instancias.

### 1.3. Relevancia e Importancia del Trabajo

Resolver este problema es muy importante ya que ayudaría a las empresas a distribuir su gran área de trabajo en territorios más pequeños de la mejor manera de tal forma que tanto clientes como empleados queden satisfechos y además ahorren tiempo. Este es uno de los problemas más complejos en resolver, lo cual lo hace más atractivo y de mayor interés para los científicos amantes de la optimización.

Para dar buenas soluciones a un problema NP-duro se tienen diferentes herramientas: los métodos exactos, las heurísticas y el desarrollo de cotas. Para este tipo de problemas los métodos exactos en ocasiones encuentran el óptimo pero solo para instancias de tamaño reducido, por lo cual, frecuentemente se recurre al diseño de heurísticas o de cotas. Para medir la calidad de estas heurísticas se requiere conocer el valor óptimo o una cota. El resolver la relajación de un modelo es una manera de obtener una cota para el modelo original, esta cota puede ser de buena calidad y aportar información relevante.

El presente trabajo es de mucha importancia ya que proporciona cotas que pueden medir la calidad de heurísticas desarrolladas para los diferentes modelos en estudio, también sirven como punto de partida para otras investigaciones. Se estudia el algoritmo del método exacto de ramificación y acotamiento en cada una de las

relajaciones de los modelos. Se evalúan diversos parámetros y estrategias algorítmicas que dependen de la estructura del problema y que afectan el comportamiento del método. Al evaluar los diferentes parámetros algorítmicos con un diseño experimental adecuado se pueden obtener los mejores valores para éstos. Con esto se espera que el método exacto resuelva las relajaciones de los diferentes modelos. Cuando el método se trunca, al menos esto ayuda a encontrar soluciones de buena calidad.

## 1.4. Objetivos

Uno de los objetivos principales de la presente investigación es obtener conocimiento, comprender el problema y su modelación. Estudiar los fundamentos de la Programación Entera y como opera el método de ramificación y acotamiento. Aprender a usar una de las herramientas computacionales disponibles (GAMS) para modelar problemas de optimización. Otro objetivo es el de evaluar el comportamiento del método de optimización exacta (Método de Ramificación y Acotamiento) al aplicarlo en la resolución de algunas relajaciones de modelos del problema de diseño territorial. Esto implica un estudio y evaluación de diversos parámetros algorítmicos que dependen de la estructura del problema y que afectan el comportamiento del método. En la evaluación interesa principalmente la calidad de las soluciones encontradas, es decir, la proximidad de la solución encontrada y el óptimo, así como el tiempo de cómputo empleado por el algoritmo de solución. Se pretende conocer el alcance de los actuales métodos exactos en función del tamaño de las instancias que pueden tratarse con dichas técnicas. Además, se estudia la sensibilidad del método exacto con respecto a la variación del número de territorios.

## Capítulo 2

# Antecedentes Científicos

El problema de diseño de territorios a partir de un conjunto de unidades básicas dado ha sido estudiado por investigadores de los diferentes campos de la Investigación de Operaciones debido al gran campo de aplicación. El diseño de territorios es similar a hacer una partición en una región geográfica (o red) en áreas más pequeñas (subredes) con el propósito de lograr una mejor organización y administración de acuerdo con los criterios de interés.

En la literatura se han encontrado trabajos enfocados en la solución del problema de diseño territorial y una manera para resolver este tipo de problemas comienza con una fase de agregación. Esta fase casi siempre consiste en la especificación de centros, en ocasiones ficticios, en cada territorio. Después de esto, se asignan las unidades básicas a estos centros y, al mismo tiempo, se busca optimizar un objetivo definido el cual está sujeto a ciertas restricciones (por ejemplo, de contigüidad, de compacidad, balance de alguna actividad, entre otras).

En este capítulo se presenta una descripción condensada del campo de la Programación Entera (PE) y de las heurísticas. Además, se presentan algunos trabajos que están relacionados íntimamente con el problema en estudio y otros más relacionados con el problema de diseño territorial.



## 2.1. Programación Entera

Una manera de resolver un problema de diseño territorial puede ser a través de modelos de programación entera, la cual presenta las siguientes ventajas:

1. Para instancias pequeñas puede alcanzarse una solución óptima.
2. Es posible formular este problema en un modelador algebraico (por ejemplo, GAMS).

A continuación se dan algunas definiciones breves de conceptos que se utilizan en la construcción de un modelo matemático y por consiguiente en la programación lineal y entera [10].

- El *modelo matemático* de un problema de la vida real (por ejemplo en la industria), es el sistema de ecuaciones y expresiones matemáticas relacionadas que describen la esencia del problema. Un modelo es una representación matemática simplificada de una realidad compleja.
- Las *variables de decisión* son las variables que ayudan a plasmar en un modelo las  $n$  decisiones  $(x_1, x_2, \dots, x_n)$  cuantificables relacionadas unas con otras, para estas variables se deben determinar los valores respectivos.
- La *función objetivo* es la medida cuantitativa del funcionamiento del sistema que se desea optimizar (es decir, minimizar o maximizar), por ejemplo: la minimización de los costos variables de la operación de un sistema eléctrico. Esta expresión se puede ver como una función matemática, la cual depende de las variables de decisión, (por ejemplo:  $z = 3x_1 + 2x_2 + \dots + 3x_n$ ).
- Una *restricción* es la representación matemática de una limitación que se puede imponer sobre los valores de las variables de decisión, casi siempre en forma de ecuaciones o desigualdades (por ejemplo:  $3x_1 + 5x_2 \leq 10$ ).

La Programación Lineal (PL) es considerada por muchos como uno de los avances científicos más importantes que tuvo sus orígenes a mediados del siglo XX [10]. En la actualidad es una herramienta común que ha ahorrado miles o millones

de dólares a muchas compañías y negocios, incluyendo industrias en diferentes países del mundo.

La PL utiliza un modelo para describir el problema, la palabra programación se toma como sinónimo de planeación y la palabra lineal significa que todas las funciones del modelo incluyendo la función objetivo deben ser funciones lineales, es decir, la PL trata la planeación de actividades para obtener un buen resultado. El problema general más estudiado con esta herramienta es el de asignar recursos limitados entre actividades competitivas de la mejor manera posible (óptima).

En muchos casos, los problemas a resolver sólo tienen sentido si las variables de decisión son enteras (por ejemplo, al momento de asignar personas). Si exigir que las variables de decisión tomen valores enteros es la única diferencia entre el modelo de un problema y el modelo de PL, entonces, se trata de un problema de Programación Entera (PE).

El modelo matemático para PE es el mismo modelo de PL agregando la restricción de que las variables tengan valores enteros. Si se exige que solo algunas de las variables tengan valores enteros, entonces el modelo es de Programación Entera Mixta (PEM), y si toda variable de un modelo es entera se llama PE pura. Está garantizado que los problemas de PE pura con una región factible acotada tienen solo un número finito de soluciones factibles, sin embargo, existen dos argumentos erróneos a este razonamiento:

- Este número puede ser astronómicamente grande, aún las computadoras más eficientes son incapaces de verificar cada solución y, si es posible, calcular el valor de la función objetivo.
- Se piensa que al eliminar algunas soluciones factibles (las no enteras) de un problema de PL será más fácil resolverlo, pero todo lo contrario, solo cuando todas estas soluciones factibles están ahí se puede garantizar la existencia de una solución factible en el vértice que es óptima para el problema completo.

De esto, se encuentra que es mucho más sencillo resolver los problemas de PL que los de PE. Existen algunos tipos especiales de problemas de PE para los cuales al resolverlos como PL se garantiza la solución entera, debido a su estructura especial. Un par de ejemplos de éstos son:

- El problema de transporte.
- El problema de asignación.

Los dos factores determinantes de la dificultad computacional de un problema de PE son:

1. El número de variables enteras.
2. La estructura del problema

Para resolver estos problemas de PE suele ser tentador usar el procedimiento aproximado y aplicar el método simplex a la *relajación* de PL (consiste en eliminar las restricciones enteras del modelo, es decir, considerar a las variables como continuas o reales en lugar de enteras), y después redondear los valores no enteros a enteros en la solución obtenida. Sin embargo, esto solo puede ser adecuado en algunas aplicaciones, en especial si los valores de las variables son tan grandes que el redondeo causa un error muy pequeño. Al hacer lo anterior se corren dos riesgos:

1. La solución óptima de PL no necesariamente es factible después de redondearla.
2. No existe garantía de que esta solución redondeada sea la solución óptima del problema original (de PE).

Existe una mejor manera de tratar problemas grandes de PE, la cual es, usar uno de los algoritmos heurísticos disponibles, los cuales son muy eficientes para problemas grandes, pero no garantizan llegar a una solución óptima, pero sí suelen ser más efectivos que el redondeo para encontrar soluciones factibles muy buenas. El desarrollo de algoritmos de PE sigue siendo un tema de investigación y se continúa analizando la eficiencia de los ya existentes. Más adelante se analizará el Método de Ramificación y Acotamiento (MRA), el cual es un algoritmo exacto, de los más populares de PE.

## 2.2. Heurísticas

En la mayoría de los trabajos de investigación que han estudiado variaciones del problema en estudio se han diseñado heurísticas. En la siguiente sección se presentan algunos de éstos trabajos. Para comprender mejor lo que se realizó en cada uno de estos estudios se presenta la definición de heurística así como algunas ventajas que presenta el usar estas técnicas.

Las *heurísticas* son procedimientos simples, a menudo basados en el sentido común, que se supone ofrecerán una buena solución (aunque no necesariamente la óptima) a problemas difíciles, de un modo fácil y rápido [3]. Existen factores que pueden hacer interesante la utilización de algoritmos heurísticos para la resolución de un problema, por ejemplo:

- Cuando no existe un método exacto de resolución o éste requiere mucho tiempo de cálculo o memoria.
- Cuando no se necesita la solución óptima.
- Cuando los datos son pocos fiables.
- Cuando limitaciones de tiempo o espacio obliguen al empleo de métodos de rápida respuesta.
- Como paso intermedio en la aplicación de otro algoritmo.

Las heurísticas tienen algunas ventajas, entre ellas:

- Permiten mayor flexibilidad para el manejo de las características del problema.
- Pueden ofrecer más de una solución, lo cual permite ampliar las posibilidades de elección del que decide.
- Suele ser más fácil de entender la formulación que los complejos métodos matemáticos que utilizan la mayoría de las técnicas exactas.

Algunos tipos de heurísticas son:

- Métodos constructivos.
- Métodos de descomposición.
- Métodos de reducción.
- Manipulación del modelo.
- Métodos de búsqueda por entornos.

GRASP es la heurística que se ha empleado en la mayoría de los trabajos más relacionados con el problema en estudio, por tal motivo, se presenta una descripción breve de esta técnica.

GRASP (por sus siglas en inglés, Greedy Randomized Adaptive Search Procedures) es un ejemplo de una heurística surgida en la década de los 80 con el fin de resolver problemas difíciles en el campo de la optimización combinatoria [3]. Una traducción literal de las palabras que forman el acrónimo es “Procedimientos de Búsqueda Voraces, Aleatorizados y Adaptativos”. Cada una de estas palabras caracteriza una de las componentes de esta metaheurística, la cual, dirige la mayor parte de su esfuerzo en construir soluciones de alta calidad que son posteriormente procesadas con el fin de conseguir otras aún mejores. La técnica general empleada por esta metaheurística es de tipo iterativo. Cada iteración de GRASP consiste de dos fases: construcción y postprocesamiento.

### **2.3. Trabajos Relacionados con el Problema de Diseño Territorial de Atención Comercial**

El problema en estudio está íntimamente ligado con los trabajos de investigación de: Vargas Suárez [20], Vargas-Suárez, Ríos-Mercado y López [19], Ríos-Mercado y Fernández [16], Caballero Hernández, Ríos-Mercado y López [8] y Segura-Ramiro et al. [18]. Todos estos trabajos buscan, al igual que el problema en estudio, dar solución a un problema real presente en una empresa distribuidora de refrescos de la localidad de Monterrey, N. L. La empresa desea particionar su gran área de

trabajo en grupos más pequeños que cumplan con ciertos criterios de planeación impuestos por la misma, esto ayudará a tener una mejor administración de la empresa y además a dar un buen servicio a sus clientes. En la mayoría de éstos trabajos se han diseñado algoritmos basados en heurísticas para dar solución de buena calidad a instancias grandes del problema en diferentes versiones.

A continuación se describen cada uno de estos trabajos, la aportación más relevante que brindaron y las diferencias que existen entre cada uno de estos trabajos con el presente.

En Vargas Suárez [20] y Vargas-Suárez, Ríos-Mercado y López [19] se busca definir agrupaciones de puntos de ventas de manera que el conjunto de grupos o territorios resulte ser balanceado con respecto a tres actividades nodales medibles (número de clientes, demanda y carga de trabajo) y que a su vez cumplan con restricciones de contigüidad territorial y de elegibilidad impuestas por la empresa. En ese trabajo no se considera criterio de compacidad y se maneja un número variable de territorios. Debido a que dicho problema es un problema NP-duro, los autores proponen una metodología de solución basada en la metaheurística GRASP (véase Sección 2.2). El objetivo fundamental de ese trabajo fue el de diseñar y construir una herramienta computacional (heurística) que permitiera encontrar una solución de buena calidad para su problema en un tiempo razonable. Su algoritmo incluye una fase de preprocesamiento que se enfoca en hacer cumplir las restricciones de asignación conjunta, las cuales aseguran que ciertas manzanas dadas pertenezcan al mismo grupo. Las demás restricciones del problema se manejan a través de la mecánica del algoritmo. La contribución más importante de su trabajo fue la modelación matemática de un problema extraído del ámbito empresarial local y la aplicación de su estudio a la construcción de un procedimiento heurístico que no había sido intentado anteriormente en problemas de ese tipo.

Ríos-Mercado y Fernández [16] introducen una extensión al problema anterior que permite una modelación más robusta y considerar criterios adicionales. Consideran el número de territorios  $p$  fijo y buscan que los territorios sean compactos.

Es decir, buscan dividir un conjunto entero de unidades básicas en  $p$  territorios, los cuales deben ser lo más compactos posibles y también requieren que los territorios encontrados estén balanceados con respecto a cada una de tres actividades definidas en cada unidad básica (número de clientes, demanda y carga de trabajo). Además buscan que estos territorios sean conexos o contiguos. Estos requerimientos son con el fin de satisfacer las necesidades de la empresa y para una mejor administración de la misma. Ellos consideran la distancia euclidiana para medir la dispersidad de los territorios. La función objetivo que proponen en el modelo mide la dispersidad de los territorios y está basada en el  $p$ -centro. Para obtener soluciones para su problema ellos proponen una implementación avanzada de GRASP denominada GRASP reactivo, donde el parámetro  $\alpha$ , que controla la lista restringida de candidatos en la fase constructiva, no es fijo sino que es automáticamente ajustado durante la ejecución del algoritmo basado en la historia de las soluciones generadas. Dado que el mayor consumo de tiempo se emplea en la fase de búsqueda local, ellos también estudian el impacto de aplicar un filtro para evitar ejecutar esta fase en soluciones malas o poco prometedoras generadas en la fase constructiva. Este algoritmo lo prueban en varios conjuntos de instancias generadas aleatoriamente a partir de datos proporcionados por el socio industrial. En su evaluación empírica, se concluye la eficiencia y robustez del GRASP reactivo al compararlo con un GRASP ordinario (con  $\alpha$  fijo). Además se muestra contundentemente que su método mejora notablemente la calidad de soluciones halladas por la empresa. La diferencia entre el problema que ellos estudian y el presente es que consideran tres actividades (número de clientes, demanda, carga de trabajo). Lo relevante de su trabajo es la exitosa aplicación de un GRASP con componentes avanzados de reactividad y filtrado lo cual no había sido considerado anteriormente para problemas de diseño territorial en general.

Otras variaciones del problema, también desde la perspectiva heurística, han sido abordadas en [8] y [18]. Por ejemplo, Caballero Hernández, Ríos-Mercado y López [8] abordan este problema considerando además restricciones de asignación conjunta, es decir, dos unidades marcadas de esta forma deben pertenecer al mismo territorio. Se mantienen las restricciones de conectividad y balance territorial (con

respecto a dos actividades). El número de territorios  $p$  es conocido. Para dar solución a su problema proponen un GRASP, con el cual, en la primera fase construyen una solución factible con respecto a las restricciones de asignación conjunta y de contigüidad. En esta fase ponen un mayor esfuerzo en obtener soluciones de calidad respecto a la medida de dispersidad y permiten cierta violación de las restricciones de balanceo. La idea de esta etapa es la de unir territorios iterativamente hasta alcanzar el número deseado ( $p$ ). El algoritmo arranca con  $n$  territorios, es decir que se asigna cada nodo a un territorio distinto. Después de la fase constructiva se ejecuta una etapa de postprocesamiento en la que se lleva a cabo una búsqueda local. En esta etapa se busca mejorar el valor de la función objetivo y recuperar factibilidad respecto a las restricciones de balanceo que son violadas. Realizaron pruebas en varios conjuntos de instancias generadas a partir de datos reales proporcionados por un socio industrial. Lo relevante de su trabajo es que, hasta donde se tiene conocimiento, no se había estudiado este tipo de problema considerando restricciones de asignación conjunta.

Segura-Ramiro et al. [18] buscan encontrar territorios balanceados con respecto a cada una de dos actividades que miden (número de clientes y demanda). También requieren que los territorios sean contiguos y compactos. El número de territorios  $p$  se conoce y es fijo. La novedad del trabajo es que ellos consideran una medida de dispersión basada en  $p$ -mediana. Su método propuesto consiste en un proceso iterativo de dos etapas donde primero los centros de los territorios son localizados y después los clientes se asignan a estos centros. La técnica que usan ha sido diseñada anteriormente para resolver solamente problemas que involucran restricciones de balance únicas. Sin embargo, en su trabajo ellos extienden la técnica y proporcionan una adaptación para tratar restricciones de balance múltiple y restricciones de contigüidad simultáneamente. Para la fase de localización ya se cuenta con territorios formados y ahora desean recalcular nuevos centros para una siguiente fase de asignación. Para cada territorio se selecciona aquel nodo para el cual se minimiza la suma de las distancias de dicho nodo a todos los demás nodos del territorio. En la fase de asignación ya se cuenta con los centros de los territorios, ahora es necesario



asignar cada una de las unidades básicas a un centro para formar así los territorios. En esta fase presentan un “modelo de asignación” (similar al modelo del problema de transporte) en el cual las variables de decisión se consideran continuas y la función objetivo es una medida basado en  $p$ -mediana, con la cual miden la compacidad. Después de resolver las divisiones generadas por considerar a las variables de decisión como continuas proceden a entrar en la fase de búsqueda local. Estas fases se realizan iterativamente hasta satisfacer un criterio de parada, encontrando así el mejor diseño de territorios que cumple con los requerimientos impuestos por la empresa. Ellos encontraron que el utilizar la función de  $p$ -mediana como medida de dispersión favorece a la conexidad, ya que la mayoría de los territorios resultaron conexos (en instancias relativamente grandes).

## 2.4. Otros Trabajos Relacionados con el Problema de Diseño Territorial

En los años sesentas, debido a los problemas a los que se enfrentaban las empresas y la industria, se inicia la formulación de modelos matemáticos para el problema de diseño territorial (PDT). Muchos autores han investigado problemas de diseño de territorios y han desarrollado modelos y algoritmos en varios contextos. Los más importantes son: diseño de territorios de ventas y diseño de distritos políticos. Aquí se presentan en forma resumida algunos de los trabajos más significativos para el problema de diseño territorial.

El interés en el problema de diseño de territorios tiene sus orígenes en el ámbito político. Hess y Samuels [9] hacen una analogía entre el diseño de territorios con fines políticos y el de ventas. Mencionan que el problema de diseño de distritos políticos conocido como el problema “un hombre-un voto” puede aplicarse de manera muy similar en otros contextos, entre ellos, el problema presente en su trabajo el cual está enfocado en la formación de territorios de atención comercial. En su problema buscan que los territorios sean lo más compactos posibles (este factor se mide en base a la distancia euclidiana, en la función objetivo), que cada territorio esté lo más

balanceado posible respecto a cada actividad medida y aunque para ellos la contigüidad es muy importante no la consideran en su formulación. Mencionan que al manejar forzosamente la compacidad generalmente se evita la discontigüidad de los territorios. Hess y Samuels proponen un modelo similar al problema de agrupamiento, solo que ellos agregan a este modelo la restricción de balanceo para las actividades medidas. Para dar solución a este problema usan su heurística (GEOLINE), la cual construye un número predeterminado de territorios de ventas compactos constituidos de pequeñas unidades básicas. Las actividades medidas de los vendedores de territorio a territorio son restringidas a ser casi iguales. El procedimiento de cálculo consiste en una secuencia de programas lineales de transporte. Su enfoque de solución es asignar unidades básicas (para esto, resuelven un problema de transporte) a centros de distritos previamente calculados. Además de buscar dar solución a este problema los autores mencionan diferentes actividades de interés que han sido desarrolladas por varias compañías de acuerdo a sus necesidades, las cuales pueden ser consideradas en problemas futuros. Su trabajo es considerado muy importante por haber sido uno de los primeros en proponer modelos y algoritmos para problemas de diseño de territorios, en especial para los de atención comercial.

Fleischmann y Paraschis [5] trabajan con otro modelo con el cual buscan resolver el problema de definir territorios o asignar 168 agentes de ventas en 1400 áreas postales (éstas constituyen las unidades básicas), el cual surge en una empresa alemana de bienes de consumo. De acuerdo a los requerimientos de la empresa (la compacidad, dado que se requiere que el costo total de traslado dentro del territorio sea mínimo, y el balance de algunas actividades medidas) ellos formulan el problema como un modelo de programación matemática. La condición de que cada área sea asignada solamente a un territorio la representan por medio de una restricción contenida en el modelo. Para medir la compacidad de los territorios ellos usan la distancia euclidiana, similar al procedimiento GEOLINE de Hess y Samuels [9]. Para calcular esta distancia usan las coordenadas geográficas de cada unidad básica, es decir, su ubicación geográfica. Resuelven el problema aplicando un procedimiento de localización-asignación, sin embargo, la dimensión de su problema y la estructura

de los datos no asegura una solución factible en el caso sencillo (origen). Esto los motivó a aplicar un algoritmo lineal de transportación. Para resolver la división de unidades básicas (llamado problema de divisiones) aplican una heurística. El modelo es el mismo que el famoso modelo de transporte con excepción de que los centros de los distritos, y por lo tanto, las distancias, se desconocen. Debido a esto, el problema se resuelve como un problema de optimización iterativa de localización-asignación, esto es, se proporciona una ubicación arbitraria inicial de los centros y se calculan las distancias. Después de esto, se asignan las unidades a los centros resolviendo el problema de transporte. La solución obtenida puede tener una unidad básica fraccionada en varios distritos (esto lo hace el algoritmo para mantener el balance en las actividades), para resolver estas divisiones ellos desarrollan una nueva heurística para realizar asignaciones factibles sencillas, a la cual le llaman “heurística de asignación local”. En todos los casos que probaron, la heurística de asignación local dio mejores resultados que la heurística basada en el problema de transporte. La facilidad de todo el procedimiento y los resultados finales reportados a la empresa fueron satisfactorios.

Una colección más amplia de literatura sobre el problema de diseño territorial pueden encontrarse en el trabajo reciente de Kalcsics, Nickel y Schröder [11], en donde presentan una revisión amplia de problemas de este tipo, sus aplicaciones y proponen soluciones para resolver estos problemas. En su trabajo, identifican características similares presentes en muchos problemas de diseño territorial, introducen un modelo base y presentan a detalle dos propuestas para resolver este modelo: un método clásico de localización-asignación basado en trabajos previos, y un recién desarrollado método computacional basado en aspectos geométricos. Presentan resultados que evalúan la eficiencia de los métodos usados para resolver problemas prácticos de gran escala. Concluyen su trabajo discutiendo extensiones para el modelo básico.

# Capítulo 3

## Planteamiento del Problema

El tema principal de este capítulo son los modelos de programación matemática derivados del problema descrito en el Capítulo 1. Se aborda a detalle el problema objeto de esta investigación, la representación del modelo en un grafo y los cuatro modelos planteados que servirán mas adelante en la experimentación.

### 3.1. Descripción del Problema

Cada unidad básica representa a una manzana y se conoce el número de estas unidades, el cual está identificado por  $n$ . Todas éstas unidades se analizan dentro del Plano Cartesiano, por lo tanto, se tienen las coordenadas geográficas que representan la ubicación de cada unidad. Se tienen los valores de las dos actividades medibles (número de clientes y demanda) asociadas a cada unidad básica.

El número de territorios se conoce y se denota por  $p$ . Como se mencionó anteriormente, es muy difícil lograr un balance simultáneo para las dos actividades nodales, por tal motivo se fija un valor de tolerancia ( $\tau^a$ ) el cual permite decidir que tan balanceados se desean que estén los territorios para cada actividad. Se supone que la demanda y el número de clientes se conocen con certeza, y que el número de clientes es fijo. Cada unidad básica debe pertenecer solamente a un territorio. Se busca balancear cada territorio respecto a las dos medidas de actividad, además que cada territorio sea contiguo y compacto.

## 3.2. Representación del Modelo en un Grafo

A fin de comprender mejor la representación del modelo en un grafo, enseguida, se mencionan algunas definiciones de teoría de grafos [4]:

- Un *vértice* o *nodo* es la unidad fundamental de la que están formados los grafos. Los vértices son tratados como un objeto indivisible y sin propiedades, aunque puedan tener una estructura adicional dependiendo de la aplicación por la cual se usa el grafo. Por ejemplo, una red semántica es un grafo en donde los vértices representan conceptos o clases de objetos.
- Un *grafo* es una pareja  $G=(V,E)$ , donde  $V$  es un conjunto de puntos, llamados vértices, y  $E$  es un conjunto de pares de vértices, llamadas aristas.
- Un *subgrafo* de un grafo  $G$  es un grafo cuyos conjuntos de vértices y aristas son subconjuntos de los de  $G$ .
- Un *camino* entre el vértice  $v_1$  y el vértice  $v_k$  de un grafo  $G=(V,E)$  es una sucesión de aristas  $\{e_1, e_2, \dots, e_k\}$  tales que  $e_i = (v_{i-1}, v_i)$  para  $i=2, 3, 4, \dots, k$ , con lo que cada arista tiene por vértice inicial el vértice final de la anterior.
- Un grafo es *conexo* si cada par de vértices está conectado por un camino; es decir, si para cualquier par de vértices  $(i,j)$ , existe al menos un camino posible desde  $i$  hacia  $j$ .

Prácticamente, cualquier red puede ser modelada en un grafo: una red de carreteras que conecta ciudades o una red eléctrica o un alcantarillado, por ejemplo.

Este problema ha sido estudiado anteriormente. Ríos-Mercado y Fernández [16] dan una representación a este problema por medio de un grafo. El problema definido previamente es modelado por un grafo  $G = (V,E)$ , donde una manzana o unidad básica  $i$  es asociada a un nodo, y un arco está presente entre los nodos  $i$  y  $j$  si las manzanas  $i$  y  $j$  son adyacentes. Ahora, cada nodo  $i \in V$  tiene ciertos parámetros asociados tal como sus coordenadas geográficas  $(c_i^x, c_i^y)$ , y dos medidas de actividad. Sea  $w_i^a$  el valor de la actividad  $a$  en el nodo  $i$ , donde  $a = 1$  (número de clientes),

$a = 2$  (demanda o volumen de venta). Un territorio es un subconjunto de nodos  $V_k \subset V$ . El número de territorios está dado por el parámetro  $p$ . Para este problema es requisito que cada nodo sea asignado solamente a un territorio. De esta manera, los territorios definen una partición de  $V$ . Una de las propiedades que se busca en una solución es que los territorios estén balanceados con respecto a cada medida de actividad. También, se permite definir el tamaño del territorio  $V_k$  con respecto a la actividad  $a$  como:  $w^a(V_k) = \sum_{i \in V_k} w_i^a$  donde  $a = 1, 2$ . Debido a la estructura discreta del problema y a la restricción de asignación única, es prácticamente imposible tener un perfecto balanceo en los territorios con respecto a cada medida de actividad. Para manejar esto, se mide el grado de balanceo calculando la desviación relativa de cada territorio con respecto al tamaño medio  $\mu^a$ , siendo  $\mu^a = w^a(V)/p$ ,  $a = 1, 2$ . Además, se introduce el parámetro de tolerancia  $\tau^a$  donde  $a = 1, 2$ , el cual permite tener cierta desviación de la meta deseada ( $\mu^a$ ). Otra característica importante es que todas las unidades básicas asignadas a cada territorio están conectadas por una ruta que está contenida totalmente dentro del territorio. En otras palabras, cada territorio  $V_k$  debe inducir un subgrafo conexo de  $G$ . Adicionalmente, se requiere que en cada territorio, las unidades básicas estén relativamente cerca entre ellas. Una forma de lograr esto para cada territorio es seleccionar un nodo apropiado para ser su centro, y entonces definir una medida de dispersión como por ejemplo,  $D = \max_{1 \leq k \leq p} \max_{j \in V_k} d_{c(k),j}$ , donde  $c(k)$  denota el índice del centro del territorio  $k$  y  $d_{c(k),j}$  representa la distancia Euclidiana del nodo  $j$  al centro del territorio  $k$ . Otra medida de dispersión es la de  $p$ -mediana, la cual se menciona más adelante (en el Modelo B, Sección 3.4). Se supone que se conocen todos los parámetros con certeza. El problema se puede describir entonces como el de encontrar una  $p$ -partición de  $V$  satisfaciendo los criterios de planeación de balanceo, contigüidad, y compacidad que minimiza la medida de distancia anterior. Nótese que minimizar la dispersión de un territorio es equivalente a maximizar su compacidad.

### 3.3. Formulación Entera Mixta Lineal (PEML)

Desde el punto de vista de la programación matemática, el problema puede modelarse como un programa entero mixto lineal.

#### Índices y conjuntos

$n$	número de unidades básicas o nodos
$p$	número de territorios
$i, j$	índices de las unidades básicas; $i, j \in V = \{1, 2, \dots, n\}$
$a$	índices de las actividades; $a \in A = \{1, 2\}$
$N^i$	(= $\{j \in V : (i, j) \in E \vee (j, i) \in E\}$ ) conjunto de nodos que son adyacentes al nodo $i$ ; $i \in V$

#### Parámetros

$w_i^a$	valor de la actividad $a$ en el nodo $i$ ; $i \in V, a \in A$
$d_{ij}$	distancia euclidiana entre $i$ y $j$ ; $i, j \in V$
$\tau^a$	tolerancia relativa con respecto a la actividad $a$ ; $a \in A, \tau^a \in [0, 1]$

#### Parámetros calculados

$w^a(B)$	(= $\sum_{j \in B} w_j^a$ ) tamaño del conjunto $B$ con respecto a $a$ ; $a \in A, B \subset V$
$\mu^a$	(= $w^a(V)/p$ ) valor promedio (meta) de la actividad $a$ ; $a \in A$

#### Variables de decisión

Se introducen variables binarias basadas en centros para modelar la medida de dispersidad.

$$x_{ij} = \begin{cases} 1 & \text{si la unidad } j \text{ es asignada al territorio con centro en } i; i, j \in V \\ 0 & \text{de otra manera} \end{cases}$$

Por consiguiente, esto implica la definición de  $x_{ii}$  como:

$$x_{ii} = \begin{cases} 1 & \text{si el nodo } i \text{ es el centro de un territorio; } i \in V \\ 0 & \text{de otra manera} \end{cases}$$

## Modelo A

$$\text{Minimizar } f(x) = \max_{i,j \in V} \{d_{ij}x_{ij}\} \quad (\text{A.1})$$

$$\text{Sujeto a: } \sum_{i \in V} x_{ij} = 1 \quad j \in V \quad (\text{A.2})$$

$$\sum_{i \in V} x_{ii} = p \quad (\text{A.3})$$

$$\sum_{j \in V} w_j^a x_{ij} \leq (1 + \tau^a) \mu^a x_{ii} \quad i \in V, a \in A \quad (\text{A.4})$$

$$\sum_{j \in V} w_j^a x_{ij} \geq (1 - \tau^a) \mu^a x_{ii} \quad i \in V, a \in A \quad (\text{A.5})$$

$$\sum_{j \in \cup_{v \in S} N^v \setminus S} x_{ij} - \sum_{j \in S} x_{ij} \geq 1 - |S| \quad i \in V; S \subset V \setminus (N^i \cup \{i\}) \quad (\text{A.6})$$

$$x_{ij} \in \{0, 1\} \quad i, j \in V \quad (\text{A.7})$$

La función objetivo (A.1) mide la dispersidad de los territorios usando la función conocida como  $p$ -centro. Las restricciones (A.2) y (A.3) garantizan que cada nodo  $j$  sea asignado solamente a un territorio y que el número de territorios diseñados sea el deseado, respectivamente. Las restricciones (A.4) y (A.5) representan el balance en cada territorio con respecto a cada medida de actividad, ya que se sabe que el tamaño de cada territorio debe encontrarse dentro de un rango (medido por el parámetro de tolerancia  $\tau^a$ ) alrededor de este promedio. Además las cotas superiores de las restricciones de balance (A.4) aseguran o garantizan que si el nodo  $i$  no es un centro, los clientes no pueden asignarse a éste. Las restricciones (A.6) garantizan la contigüidad de los territorios. Nótese, que el número de estas restricciones crece exponencialmente dependiendo del tamaño del conjunto  $V$ . Las restricciones (A.7) garantizan un valor binario para cada variable de decisión. El problema en estudio es NP-duro [16].

Este modelo no es precisamente lineal debido a que la función objetivo (A.1) no es una función lineal. Sin embargo, el modelo A puede ser linealizado si se reemplaza (A.1) por (A.8) y (A.9) dadas a continuación.



$$\text{Minimizar } g(x) = z \tag{A.8}$$

$$z \geq d_{ij}x_{ij} \quad i, j \in V \tag{A.9}$$

A su vez, las restricciones (A.9) se reemplazan por

$$z \geq \sum_{i \in V} d_{ij}x_{ij} \quad j \in V \tag{A.10}$$

ya que éstas proporcionan una formulación más robusta [2].

Como se mencionó anteriormente, el número de restricciones dadas en (A.6) es exponencial y por tanto no pueden escribirse explícitamente ni para valores relativamente medianos de  $n$ . La fórmula para calcular el número de restricciones que se desprenden de las restricciones de conectividad(A.6) es:

$$\sum_{i=1}^{|V|} \left[ \sum_{l=1}^{c_i} \binom{c_i}{l} \right], \text{ donde } c_i = |V \setminus (N^i \cup \{i\})|; i \in V$$

En la Tabla 3.1 se muestra el número de restricciones para algunos valores de  $n$ .

$n$	$p$	Número de Restricciones (A.6)
10	3	854
20	5	1,834,988

**Tabla 3.1:** Número de restricciones de conectividad

Por tal motivo, en el presente trabajo se procedió a trabajar con una relajación del modelo  $A$  que consiste en ignorar (A.6).

*Modelo  $A_R$  (Modelo  $A$  linealizado y relajado)*

$$\text{Minimizar} \quad g(x) = z \quad (\text{A.11})$$

$$\text{Sujeto a:} \quad z \geq \sum_{i \in V} d_{ij} x_{ij} \quad j \in V \quad (\text{A.12})$$

$$\sum_{i \in V} x_{ij} = 1 \quad j \in V \quad (\text{A.13})$$

$$\sum_{i \in V} x_{ii} = p \quad (\text{A.14})$$

$$\sum_{j \in V} w_j^a x_{ij} \leq (1 + \tau^a) \mu^a x_{ii} \quad i \in V, a \in A \quad (\text{A.15})$$

$$\sum_{j \in V} w_j^a x_{ij} \geq (1 - \tau^a) \mu^a x_{ii} \quad i \in V, a \in A \quad (\text{A.16})$$

$$x_{ij} \in \{0, 1\} \quad i, j \in V \quad (\text{A.17})$$

$$z \geq 0 \quad (\text{A.18})$$

### 3.4. Modelos Relacionados

Como se mencionó en la revisión de literatura, existen otros modelos que han sido abordados desde la perspectiva heurística. Por tal motivo, es también de interés en esta tesis aportar un estudio sobre estos otros modelos, los cuales se describen a continuación.

*Modelo  $B$* : La formulación de este modelo es igual a la del modelo  $A$ , con excepción de la función objetivo. En el modelo  $A$  se busca minimizar la máxima distancia de cada nodo al nodo centro del territorio al que pertenecen. En cambio, en el modelo  $B$  se desea minimizar la distancia total de cada nodo al nodo centro del territorio al que pertenecen, similar al problema de la  $p$ -mediana [15] (uno de los problemas de localización más profundamente estudiados). Es decir, en este modelo (A.1) es reemplazada por la función objetivo (B.1).

*Modelo B*

$$\text{Minimizar} \quad f(x) = \sum_{i \in V} \sum_{j \in V} d_{ij} x_{ij} \quad (\text{B.1})$$

$$\text{Sujeto a:} \quad \sum_{i \in V} x_{ij} = 1 \quad j \in V \quad (\text{B.2})$$

$$\sum_{i \in V} x_{ii} = p \quad (\text{B.3})$$

$$\sum_{j \in V} w_j^a x_{ij} \leq (1 + \tau^a) \mu^a x_{ii} \quad i \in V, a \in A \quad (\text{B.4})$$

$$\sum_{j \in V} w_j^a x_{ij} \geq (1 - \tau^a) \mu^a x_{ii} \quad i \in V, a \in A \quad (\text{B.5})$$

$$\sum_{j \in \cup_{v \in S} N^v \setminus S} x_{ij} - \sum_{j \in S} x_{ij} \geq 1 - |S| \quad i \in V; S \subset V \setminus (N^i \cup \{i\}) \quad (\text{B.6})$$

$$x_{ij} \in \{0, 1\} \quad i, j \in V \quad (\text{B.7})$$

Este modelo fue introducido por Segura-Ramiro et al. [18]. Tal como se procedió con el modelo  $A$ , el modelo  $B_R$  (modelo  $B$  relajado) consiste en relajar de  $B$  las restricciones (B.6).

*Modelo C:* Un tercer modelo consiste en una variación del modelo  $A$  donde se tienen además restricciones de asignación conjunta [8]. Lo anterior, con el fin de cumplir con cierta política de la empresa, la cual consiste en asignar dos unidades básicas a un mismo territorio. Esto beneficia a la empresa ya que si dos puntos de ventas tienen un mismo dueño y están dentro del mismo territorio ahorran tiempo de facturación o cobros. Entonces, en esta variación se tiene también como parámetro conocido un conjunto  $H$  que contienen todas las parejas de nodos que se desea estén en el mismo territorio, definido como sigue

$$H = \{(j, k) \text{ tal que } j \text{ y } k \text{ deben pertenecer al mismo territorio}\}$$

En ese sentido este modelo  $C$  es idéntico al modelo  $A$  con el añadido de la siguiente restricción

$$x_{ij} = x_{ik} \quad i \in V \quad (j, k) \in H$$

## Modelo C

$$\text{Minimizar } f(x) = \max_{i,j \in V} \{d_{ij}x_{ij}\} \quad (\text{C.1})$$

$$\text{Sujeto a: } \sum_{i \in V} x_{ij} = 1 \quad j \in V \quad (\text{C.2})$$

$$\sum_{i \in V} x_{ii} = p \quad (\text{C.3})$$

$$\sum_{j \in V} w_j^a x_{ij} \leq (1 + \tau^a) \mu^a x_{ii} \quad i \in V, a \in A \quad (\text{C.4})$$

$$\sum_{j \in V} w_j^a x_{ij} \geq (1 - \tau^a) \mu^a x_{ii} \quad i \in V, a \in A \quad (\text{C.5})$$

$$\sum_{j \in \cup_{v \in S} N^v \setminus S} x_{ij} - \sum_{j \in S} x_{ij} \geq 1 - |S| \quad i \in V; S \subset V \setminus (N^i \cup \{i\}) \quad (\text{C.6})$$

$$x_{ij} = x_{ik} \quad i \in V, (j, k) \in H \quad (\text{C.7})$$

$$x_{ij} \in \{0, 1\} \quad i, j \in V \quad (\text{C.8})$$

Relajando de este modelo las restricciones de conexidad (C.6) y linealizando la función objetivo, como se hizo en el modelo  $A_R$ , se obtiene el modelo  $C_R$ , a emplear en la experimentación.

*Modelo D:* Este modelo es similar a la formulación del modelo  $A$ , excepto por las restricciones de balance, aquí no se consideran (A.4) y (A.5).

## Modelo D

$$\text{Minimizar } f(x) = \max_{i,j \in V} \{d_{ij}x_{ij}\} \quad (\text{D.1})$$

$$\text{Sujeto a: } \sum_{i \in V} x_{ij} = 1 \quad j \in V \quad (\text{D.2})$$

$$\sum_{i \in V} x_{ii} = p \quad (\text{D.3})$$

$$\sum_{j \in \cup_{v \in S} N^v \setminus S} x_{ij} - \sum_{j \in S} x_{ij} \geq 1 - |S| \quad i \in V; S \subset V \setminus (N^i \cup \{i\}) \quad (\text{D.4})$$

$$x_{ij} \in \{0, 1\} \quad i, j \in V \quad (\text{D.5})$$

Si se linealiza el modelo  $D$  como se hizo con el modelo  $A_R$  y se relajan las restricciones de conexidad (D.4), se obtiene el modelo  $D_R$ , el cual es idéntico al conocido problema del  $p$ -centro [12]. Éste es el cuarto modelo a estudiar en esta

tesis.

Se estudian estos modelos por varias razones. En primera instancia, los modelos mencionados han aparecido anteriormente en aplicaciones importantes y relevantes en casos reales tal como ya se discutió. Como estos modelos han sido abordados desde la perspectiva heurística, el estudio de la relajación de dichos modelos aporta por consiguiente una cota inferior que sirve potencialmente para medir la calidad de las soluciones heurísticas. Por otro lado, es también importante saber el alcance del método exacto en la solución de estos diferentes modelos y a partir de esta información poder justificar el uso de las heurísticas.

# Capítulo 4

## Metodología de Solución

En el presente capítulo se analiza uno de los algoritmos más populares de Programación Entera (PE), el Método de Ramificación y Acotamiento (MRA), y algunos de sus parámetros algorítmicos que permiten tener cierto control sobre la forma de trabajar del método. Se presentan estrategias algorítmicas para el método exacto a fin de mejorar su manera de trabajar (de acuerdo a la estructura del modelo) y también se presentan estrategias para fortalecer uno de los modelos a estudiar.

### 4.1. Método de Ramificación y Acotamiento

El MRA [10] es uno de los métodos más populares en la solución exacta de problemas de programación entera. Esta técnica y algunas variaciones se han aplicado con cierto éxito a diversos problemas de la Investigación de Operaciones (IO), pero es más conocida por sus aplicaciones a los problemas de PE.

El MRA comienza resolviendo la relajación lineal del problema (es decir, sin considerar las restricciones de integralidad) y construye un árbol con soluciones enteras particionando al conjunto de soluciones factibles de modo de descartar soluciones fraccionarias. Sin embargo, este hecho de descomponer puede llevar a un problema intratable por lo que se debe podar el árbol de soluciones de manera inteligente. En seguida, se describe cómo opera el método.

*I.-Inicialización*

Sin perder generalidad, se supone que el problema a resolver es de la forma:

$$(P) \quad \min \quad z = c \cdot x$$

$$\text{sujeto a } A \cdot x \leq b$$

$$x \in \mathbb{Z}^+$$

donde  $c=(c_1, c_2, \dots, c_n)$  es el vector costo,  $x=(x_1, x_2, \dots, x_n)^T$  es el vector de las  $n$  variables de decisión,  $A$  es la matriz de los coeficientes de las  $m$  restricciones,  $b=(b_1, b_2, \dots, b_m)^T$  es el vector de recursos (ó los lados derechos de las restricciones) y  $\mathbb{Z}^+$  es el conjunto de números enteros no negativos.

Se comienza el algoritmo resolviendo el problema relajado, es decir, en el cual no se consideran las restricciones de integralidad (se deja que la variable  $x$  sea una variable libre o real y no entera estrictamente):

*Relajación lineal*

$$(P_0) \quad \min \quad z = c \cdot x$$

$$\text{sujeto a } A \cdot x \leq b$$

$$x \in \mathbb{R}^+$$

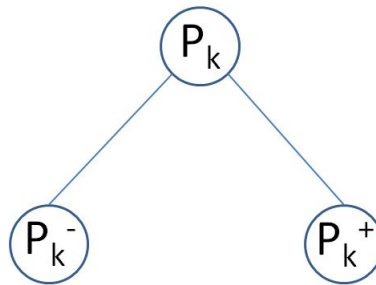
donde  $\mathbb{R}^+$  es el conjunto de los números reales mayores o iguales que cero (no negativos). Si la solución es entera, se finaliza el algoritmo. Si no, se define una cota superior que más adelante servirá como solución de apoyo ( $Z_U$ ).  $Z_U$  representa el valor de la mejor solución factible conocida. Al iniciar, si no se conoce, se toma el valor de  $Z_U = +\infty$ .

Nota: Si el problema es binario, las restricciones de  $x_i \in \{0, 1\}$  al relajar se transforman en  $0 \leq x_i \leq 1$ .

*II.-Ramificación*

Nótese que al llegar a este paso algunas variables ya se fijaron en algún valor

y se toma una variable no fija para ramificar, esto es un subproblema. Supongamos que en algún nodo del árbol se tiene un subproblema ( $P_k$ , ver Figura 4.1) con un conjunto de restricciones y variables  $R_k$  (se entiende que en  $R_k$  ya se incluyen las variables que se han fijado previamente), y solución óptima tal que  $x_i^* = f$  (para algún  $i$ ) con  $f$  un número fraccionario. A este subproblema  $P_k$  también se le conoce como nodo (del árbol de enumeración). Entonces, se ramifica este problema en otros dos subproblemas a los cuales se agrega a cada uno una restricción que le impida tomar el valor  $x_i = f$ , dichos subproblemas denotados por  $P_k^-$  y  $P_k^+$  se presentan a continuación:



**Figura 4.1:** Ilustración del subproblema  $P_k$

$$\begin{array}{ll}
 (P_k^-) \min & z = c \cdot x \\
 \text{sujeto a} & x \in R_k \\
 & x_i \leq \lfloor f \rfloor \\
 (P_k^+) \min & z = c \cdot x \\
 \text{sujeto a} & x \in R_k \\
 & x_i \geq \lfloor f \rfloor + 1
 \end{array}$$

La parte entera de  $\lfloor f \rfloor$  es decir,  $\lfloor f \rfloor$  es el número entero más próximo menor a él en caso de ser  $f$  un número fraccionario, por ejemplo:

$$\lfloor 3.546 \rfloor = 3$$

$$\lfloor 0.543 \rfloor = 0$$

Si el número es entero la parte entera será simplemente él mismo ( $\lfloor 4 \rfloor = 4$ ). A estos dos subproblemas  $P_k^-$  y  $P_k^+$  se les llama hijos de  $P_k$ .

Para seleccionar uno de los subproblemas restantes, nodos que no han sido sondeados ni se les ha hecho una partición, se utiliza alguna *regla de selección de*



*subproblemas* y se le hace una partición en dos o más nuevos subproblemas. Dentro del subproblema, representado por un nodo, que se va a ramificar se puede fijar una regla para seleccionar la variable a ramificar (ver opción `varsel`, Apéndice B). Además se pueden dar prioridades, basadas en el conocimiento del problema, a las variables ya que esto puede reducir dramáticamente el número de subproblemas buscados y por consiguiente reducir el tiempo de solución (ver opción `mipordind`, Apéndice B).

Las dos reglas más populares para seleccionar un subproblema para hacer una partición son:

- *Regla de la mejor cota.* Se elige el subproblema que tiene la cota más favorable (la menor cota inferior en el caso de minimización) porque parecería que este subproblema es el que más promete contener la solución óptima. Esta regla tiende a minimizar el número de iteraciones que requiere el algoritmo.
- *Regla de la cota más reciente.* Se selecciona el subproblema de *creación más reciente* que no haya sido sondeado, rompiendo los empates entre subproblemas creados al mismo tiempo al tomar aquel que tenga la cota más favorable. Las ventajas de esta regla son una contabilidad y registro menos elaborado y una mayor oportunidad de obtener las cotas de manera eficiente. El método seleccionado para obtener las cotas debe representar un buen término medio entre lo ajustado de las cotas y el esfuerzo computacional.

### III.-Acotamiento

Considerar un subproblema no resuelto  $P_i$  y resolver su relajación lineal. Para cada nuevo subproblema, se obtiene una *cota inferior*  $Z_L$  (valor óptimo del problema relajado de  $P_i$ ) sobre el valor de la función objetivo para las soluciones factibles en el subproblema.

### IV.-Sondeo

Aquellos subproblemas cuyas cotas inferiores ( $Z_L$ ) excedan la cota superior actual ( $Z_U$ ) sobre el valor de la función objetivo, se eliminan o podan. Otros subproblemas también se pueden descartar o eliminar si se encuentran que no son de interés, bien porque el subproblema no tiene soluciones factibles (por consiguiente, todos los subproblemas que se generaran a apartir de éste tendrían soluciones infactibles) o porque ya se encontró su mejor solución. Se dice que un subproblema que se elimina por estas u otras razones se considera como ya visto o **sondeado**.

Cada nuevo subproblema se excluye o sondea si:

- *Prueba de sondeo 1.* El problema en el nodo es infactible por lo que todos los subproblemas generados a partir de él serán también infactibles.
- *Prueba de sondeo 2.* El problema en el nodo tiene un valor óptimo  $Z_L$  peor que la mejor solución entera encontrada  $Z_L \geq Z_U$ , esto implica que el óptimo (al problema original) no puede estar contenido en este nodo o cualquiera de sus descendientes.
- *Prueba de sondeo 3.* El problema en el nodo tiene una solución entera. Se ha identificado la mejor solución factible en el subproblema (de manera que  $Z_L$  corresponde al valor de su función objetivo); si ocurre esto y  $Z_L$  es mejor que  $Z_U$ , es decir  $Z_L < Z_U$ , entonces se restablece  $Z_U = Z_L$ , se registra esta solución como la nueva solución de apoyo y se aplica de nuevo la prueba de sondeo 2 a todos los subproblemas restantes (no sondeados).

#### *V.-Optimalidad*

El proceso termina cuando no existen subproblemas restantes (no sondeados) y la solución de apoyo actual  $Z_U$  es la solución óptima, si no existe solución de apoyo, es decir si  $Z_U$  es todavía igual a  $\infty$ , entonces el problema no tiene soluciones factibles. De otra manera, se regresa al paso de ramificación (II).

En caso de querer maximizar la función objetivo en el problema original, el procedimiento es el mismo que el caso en donde se desea minimizar excepto que

se invierten los papeles de las cotas inferior y superior.  $Z_U$  se sustituye por  $Z_L$  y viceversa,  $\infty$  se convierte en  $-\infty$  y las direcciones de las desigualdades también se invierten.

Los pasos de ramificación y acotamiento permiten una gran flexibilidad al diseñar un algoritmo específico para el problema de interés y tienen un efecto importante sobre la eficiencia computacional del algoritmo.

La forma de trabajar del MRA permite tener cierto control sobre algunos de sus parámetros algorítmicos, con el fin de ayudar al método a encontrar resultados que sean óptimos y obtenidos en tiempos razonables. Algunos de estos parámetros son:

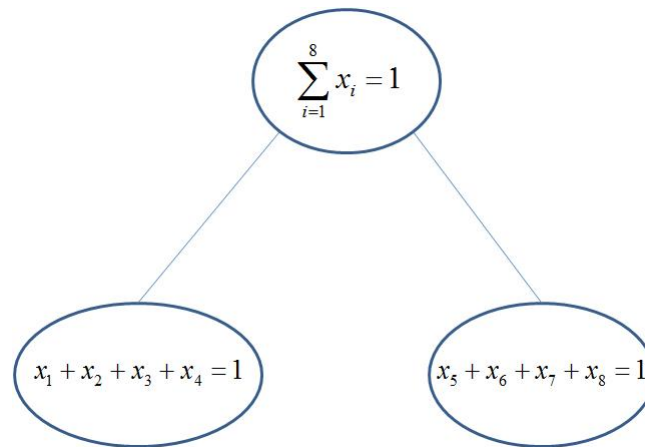
1. *Dar o no prioridad a las variables a ramificar.* Las prioridades se deben asignar basadas en el conocimiento del problema. Las variables con prioridades más altas serán ramificadas antes de las variables con prioridades más bajas. Esta dirección de búsqueda del árbol puede a menudo reducir dramáticamente el número de nodos buscados.
2. *Utilizar o no conjuntos ordenados especiales (SOS, por sus siglas en inglés).* El utilizar estos conjuntos permite ramificar en cada nodo sobre una restricción en lugar de ramificar sobre una variable como normalmente se hace. Por lo cual, se espera que al utilizarlos el problema converja más rápido a la solución deseada.

Por ejemplo, se tiene la restricción  $\sum_{i=1}^8 x_i = 1$  (ver Figura 4.2), lo que pasa si se utilizan los conjuntos ordenados especiales es que en un algún nodo se ramifica sobre una restricción en lugar de una variable, un ejemplo de esto en la restricción anterior sería: la suma de las primeras cuatro variables es igual a 1 ó la suma de las otras variables debe ser igual a 1, de esto se sabe que las demás variables que no suman 1 deben sumar 0, es decir:

a) Si  $x_1 + x_2 + x_3 + x_4 = 1$  entonces  $x_5 + x_6 + x_7 + x_8 = 0$ , lo cual implica que

$$x_5 = x_6 = x_7 = x_8 = 0 \text{ ó}$$

b) Si  $x_5 + x_6 + x_7 + x_8 = 1$  entonces  $x_1 + x_2 + x_3 + x_4 = 0$ , lo cual implica que



**Figura 4.2:** Ejemplo de aplicar los conjuntos ordenados especiales.

$$x_1 = x_2 = x_3 = x_4 = 0$$

Lo anterior ayuda al método de solución a fijar cuatro variables a 0, se espera que haga menos iteraciones y por consiguiente que disminuya el tiempo de ejecución.

3. *La cantidad de sondeos de implicaciones lógicas realizados.* Determina la cantidad de sondeos lógicos realizados sobre un problema de programación entera mixta. El MRA utiliza técnicas de sondeo básicas (probing) previas al desarrollo del algoritmo, las cuales, provisionalmente ponen una variable binaria en uno de sus límites 1 ó 0 y exploran las consecuencias lógicas que se desprenden de hacer esto. Este tipo de mecanismos fueron aportadas por Savelsbergh [17].

Por ejemplo:

Se tiene el siguiente PEM.

Minimizar	$24x_1 + 12x_2 + 16x_3 + 4y_1 + 2y_2 + 3y_3$
Sujeto a	$y_1 + 3y_2 \geq 15$
	$y_1 + 2y_3 \geq 10$
	$2y_1 + y_2 \geq 20$
	$y_1 \leq 15x_1$
	$y_2 \leq 20x_2$

$$y_3 \leq 20x_2$$

$$y_3 \leq 5x_3$$

$$y_1, y_2, y_3 \in \mathfrak{R}_+$$

$$x_1, x_2, x_3 \in \{0, 1\}$$

El uso de las técnicas básicas, previas al desarrollo del algoritmo, pone la cota superior sobre las variables continuas a 15, 20 y 5 respectivamente. Las implicaciones lógicas que pueden ser identificadas en la formulación anterior se presentan en la Tabla 4.1. Este tipo de mecanismos son técnicas de preprocesamiento (probing).

Implicación	Derivación
$y_1 = 0$	$(x_1 = 0 \Rightarrow y_1 = 0)$
$y_2 = 20$	$(x_1 = 0 \Rightarrow y_1 = 0 \Rightarrow y_2 = 20)$
$x_1 = 0 \Rightarrow y_3 = 5$	$(x_1 = 0 \Rightarrow y_1 = 0 \Rightarrow y_3 = 5)$
$x_2 = 1$	$(x_1 = 0 \Rightarrow y_1 = 0 \Rightarrow y_2 = 20 \Rightarrow x_2 = 1)$
$x_3 = 1$	$(x_1 = 0 \Rightarrow y_1 = 0 \Rightarrow y_3 = 5 \Rightarrow x_3 = 1)$
$y_2 = 0$	$(x_2 = 0 \Rightarrow y_2 = 0)$
$x_2 = 0 \Rightarrow y_1 = 15$	$(x_2 = 0 \Rightarrow y_2 = 0 \Rightarrow y_1 = 15)$
$x_1 = 1$	$(x_2 = 0 \Rightarrow y_2 = 0 \Rightarrow y_1 = 15 \Rightarrow x_1 = 1)$
$x_3 = 0 \Rightarrow y_3 = 0$	$(x_3 = 0 \Rightarrow y_3 = 0)$
$x_1 = 1$	$(x_3 = 0 \Rightarrow y_3 = 0 \Rightarrow y_1 \geq 10 \Rightarrow x_1 = 1)$

**Tabla 4.1:** Implicaciones lógicas

Supóngase, sin perder generalidad, que una variable  $x_k$  es provisionalmente una cota superior. Si allí ya existen implicaciones lógicas asociadas a la variable  $x_k$  siendo fijada en su cota superior, se pueden efectuar durante la búsqueda otras consecuencias lógicas. Obsérvese, que el efectuar una implicación lógica puede provocar la realización de varias otras implicaciones lógicas. La cantidad de sondeos lógicos realizados sobre un modelo de PEM se puede variar en base

al conocimiento del problema, éstos sondeos pueden ser tan poderosos como consumir mucho tiempo (ver la opción `probe`, Apéndice B).

Algunas opciones que se pueden considerar para este parámetro son:

- a) No realizar sondeos de implicaciones lógicas.
- b) Que el método decida (lo haga automáticamente).
- c) Realizar sondeo de implicaciones lógicas limitado.

Este último puede causar reducciones o aumentos dramáticos en el tiempo de solución dependiendo del modelo particular.

4. *Regla para seleccionar la variable a ramificar.* Esta opción se utiliza para fijar la regla para seleccionar la variable a ramificar en el nodo que se ha seleccionado. Algunas opciones que se pueden considerar para este parámetro son:

- a) *Que el método seleccione la mejor regla basada en el problema y su progreso.*
- b) *Ramificar sobre variables con infactibilidad mínima.* Esta regla puede conducir más rápidamente a una primera solución factible entera, pero usualmente será lenta para alcanzar la solución óptima entera.
- c) *Ramificación sobre variables con infactibilidad máxima.* Esta regla propicia cambios más grandes en el árbol de búsqueda anterior para alcanzar en tiempos rápidos la solución óptima.
- d) *Ramificación fuerte.* Este ajuste causa la selección de la variable basada en resolver parcialmente un número de subproblemas con ramas opcionales para ver que rama es la más prometedora. Esto es a menudo eficaz en problemas grandes y difíciles.

La elección de estos parámetros puede afectar el comportamiento del MRA, en cuanto a tiempo y calidad de solución encontrada (cuando el algoritmo para antes de llegar a la solución óptima). El valor específico de los mismos depende mucho del problema o de la instancia particular que se desea resolver, por tanto generalmente

no es obvio saber de antemano cuales son los valores más idóneos. Por tal motivo, surge la necesidad de hacer un estudio detallado que permita una eficaz evaluación de dichos parámetros algorítmicos para el modelo en cuestión. Algunos de estos parámetros se describen en el Apéndice B.

El encontrar estos valores más idóneos para los modelos descritos es una de las contribuciones principales de esta tesis.

## 4.2. Fortalecer el Modelo $A_R$

A fin de fortalecer el modelo  $A_R$ , se presentan 3 restricciones válidas que son de interés a investigar en la experimentación, las cuales son:

$$x_{ij} \leq x_{ii} \quad i, j \in V \quad (\text{E.1})$$

$$\sum_{j \in V} x_{ij} \leq nx_{ii} \quad i \in V \quad (\text{E.2})$$

$$\sum_{j \in V} x_{ij} \leq (2n/p)x_{ii} \quad i \in V \quad (\text{E.3})$$

La restricción (E.1) asegura que no se asignen nodos a un nodo  $i$  si éste no es centro (es decir, si  $x_{ii} = 0 \Rightarrow x_{ij} = 0$ ). Ésta condición está presente implícitamente en la cota superior de la restricción de balance (A.15). Note que en la restricción (E.2) es redundante, si  $x_{ii} = 1 \Rightarrow \sum_{j \in V} x_{ij} \leq n$ , esta asegura (en caso de ser centro  $x_{ii}$ ) que no se asignen a éste más nodos de los  $n$  nodos que existen. Ahora bien, en (E.3) se reemplaza el término  $n$  por  $2n/p$ . Al igual que (E.1) y (E.2), esta restricción asegura que si  $x_{ii} = 0 \Rightarrow x_{ij} = 0 \forall j$ . La restricción (E.3) es más fuerte (mejor) que la (E.2). Esto ocasiona que el poliedro o la región factible sea más acotada y, por lo tanto, se espera que la relajación lineal sea mejor y por consiguiente converja más rápido a la solución óptima. Teóricamente con la restricción (E.3) se pueden estar eliminando soluciones factibles del problema. Sin embargo, en el trabajo empírico se ha mostrado que en la práctica la cantidad de nodos que se asignan a cada territorio formado nunca excede a  $2n/p$  (ver Figuras 5.6 y 5.7). Por tanto las restricciones (E.3) son válidas, al menos desde el punto de vista empírico para las instancias probadas.

Estas restricciones son válidas y garantizan una mejor representación del poliedro de soluciones enteras, por lo cual, se espera que al agregarlas dicho poliedro

resulte ser mas acotado y por consiguiente el método converja más rápido a la solución óptima; sin embargo, esto no es tan obvio ya que al agregarlas el modelo se hace más grande.



# Capítulo 5

## Estudio Computacional

Se considera que la experimentación es una disciplina iterativa, para la cual se debe prestar debida atención a la planeación, el diseño y el análisis de los experimentos. Un diseño experimental es una prueba o una serie de pruebas en las cuales se introducen cambios deliberados en las variables de entrada de un proceso o sistema, de manera que sea posible observar e identificar las causas de los cambios en la respuesta de salida. El diseño de experimentos es el proceso de planear un experimento para obtener datos apropiados, que pueden ser analizados mediante métodos estadísticos, con objetivo de producir conclusiones válidas y objetivas. Se requiere un enfoque estadístico del diseño de experimentos para obtener conclusiones significativas a partir de los datos [6].

En este capítulo se presenta el plan que se siguió a fin de realizar la experimentación. Se describe el diseño de experimentos que aporta la información necesaria al trabajo para estudiar los efectos, ya sean positivos o negativos, de las variables o parámetros de interés. Además, se realiza un análisis de los resultados obtenidos para poder seleccionar la mejor información que aporte algún beneficio a los que estudien este problema de diseño territorial más adelante.

## 5.1. Objetivos y Datos de Prueba

Los objetivos de esta evaluación consisten en determinar, en base a un experimento computacional y análisis estadístico, lo siguiente:

1. El efecto que tienen algunos parámetros algorítmicos del método exacto (MRA) sobre el tiempo y la calidad de la solución al resolver los modelos.
2. Si el agregar algunas restricciones válidas tiene algún efecto, ya sea positivo o negativo, sobre el tiempo y la calidad de la solución.
3. Si el variar el parámetro  $p$  (número de territorios) afecta la calidad y el tiempo de solución.
4. Conocer el alcance de los actuales métodos exactos en función del tamaño de las instancias que pueden tratarse con dichas técnicas.

Para los experimentos se generaron aleatoriamente problemas basados en datos reales proporcionados por un socio industrial. Para simular las manzanas geográficas en una ciudad se generaron los  $n$  nodos aleatoriamente en un plano cartesiano. Las coordenadas  $x$  y  $y$  que definen la ubicación del nodo son generadas con una distribución uniforme dentro de un plano de  $[500 \times 500]$ . Las dos actividades nodales fueron generadas a partir de una distribución uniforme en los rangos  $[0,3]$  y  $[1,12]$  para el número de clientes y la demanda, respectivamente. Además, dos diferentes tipos de instancias de acuerdo al parámetro  $\tau^a$  son consideradas ( $DS10$  y  $DS05$ ). Entonces, para cada modelo se tienen instancias de varios tipos y tamaños a considerar en la experimentación.

### Tipos de instancias

- $DS10$ : Son los problemas para los cuales el parámetro de tolerancia de las restricciones de balance (A.4) y (A.5)  $\tau^a = 0.10$  (es decir, se permite una desviación de un 10%).
- $DS05$ : Son los problemas para los cuales el parámetro de tolerancia de las restricciones de balance (A.4) y (A.5)  $\tau^a = 0.05$  (es decir, se permite una desviación de un 5%).

Tamaño de instancias

- (60,4) Problemas con 60 nodos y 4 territorios.
- (100,5) Problemas con 100 nodos y 5 territorios.

Una solución se considera óptima en los experimentos a realizar si dicha solución satisface el criterio de optimalidad relativa de 0.01 (COR, ver la opción `epgap`, Apéndice B), es decir, cuando se garantiza que la solución obtenida se encuentra a lo mucho a un 1% del valor óptimo. El intervalo de optimalidad relativa (IOR) se calcula de la siguiente manera:  $\frac{Z_U - Z_L}{Z_L}$  donde  $Z_U$  es la cota superior (solución factible de PEM) y  $Z_L$  es la cota inferior (mejor posible encontrada). Si se desea el porcentaje de este intervalo sólo se multiplica por 100.

En la experimentación, se hace uso de GAMS [7], el cual es un modelador algebraico para problemas de optimización con interfaz a varios métodos de solución de uso comercial. En nuestro caso, los problemas de PL y PEM se resuelven con el método GAMS/CPLEX, es decir, el método de optimización llamado por GAMS es CPLEX [7] que es hoy en día uno de los métodos más potentes a nivel mundial en la resolución de problemas de PL y PEM. Los experimentos fueron realizados en un ordenador SunFire V440 con el sistema operativo Solaris 9 propiedad del Laboratorio de Cómputo de Alto Desempeño del Programa de Posgrado en Ingeniería de Sistemas de la FIME, UANL. Se utilizó la versión 9.0 de CPLEX en cada experimento.

## 5.2. Diseño de Experimentos

La primera experimentación (experimentos A, B, C y D) consiste en realizar ejecuciones para 3 modelos ( $A_R$ ,  $B_R$ ,  $C_R$ ), variando el tamaño y el tipo de problema. Además se modifican algunos parámetros algorítmicos del método exacto, los cuales son:

1. Dar o no prioridades a las variables a ramificar.
2. Utilizar o no conjuntos ordenados especiales (SOS, por sus siglas en inglés).

3. La cantidad de sondeos de implicaciones lógicas realizados.
4. La regla para seleccionar la variable de ramificación.

Al determinar el mejor valor para algún parámetro dado, éste se fija en los experimentos restantes en cada modelo.

En la segunda experimentación (experimento E) se realizan corridas para el modelo  $A_R$ , variando el tamaño, el tipo de problema y agregando una a una las tres restricciones mencionadas en la Sección 4.2 (E.1, E.2 y E.3), con el fin de fortalecer el modelo.

En la tercera experimentación (experimento F), se analiza la complejidad del problema al variar el número de territorios ( $p$ ), es decir se desea saber si el tiempo de ejecución aumenta conforme se modifica  $p$ . Si ocurre lo anterior, se dice que el problema se vuelve más difícil. Además, se espera que al aumentar el número de territorios disminuya el valor de la función objetivo, debido a que habrá más centros y los nodos que se asignen a éstos estarán más cercanos. Para esto, se realizan corridas probando cada uno de los modelos ( $A_R$ ,  $B_R$ ,  $C_R$ ) con los parámetros que proporcionaron la mejor calidad y tiempo de solución en la primera experimentación. Esto sólo se prueba para los problemas de tipo DS05 de tamaño (60,4) y (100,5). Los valores que se prueban para  $p$  son: para los problemas con 60 nodos,  $p = 2, 3, 4, 5$ ; para los problemas con 100 nodos,  $p = 2, 3, 4, 5, 6, 7$

En la cuarta experimentación (experimento G), se mide el alcance de los actuales métodos exactos en función del tamaño de las instancias que pueden tratarse con dichas técnicas. Se consideran instancias con hasta 2000 nodos para algunos de los modelos, a fin de determinar el tamaño máximo que se puede resolver en cada modelo ( $A_R$ ,  $B_R$ ,  $C_R$ ,  $D_R$ ).

En todas estas pruebas se mide el tiempo que tarda el algoritmo en dar una solución óptima. Para analizar los resultados obtenidos en cada experimento se emplea una prueba no paramétrica llamada Kruskal-Wallis (conocida también como

la prueba H), debido a que los datos no cumplen con las condiciones necesarias para aplicar un ANOVA (análisis de varianza), entre ellas que las respuestas no están normalmente distribuidas. La prueba de Kruskal-Wallis (véase Freund, Miller y Miller [6]) es una de las pruebas no paramétricas más poderosas y se aplica cuando se tiene un diseño completamente al azar pero no se puede suponer normalidad. Sirve para determinar la igualdad o diferencia entre tratamientos. En caso de que en alguno de los experimentos se determine que no hay evidencia para decir que los tratamientos son iguales (a partir de la prueba de Kruskal-Wallis) y éstos sean más de dos, entonces se considera una prueba estadística no paramétrica llamada *Mann-Whitney* [6] (también conocida como la prueba U o prueba de Wilcoxon). Se toman los tratamientos de dos en dos y se aplica esta prueba.

Se utilizó el software MINITAB 14 [14] como herramienta estadística para analizar los datos y de esta manera poder establecer conclusiones con validez estadística. Los tiempos de cómputo obtenidos en los experimentos se introducen en MINITAB, se selecciona la prueba Kruskal-Wallis y se comienza el análisis. Esta prueba no paramétrica cuenta con las siguientes hipótesis:

$H_0$ : Todas las poblaciones tienen medias iguales ( $\mu_1 = \mu_2 = \dots = \mu_k$ ), es decir los tratamientos son iguales.

$H_1$ : No todas las medias poblacionales son iguales, es decir no todos los tratamietnos son iguales.

Con esto se prueba la hipótesis nula ( $H_0$ ) de que se estan muestreando  $k$  poblaciones continuas idénticas contra la alternativa de que al menos una población tiene media diferente a las demás [6].

Esta herramienta estadística calcula las medias de los  $k$  tratamientos y un valor  $q$ , con el cual determina si se rechaza o no la hipótesis nula. Si el valor de  $q < \beta$  (donde  $\beta = 0.05$  es el nivel de significancia) se rechaza  $H_0$  y se toma la alternativa, en otro caso se acepta  $H_0$ .

En caso de que en algún experimento (de más de dos niveles) la prueba de Kruskal-Wallis de como respuesta que no hay evidencia para decir que los tratamietnos son iguales, se aplica a estos niveles la prueba de Mann-Whitney (de dos en dos).

Esta prueba no paramétrica cuenta con las siguientes hipótesis:

$H_0$ : Las poblaciones tienen medias iguales ( $\mu_1 = \mu_2$ ), es decir los tratamientos son iguales.

$H_1$ : Las poblaciones tienen medias diferentes ( $\mu_1 \neq \mu_2$ ), es decir no todos los tratamientos son iguales.

$H_1$ :  $\mu_1 > \mu_2$ , es decir la media poblacional de un primer tratamiento es mayor que la media poblacional para otro tratamiento.

$H_1$ :  $\mu_1 < \mu_2$ , es decir la media poblacional de un primer tratamiento es menor que la media poblacional para otro tratamiento.

MINITAB calcula la medias de los dos tratamientos y un valor  $r$ , con el cual determina si se rechaza o no la hipótesis nula. Si el valor de  $r < \rho$  (donde  $\rho = 0.05$  es el nivel de significancia) se rechaza  $H_0$  y se toma la alternativa que se está probando, en otro caso se acepta  $H_0$ .

## 5.3. Resultados

### Experimento A

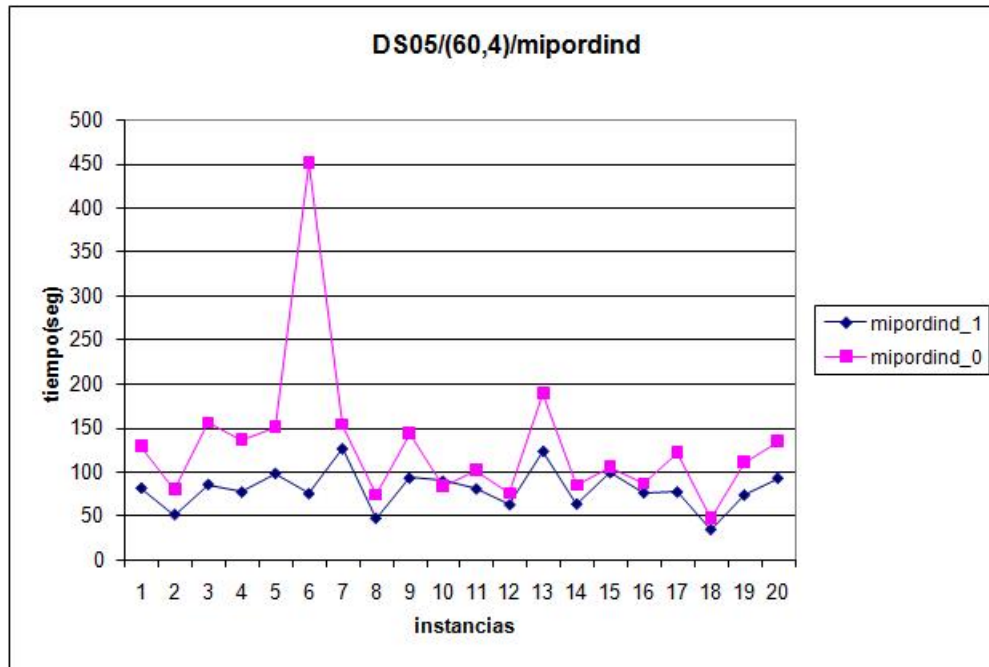
En esta parte, se estudia la sensibilidad del algoritmo del método exacto al dar o no prioridades a las variables a ramificar. Esto se mide en base a si mejora el tiempo de solución. Para esto, se considera dar mayor prioridad para ramificar primeramente sobre las variables  $x_{ii}$ , después de estas variables se consideran las  $x_{ij}$ . Esto se realiza para el modelo  $A_R$  con 20 instancias para cada tipo de problema y tamaño, en total para este experimento son 80 instancias a dos niveles (mipordind\_0 y mipordind\_1) es decir, se realizan 160 ejecuciones. Los resultados se muestran en la Tabla 5.1 y Figuras 5.1, 5.2. La primera columna en la Tabla 5.1 indica de que tipo son las instancias que se están probando, la segunda el tamaño de éstas, la tercera y cuarta columna indican los tiempos de cómputo promedio que el MRA empleó al ejecutar las 20 instancias sin prioridades y con prioridades, respectivamente. Todas las instancias se resolvieron de manera óptima.

**mipordind\_0**= Dar prioridades a las variables  $x_{ii}$  sobre las  $x_{ij}$ .

**mipordind\_1**= No dar ninguna prioridad.

Tipo	$(n, p)$	mipordind_0	mipordind_1
DS05	(60,4)	130.0	79.7
DS10	(60,4)	79.4	51.6
DS05	(100,5)	3125.5	895.5
DS10	(100,5)	3783.5	794.1

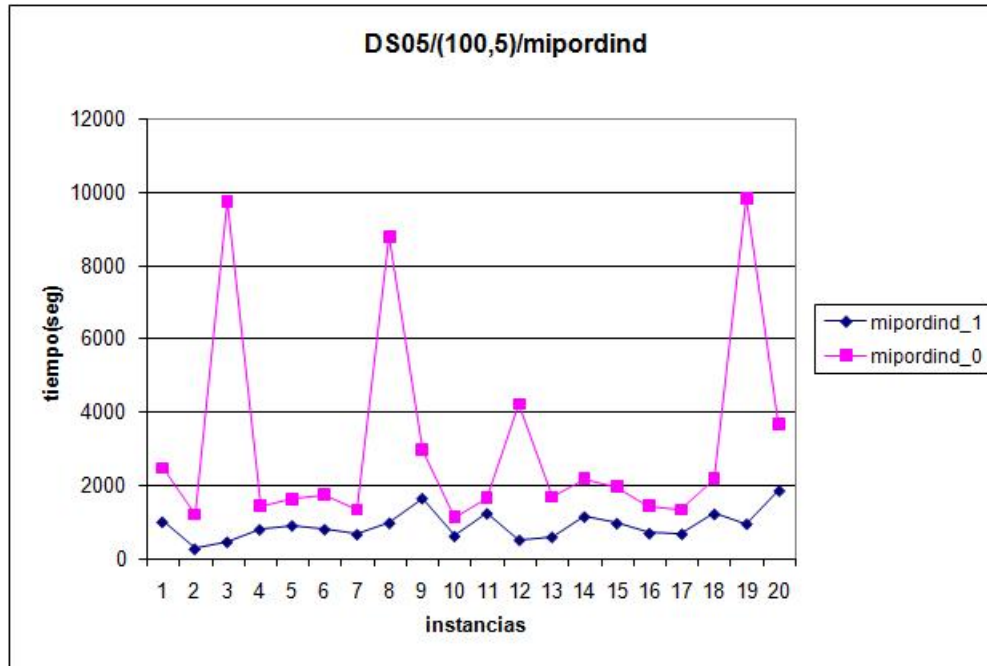
**Tabla 5.1:** Tiempo de cómputo promedio, variando las prioridades para el modelo  $A_R$



**Figura 5.1:** Utilizar o no las prioridades en instancias de DS05/(60,4)

El valor  $q$  obtenido en la prueba estadística para las diferentes instancias del modelo  $A_R$  es de 0 el cual es menor que  $\beta$ . Entonces, se puede decir que se rechaza  $H_0$ , es decir no hay evidencia para decir que los tratamientos son iguales. Se puede observar, además, en la Tabla 5.1 y Figuras 5.1 y 5.2 que los tiempos de solución disminuyen considerablemente al darle prioridad de ramificar a las variables  $x_{ii}$  sobre las  $x_{ij}$ . El usar prioridades reduce el tiempo de solución significativamente, por lo cual, se considera importante este parámetro y se deja fijo para las siguientes

experimentaciones. Se considera activar el orden de prioridades para los otros tres modelos, debido a que presentan una estructura muy similar a este modelo. De esta manera, tanto en este modelo  $A_R$  como en los demás ( $B_R$ ,  $C_R$  y  $D_R$ ) se considera dar prioridades a las variables  $x_{ii}$  sobre las  $x_{ij}$ .



**Figura 5.2:** Utilizar o no las prioridades, en instancias de DS05/(100,5)

## Experimento B

En esta parte se desea determinar si el MRA se ve afectado al utilizar conjuntos ordenados especiales. Para esto, se realiza el experimento con 80 instancias a dos niveles (SOS\_0 y SOS\_1) para los modelos  $A_R$ ,  $B_R$ , y  $C_R$ , es decir 480 ejecuciones. Los resultados se muestran en las Tablas 5.2, 5.3 y 5.4 y en la Figura 5.3. En la primera columna de las Tablas 5.2, 5.3 y 5.4 se muestra el tipo de instancias que se están experimentando, en la segunda el tamaño de éstas, en la tercera y cuarta columna se presentan los tiempos de cómputo promedio que el MRA utilizó al resolver las 20 instancias sin utilizar conjuntos ordenados especiales y utilizando éstos, respectivamente. El criterio de parada para cada uno de estos experimentos fue de 2,000 segundos.



**SOS\_0** = No utilizar conjuntos ordenados especiales en el modelo.

**SOS\_1** = Utilizar conjuntos ordenados especiales en el modelo.

Tipo	$(n, p)$	SOS_0	SOS_1
DS05	(60,4)	79.7	232.2
DS10	(60,4)	57.0	127.1
DS05	(100,5)	895.5	(*)2000.0
DS10	(100,5)	794.1	(*)1874.1
(*)Soluciones que se quedaron entre el 85 y 179 % del valor óptimo (en promedio).			

**Tabla 5.2:** Tiempo de cómputo promedio, utilizando o no conjuntos ordenados especiales para  $A_R$

Tipo	$(n, p)$	SOS_0	SOS_1
DS05	(60,4)	3.8	4.3
DS10	(60,4)	4.6	11.8
DS05	(100,5)	44.9	90.5
DS10	(100,5)	33.3	57.0

**Tabla 5.3:** Tiempo de cómputo promedio, utilizando o no conjuntos ordenados especiales para  $B_R$

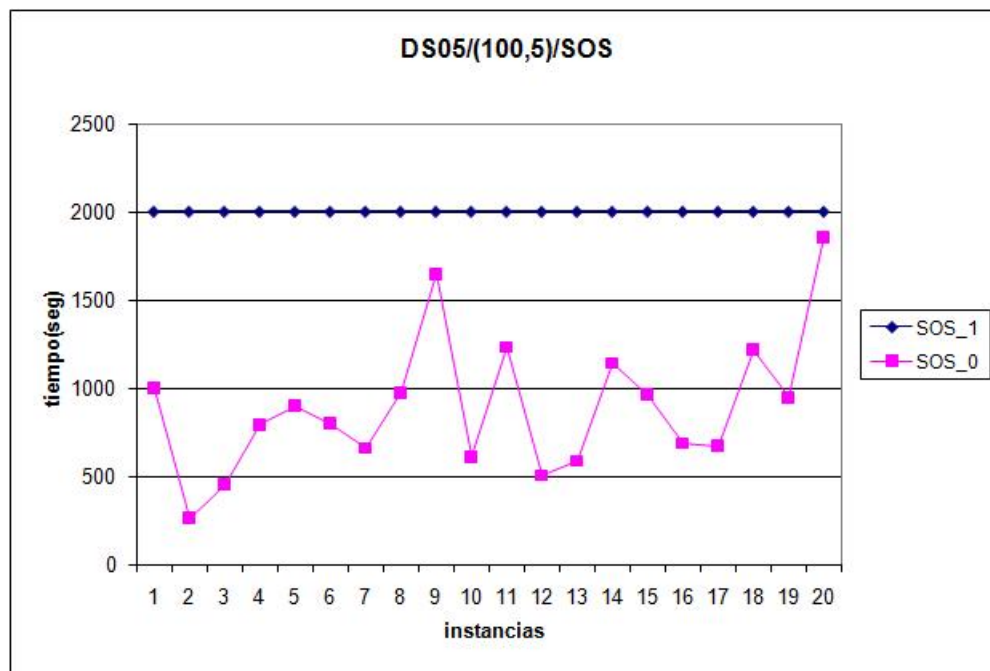
Si se utilizan los conjuntos ordenados especiales en las instancias de tipo DS05 y tamaño (100,5) en el modelo  $A_R$ , ninguna de las 20 instancias llega al valor óptimo. Para las de tipo DS10 del mismo tamaño, en solo 2 de 20 instancias se llega al valor óptimo (ver Tabla 5.2). Algo similar ocurre para las instancias de tipo DS05 y DS10, de tamaño (100,5) en modelo  $C_R$ , al utilizar los conjuntos ordenados especiales ninguna de las 20 instancias llega al valor óptimo (ver Tabla 5.4).

Con la prueba no paramétrica se determina que el valor de  $q$  es igual a 0 para

Tipo	$(n, p)$	SOS_0	SOS_1
DS05	(60,4)	69.7	408.1
DS10	(60,4)	57.4	248.2
DS05	(100,5)	720.8	(*)2000.0
DS10	(100,5)	745.9	(*)2000.0

(\*)Soluciones que se quedaron entre el 85 y 179 % del valor óptimo (en promedio).

**Tabla 5.4:** Tiempo de cómputo promedio, utilizando o no conjuntos ordenados especiales para  $C_R$



**Figura 5.3:** Usar conjuntos ordenados especiales o no, en instancias DS05/(100,5), modelo  $A_R$

las diferentes instancias de los modelos  $A_R$  y  $C_R$ , es decir que la hipótesis  $H_0$  se rechaza. Para estos modelos las diferencias entre usar y no usar SOS son significativas y la opción que propociona mejores tiempos (ver Tabla 5.2 y 5.4) es la de no usar los conjuntos ordenados especiales. En cambio, para las instancias del modelo  $B_R$  el valor de  $q$  es mayor a  $\beta$ , es decir las diferencias son muy pequeñas (no significativas), la prueba determina que da lo mismo cualquier opción que se seleccione (los

tratamientos son iguales), sin embargo se considera el no usar estos conjuntos en base a los tiempos promedios obtenidos (ver Tabla 5.3). Se fija el parámetro de no usar estos conjuntos ordenados especiales para los siguientes experimentos.

Esto también se puede observar en la Figura 5.3. En los tres modelos el utilizar conjuntos ordenados especiales afectó negativamente el tiempo y la calidad de las soluciones sobre todo en las instancias de 100 nodos para los modelos  $A_R$  y  $C_R$ .

## Experimento C

En esta sección del trabajo se estudia la sensibilidad del MRA al realizar o no sondeos de implicaciones lógicas. Para realizar esto, se consideran 80 instancias de los diferentes tamaños y tipos de problemas para el modelo  $A_R$ , este parámetro se mide a tres niveles (probe\_-1, probe\_0 y probe\_1), es decir, se tienen 240 corridas por realizar. Los resultados se muestran en la Tabla 5.5 y Figura 5.4. En la primera columna de la Tabla 5.5 se muestra el tipo de instancias que se están probando en los experimentos, en la segunda, el tamaño de éstas, en la tercera, cuarta y quinta columna se presentan los tiempos de cómputo promedio utilizado por el MRA al resolver las 20 instancias en los tres niveles probe\_-1, probe\_0 y probe\_1, respectivamente.

**probe\_-1** = no realizar sondeos de implicaciones lógicas.

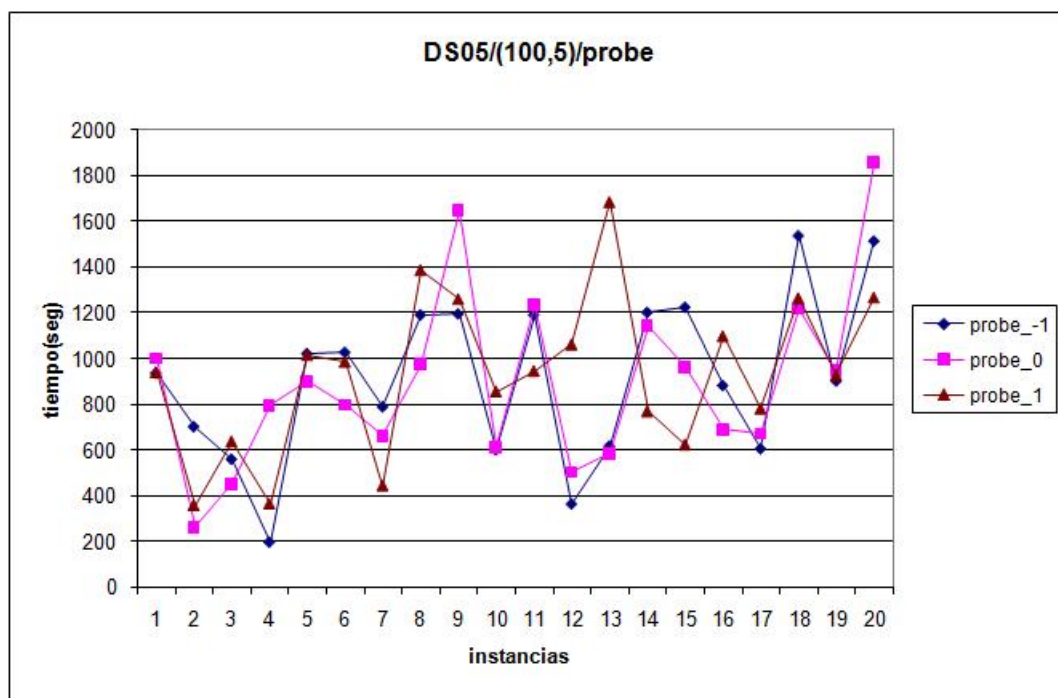
**probe\_0** = automáticamente, en base al progreso del problema.

**probe\_1** = sondeo de implicaciones lógicas limitado.

El valor de  $q$  determinado con la prueba no paramétrica para las instancias del modelo  $A_R$  fue mayor que  $\beta$ , es decir no se rechaza la hipótesis  $H_0$ . En otras palabras, la prueba no proporciona información para decir que los tratamientos son diferentes. Por tanto, se considera que los tratamientos no son diferentes. Debido a que la prueba no fue significativa, se tomó la decisión de considerar los tiempos promedios más pequeños para seleccionar el mejor nivel (ver Tabla 5.5). Se determinó que dejar que automáticamente se realicen los sondeos de implicaciones lógicas (probe\_0) resuelve el problema en menos tiempo que las otras dos opciones, por tal motivo se considera

Tipo	$(n, p)$	probe_-1	probe_0	probe_1
DS05	(60,4)	80.05	79.79	89.73
DS10	(60,4)	60.52	51.67	72.53
DS05	(100,5)	913.35	895.55	934.46
DS10	(100,5)	1302.98	794.12	839.98

**Tabla 5.5:** Tiempo de cómputo promedio, realizando o no sondeos de implicaciones lógicas en el modelo  $A_R$ .



**Figura 5.4:** Evaluación de sondeos de implicaciones lógicas, en DS05/(100,5) del modelo  $A_R$ .

esta opción fija para los siguientes experimentos. Debido a que la estructura de los modelos es similar, es de esperarse que el efecto que tuvo la manera de seleccionar los sondeos lógicos en el modelo  $A_R$  sea similar para los otros tres modelos. Por tal motivo, se fija este valor (probe\_0) en los siguientes experimentos para los cuatro modelos en estudio. Se puede apreciar en la Tabla 5.5 y Figura 5.4 que los tiempos promedios de solución para el modelo  $A_R$  son muy similares.

## Experimento D

En esta parte se desea analizar la sensibilidad del método exacto (MRA) con respecto a la regla de selección de la variable a ramificar. Para esto, se realizan pruebas en 20 instancias de cada tipo y tamaño para los tres modelos ( $A_R$ ,  $B_R$ ,  $C_R$ ), este parámetro se mide a cuatro niveles (`varsel_-1`, `varsel_0`, `varsel_1`, `varsel_3`), es decir, se tiene un total de 960 corridas. La descripción más detallada de cada uno de estos niveles (tratamientos) se encuentra en `varsel`, Apéndice B. Los resultados se muestran en las Tablas 5.6, 5.7, 5.8 y Figura 5.5. En la primera columna de estas tablas se muestra el tipo de instancias que se está probado en cada experimento, en la segunda, el tamaño de éstas, en la tercera, cuarta, quinta y sexta columna se presenta el tiempo de cómputo promedio utilizado por el MRA al resolver las 20 instancias en los niveles `varsel_-1`, `varsel_0`, `varsel_1`, `varsel_3`, respectivamente. El criterio de parada en cada uno de los experimentos es de 2,000 segundos. En la primera columna de las Tablas 5.9, 5.10, 5.11, 5.12, 5.13, 5.14 y 5.15 se muestra la Hipótesis alternativa ( $H_1$ ) presente en la prueba que se realiza de dos en dos (con la prueba de Mann-Whitney) en caso de que no haya evidencia para decir que son iguales los tratamientos y en la segunda columna se escribe Sí o No dependiendo de si se acepta o rechaza la Hipótesis nula ( $H_0: \mu_1 = \mu_2$ ).

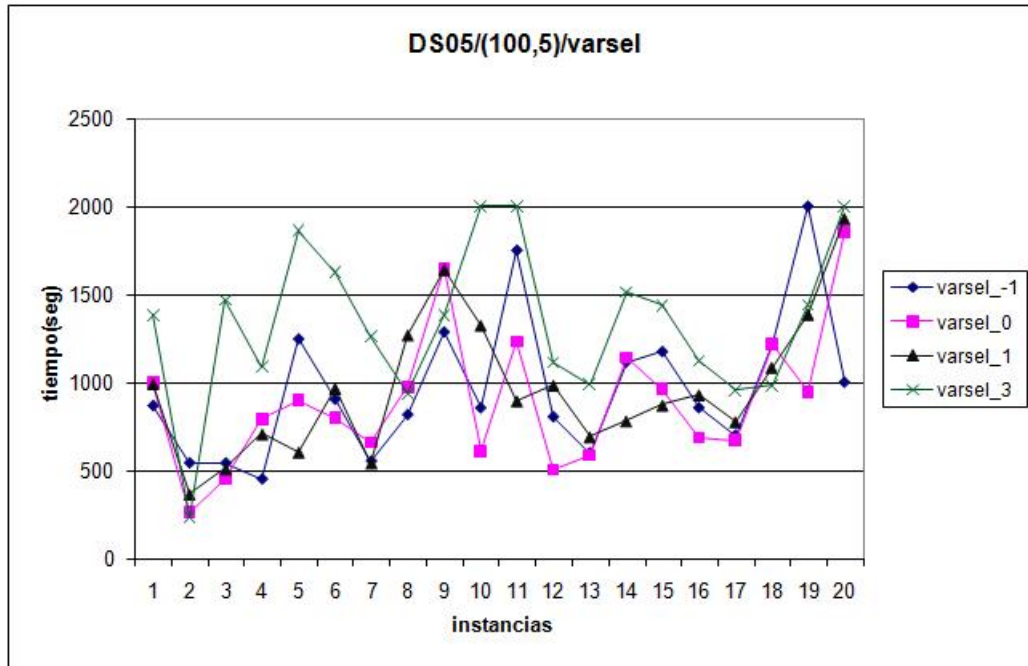
**varsel\_-1** = Ramificar sobre variables con infactibilidad mínima.

**varsel\_0** = Ramificar variable automáticamente seleccionada.

**varsel\_1** = Ramificar sobre variables con infactibilidad máxima.

**varsel\_3** = Ramificación fuerte.

La mayoría de las instancias en el modelo  $A_R$  (mostradas en la Tabla 5.6) se resolvieron de manera óptima, excepto 1 de tipo DS05 y tamaño (100,5) en el nivel `varsel_-1`, 3 de las 20 instancias con estas mismas características en el nivel `varsel_3` y una de las 20 instancias de tipo DS10, tamaño (100,5) y nivel `varsel_1`. En el modelo  $B_R$  todas las instancias se resolvieron de manera óptima. La mayoría de las instancias en el modelo  $C_R$  (mostradas en la Tabla 5.8) se resolvieron de manera óptima. Excepto 6 de las 20 instancias de tipo DS05 y tamaño (100,5) en el nivel `varsel_-1` y



**Figura 5.5:** Regla para seleccionar la variable a ramificar, en instancias DS05/(100,5) en el modelo  $A_R$

Tipo	$(n, p)$	vargsel_1	vargsel_0	vargsel_1	vargsel_3
DS05	(60,4)	71.96	81.96	66.37	106.23
DS10	(60,4)	54.48	57.05	47.13	62.95
DS05	(100,5)	963.36	895.55	961.50	1338.41
DS10	(100,5)	902.13	794.12	938.22	1078.86

**Tabla 5.6:** Tiempo de cómputo promedio, regla para seleccionar la variable a ramificar en el modelo  $A_R$ .

una de estas mismas características en el nivel varsel\_0. También, para 8 instancias de tipo DS10 y tamaño (100,5) en el nivel varsel\_1 y otras dos de estas mismas características en el nivel varsel\_0 y varsel\_1, no se llegó al óptimo.

Al aplicar la prueba estadística a las instancias de 60 nodos del modelo  $A_R$  se obtuvo el valor de  $q=0.010$  para las DS05 y  $q=0.563$  para las DS10. Se rechaza  $H_0$ , es decir, no hay evidencia para decir que los tratamientos son iguales para las instancias DS05. En los tratamientos para las instancias DS10 no se rechaza  $H_0$ , es

Tipo	$(n, p)$	varsel_-1	varsel_0	varsel_1	varsel_3
DS05	(60,4)	3.95	3.88	3.83	4.39
DS10	(60,4)	5.24	4.69	4.81	6.38
DS05	(100,5)	41.32	44.96	48.44	82.93
DS10	(100,5)	34.14	33.38	33.41	46.50

**Tabla 5.7:** Tiempo de cómputo promedio, regla para seleccionar la variable a ramificar en el modelo  $B_R$ .

Tipo	$(n, p)$	varsel_-1	varsel_0	varsel_1	varsel_3
DS05	(60,4)	62.86	69.76	50.19	92.91
DS10	(60,4)	47.62	57.41	39.32	73.37
DS05	(100,5)	(*)1308.71	670.81	698.88	934.64
DS10	(100,5)	(*)1368.65	726.99	946.98	882.66
(*)Soluciones que se quedaron a un 20 % del valor óptimo (en promedio).					

**Tabla 5.8:** Tiempo de cómputo promedio, regla para seleccionar la variable a ramificar en el modelo  $C_R$ .

decir la prueba no proporciona información para decir que no son diferentes, por tanto consideraré que los tratamientos son iguales. Debido a que no hay evidencia para decir que los tratamientos para las DS05 son iguales, se analizan de dos en dos cada uno de estos niveles por medio de la prueba de Mann-Whitney (ver Tabla 5.9).

De las pruebas estadísticas y analizando los tiempos presentados en la Tabla 5.6 se concluye que la mejor opción para las instancias de 60 nodos del modelo  $A_R$  es ramificar sobre variables con infactibilidad máxima (varsel\_1). Para el mismo modelo pero ahora para las instancias de 100 nodos se encontró al aplicar la prueba estadística que para los problemas de tipo DS05 y DS10 el valor de  $q$  era menor a  $\beta$ , es decir no hay evidencia para decir que los tratamientos son iguales. Se procedió de manera similar al caso anterior y se analizaron estos tratamientos de dos en dos con

Hipótesis alternativa $H_1$	Se rechaza $H_0$
$\text{varsel\_1} \neq \text{varsel\_0}$	No
$\text{varsel\_1} \neq \text{varsel\_1}$	No
$\text{varsel\_1} \neq \text{varsel\_3}$	Sí
$\text{varsel\_1} < \text{varsel\_3}$	Sí
$\text{varsel\_0} \neq \text{varsel\_1}$	Sí
$\text{varsel\_0} > \text{varsel\_1}$	Sí
$\text{varsel\_0} \neq \text{varsel\_3}$	No
$\text{varsel\_1} \neq \text{varsel\_3}$	Sí
$\text{varsel\_1} < \text{varsel\_3}$	Sí

**Tabla 5.9:** Hipótesis alternativa para las instancias DS05/(60,4) del Modelo  $A_R$ .

Hipótesis alternativa $H_1$	Se rechaza $H_0$
$\text{varsel\_1} \neq \text{varsel\_0}$	No
$\text{varsel\_1} \neq \text{varsel\_1}$	No
$\text{varsel\_1} \neq \text{varsel\_3}$	Sí
$\text{varsel\_1} < \text{varsel\_3}$	Sí
$\text{varsel\_0} \neq \text{varsel\_1}$	No
$\text{varsel\_0} \neq \text{varsel\_3}$	Sí
$\text{varsel\_0} < \text{varsel\_3}$	Sí
$\text{varsel\_1} \neq \text{varsel\_3}$	Sí
$\text{varsel\_1} < \text{varsel\_3}$	Sí

**Tabla 5.10:** Hipótesis alternativa para las instancias DS05/(100,5) del Modelo  $A_R$ .

la prueba de Mann-Whitney (ver Tabla 5.10). En base a las pruebas estadísticas y a la Tabla 5.6 se concluye que para las DS05 es mejor que el algoritmo ramifique la variable automáticamente seleccionada ( $\text{varsel\_0}$ ). Para las instancias de tipo DS10 se procedió de manera similar al caso de las instancias DS05/(100,5) (ver Tabla 5.11), en base a esta prueba y a los tiempos promedios obtenidos se encontró que al



Hipótesis alternativa $H_1$	Se rechaza $H_0$
$\text{varsel\_1} \neq \text{varsel\_0}$	No
$\text{varsel\_1} \neq \text{varsel\_1}$	No
$\text{varsel\_1} \neq \text{varsel\_3}$	No
$\text{varsel\_0} \neq \text{varsel\_1}$	No
$\text{varsel\_0} \neq \text{varsel\_3}$	Sí
$\text{varsel\_0} < \text{varsel\_3}$	Sí
$\text{varsel\_1} \neq \text{varsel\_3}$	Sí
$\text{varsel\_1} < \text{varsel\_3}$	Sí

**Tabla 5.11:** Hipótesis alternativa para las instancias DS10/(100,5) del Modelo  $A_R$ .

seleccionar la variable con infactibilidad máxima ( $\text{varsel\_1}$ ) se obtuvieron los mejores tiempos. Se fijará para las instancias de 60 y 100 nodos la regla de ramificar la variable con infactibilidad máxima ( $\text{varsel\_1}$ ) ya que las diferencias entre el  $\text{varsel\_0}$  y el  $\text{varsel\_1}$  fueron muy pequeñas (diferencias no significativas).

Por medio de la prueba no paramétrica se encontró que el valor de  $q$  para las diferentes instancias del modelo  $B_R$  eran mayores a  $\beta$ . No se rechaza  $H_0$ , es decir la prueba no proporciona información para decir que los tratamientos son diferentes, por tanto considero que los tratamientos son iguales. Se optó por elegir la opción de ramificar la variable automáticamente seleccionada ( $\text{varsel\_0}$ ), ya que en la mayoría de los problemas este valor del parámetro regresó tiempos menores (ver Tabla 5.7).

Para el modelo  $C_R$ , la prueba estadística presentó resultados similares a los del modelo  $A_R$ , el valor de  $q$  fue menor a  $\beta$  para los diferentes tipos de instancias, es decir no hay evidencia para decir que los tratamientos son iguales. Después de aplicar la prueba estadística de Mann-Whitney a los tratamientos de dos en dos (ver Tablas 5.12, 5.13, 5.14 y 5.15) y ver los datos en la Tabla 5.8, se encuentra que para las instancias de 60 nodos es mejor ramificar sobre variables con infactibilidad máxima ( $\text{varsel\_1}$ ). Para las instancias de 100 nodos se encuentra que es mejor ramificar la

variable automáticamente seleccionada (varsel\_0).

Hipótesis alternativa $H_1$	Se rechaza $H_0$
$\text{varsel\_1} \neq \text{varsel\_0}$	No
$\text{varsel\_1} \neq \text{varsel\_1}$	No
$\text{varsel\_1} \neq \text{varsel\_3}$	Sí
$\text{varsel\_1} < \text{varsel\_3}$	Sí
$\text{varsel\_0} \neq \text{varsel\_1}$	No
$\text{varsel\_0} \neq \text{varsel\_3}$	No
$\text{varsel\_1} \neq \text{varsel\_3}$	Sí
$\text{varsel\_1} < \text{varsel\_3}$	Sí

**Tabla 5.12:** Hipótesis alternativa para las instancias DS05/(60,4) del Modelo  $C_R$ .

Hipótesis alternativa $H_1$	Se rechaza $H_0$
$\text{varsel\_1} \neq \text{varsel\_0}$	No
$\text{varsel\_1} \neq \text{varsel\_1}$	No
$\text{varsel\_1} \neq \text{varsel\_3}$	Sí
$\text{varsel\_1} < \text{varsel\_3}$	Sí
$\text{varsel\_0} \neq \text{varsel\_1}$	Sí
$\text{varsel\_0} > \text{varsel\_1}$	Sí
$\text{varsel\_0} \neq \text{varsel\_3}$	No
$\text{varsel\_1} \neq \text{varsel\_3}$	Sí
$\text{varsel\_1} < \text{varsel\_3}$	Sí

**Tabla 5.13:** Hipótesis alternativa para las instancias DS10/(60,4) del Modelo  $C_R$ .

En la Figura 5.5 se observa que los tiempos son muy similares para dos de las cuatro diferentes maneras de seleccionar la variable a ramificar. En las Tablas 5.6, 5.7 y 5.8 se muestra que la diferencia entre los tiempos de cómputo promedio es

Hipótesis alternativa $H_1$	Se rechaza $H_0$
$\text{varsel\_1} \neq \text{varsel\_0}$	Sí
$\text{varsel\_1} > \text{varsel\_0}$	Sí
$\text{varsel\_1} \neq \text{varsel\_1}$	Sí
$\text{varsel\_1} > \text{varsel\_1}$	Sí
$\text{varsel\_1} \neq \text{varsel\_3}$	No
$\text{varsel\_0} \neq \text{varsel\_1}$	No
$\text{varsel\_0} \neq \text{varsel\_3}$	Sí
$\text{varsel\_0} < \text{varsel\_3}$	Sí
$\text{varsel\_1} \neq \text{varsel\_3}$	Sí
$\text{varsel\_1} < \text{varsel\_3}$	Sí

**Tabla 5.14:** Hipótesis alternativa para las instancias DS05/(100,5) del Modelo  $C_R$ .

Hipótesis alternativa $H_1$	Se rechaza $H_0$
$\text{varsel\_1} \neq \text{varsel\_0}$	Sí
$\text{varsel\_1} > \text{varsel\_0}$	Sí
$\text{varsel\_1} \neq \text{varsel\_1}$	Sí
$\text{varsel\_1} > \text{varsel\_1}$	Sí
$\text{varsel\_1} \neq \text{varsel\_3}$	Sí
$\text{varsel\_1} > \text{varsel\_3}$	Sí
$\text{varsel\_0} \neq \text{varsel\_1}$	No
$\text{varsel\_0} \neq \text{varsel\_3}$	No
$\text{varsel\_1} \neq \text{varsel\_3}$	No

**Tabla 5.15:** Hipótesis alternativa para las instancias DS10/(100,5) del Modelo  $C_R$ .

muy poca, esto es mas visible cuando se experimenta con  $\text{varsel\_0}$  y  $\text{varsel\_1}$  para los diferentes modelos.

## Experimento E

En esta parte del trabajo se desea fortalecer el modelo  $A_R$ . Para esto se realizan pruebas con dicho modelo agregando una a una tres restricciones que son válidas. Esto se mide en 80 instancias de los diferentes tipos y tamaños del problema para el modelo  $A_R$ . Cada una de estas se mide a cuatro niveles (model\_1, model\_2, model\_3 y model\_4) y en total se tienen 320 pruebas. Las tres restricciones que se van a probar son (E.1), (E.2) y (E.3) mencionadas en la sección 4.2.

**model\_1** = modelo  $A_R$

**model\_2** = modelo  $A_R$  + (E.1)

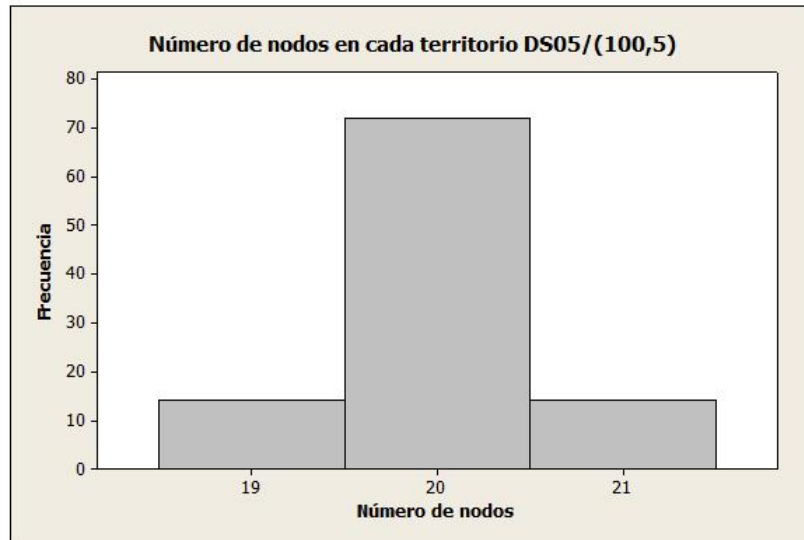
**model\_3** = modelo  $A_R$  + (E.2)

**model\_4** = modelo  $A_R$  + (E.3)

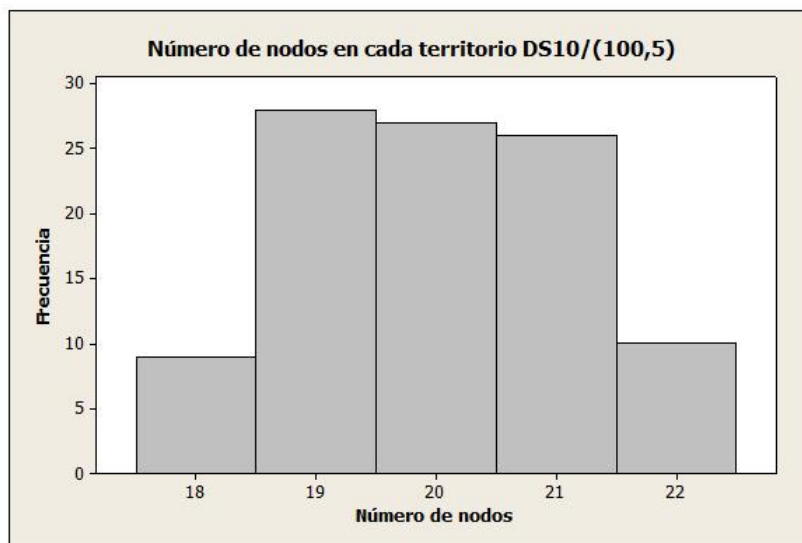
Como se mencionó anteriormente, la restricción (E.3) no es estrictamente válida para el problema en estudio por tal motivo, se hizo un histograma de frecuencias en el cual se muestra que el promedio de los nodos que quedan asignados en cada territorio formado es siempre menor o igual que  $(2n/p)$  y por consiguiente la restricción (E.3) se espera que sea válida. Se puede ver en los histogramas (Figuras 5.6 y 5.7) que entre menor sea  $\tau$ , el número de nodos en cada territorio se acerca al número promedio  $n/p$  y nunca es mayor que  $2n/p$ . En la Tabla 5.16 y en la Figura 5.8 se pueden observar los resultados obtenidos. En todas las experimentaciones se considera como criterio de parada 2,000 segundos.

En la primera columna de la Tabla 5.16 se muestran los tipos de instancias a probar en los experimentos, en la segunda, el tamaño de éstas, en la tercera, cuarta, quinta y sexta columnas se presentan los tiempos de cómputo promedio utilizado por el MRA para las diferentes instancias en los niveles model\_1, model\_2, model\_3 y model\_4, respectivamente.

La mayoría de las instancias del modelo  $A_R$  (mostradas en la Tabla 5.16) se resolvieron de manera óptima excepto 13 de tipo DS05/(100,5) en el nivel model\_2.



**Figura 5.6:** Nodos en cada territorio formado para instancias DS05/(100,5)



**Figura 5.7:** Nodos en cada territorio formado para instancias DS10/(100,5)

Para las instancias de tipo DS10/(100,5), una en el nivel model\_1 y 15 en el nivel model\_2 no fueron óptimas.

En la primera columna de las Tablas 5.17, 5.18, 5.19 y 5.20 se muestra la Hipótesis alternativa ( $H_1$ ) presente en la prueba que se realiza de dos en dos (con la prueba de Mann-Whitney) en caso de que no haya evidencia para decir que los tratamientos son iguales y en la segunda columna se escribe Sí o No dependiendo de si se acepta o rechaza la Hipótesis nula ( $H_0: \mu_1 = \mu_2$ ).

Tipo	$(n, p)$	model_1	model_2	model_3	model_4
DS05	(60,4)	66.3	139.8	93.7	82.4
DS10	(60,4)	47.1	99.5	62.8	66.1
DS05	(100,5)	961.5	(*)1779.0	984.3	1026.9
DS10	(100,5)	764.6	(*)1811.2	961.2	983.4
(*)Soluciones que se quedaron entre el 73 y 95 % del valor óptimo (en promedio).					

**Tabla 5.16:** Tiempo de cómputo promedio, fortalecer el modelo  $A_R$ .

Hipótesis alternativa $H_1$	Se rechaza $H_0$
model_1 $\neq$ model_2	Sí
model_1 $<$ model_2	Sí
model_1 $\neq$ model_3	Sí
model_1 $<$ model_3	Sí
model_1 $\neq$ model_4	Sí
model_1 $<$ model_4	Sí
model_2 $\neq$ model_3	Sí
model_2 $>$ model_3	Sí
model_2 $\neq$ model_4	Sí
model_2 $>$ model_4	Sí
model_3 $\neq$ model_4	No

**Tabla 5.17:** Hipótesis alternativa para las instancias DS05/(60,4), para fortalecer  $A_R$ .

Por medio de la prueba estadística se encontró que el valor de  $q$  era menor a  $\beta$  para las diferentes instancias, es decir no hay evidencia para decir que los tratamientos son iguales. Se procedió de manera similar a lo ocurrido con en modelo  $A_R$  en el experimento D. Después de analizar los tratamientos de par en par con la prueba de Mann-Whitney (ver Tablas 5.17, 5.18, 5.19 y 5.20) y los tiempos presentes en la

Tabla 5.16 se concluye que bajo la opción modelo\_1 se registran los mejores tiempos (menores).

Hipótesis alternativa $H_1$	Se rechaza $H_0$
model_1 $\neq$ model_2	Sí
model_1 < model_2	Sí
model_1 $\neq$ model_3	No
model_1 $\neq$ model_4	Sí
model_1 < model_4	Sí
model_2 $\neq$ model_3	Sí
model_2 > model_3	Sí
model_2 $\neq$ model_4	Sí
model_2 > model_4	Sí
model_3 $\neq$ model_4	No

**Tabla 5.18:** Hipótesis alternativa para las instancias DS10/(60,4), para fortalecer  $A_R$ .

Hipótesis alternativa $H_1$	Se rechaza $H_0$
model_1 $\neq$ model_2	Sí
model_1 < model_2	Sí
model_1 $\neq$ model_3	No
model_1 $\neq$ model_4	No
model_2 $\neq$ model_3	Sí
model_2 > model_3	Sí
model_2 $\neq$ model_4	Sí
model_2 > model_4	Sí
model_3 $\neq$ model_4	No

**Tabla 5.19:** Hipótesis alternativa para las instancias DS05/(100,5), para fortalecer  $A_R$ .

Hipótesis alternativa $H_1$	Se rechaza $H_0$
model_1 $\neq$ model_2	Sí
model_1 < model_2	Sí
model_1 $\neq$ model_3	Sí
model_1 < model_3	Sí
model_1 $\neq$ model_4	Sí
model_1 < model_4	Sí
model_2 $\neq$ model_3	Sí
model_2 > model_3	Sí
model_2 $\neq$ model_4	Sí
model_2 > model_4	Sí
model_3 $\neq$ model_4	No

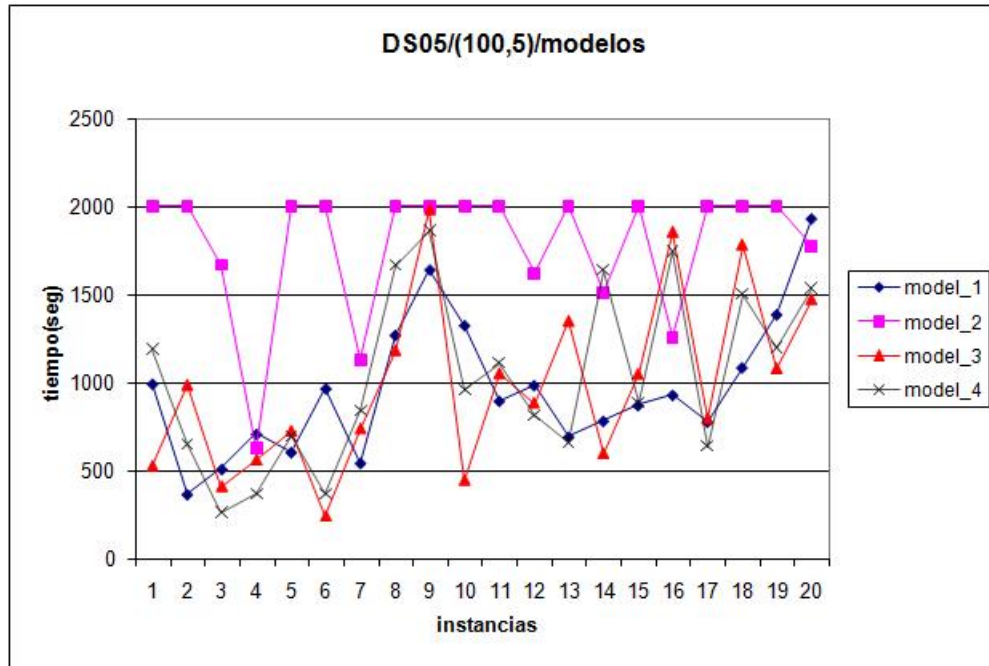
**Tabla 5.20:** Hipótesis alternativa para las instancias DS10/(100,5), para fortalecer  $A_R$ .

En la Figura 5.8 se puede observar que los modelos con las opciones model\_1, model\_3 y model\_4 obtiene los tiempos promedio de solución muy similares y menores que el model\_2. En la Tabla 5.16 se observa que bajo la opción model\_1 siempre se obtiene el menor tiempo de cómputo promedio para los diferentes tamaños y tipos de problemas. Si bien se mejoró la representación del poliedro de soluciones enteras no se pudo mejorar los tiempos de solución agregando estas restricciones. Se opta por dejar el modelo  $A_R$  original (model\_1) para las siguientes experimentaciones.

## Experimento F

En esta parte del trabajo se estudia la sensibilidad del algoritmo del método con respecto a la elección del parámetro  $p$ , es decir, se desea saber si el variar el número de territorios afecta el tiempo de solución. Para esto se realizan pruebas con las primeras 5 instancias del tipo DS05 probadas en los experimentos anteriores para cada modelo ( $A_R$ ,  $B_R$  y  $C_R$ ). Para los problemas de tamaño 100 se varia  $p$  en 6 valores (2, 3, 4, 5, 6, 7) y para los problemas de tamaño 60 se varia  $p$  en 4 valores (2, 3, 4, 5), lo cual da un total de 150 corridas. Todos estos experimentos se corren con





**Figura 5.8:** Restricciones válidas en el modelo  $A_R$ , en instancias DS05/(100,5)

un criterio de parada de 2,000 segundos. Las Tablas 5.21, 5.22 y 5.23 muestran en la primera columna las instancias con las cuales se realizan los experimentos, en la segunda, tercera, cuarta, quinta, sexta y séptima columna se presentan los tiempos de cómputo empleado por el MRA en cada uno de los niveles  $p = 2, 3, 4, 5, 6$  y  $7$ , respectivamente. En la primera columna de las Tablas 5.24 y 5.25 se muestra la Hipótesis alternativa ( $H_1$ ) presente en la prueba que se realiza de dos en dos (con la prueba de Mann-Whitney) en caso de que no haya evidencia para decir que son iguales los tratamientos y en la segunda columna se escribe Sí o No dependiendo de si se acepta o rechaza la Hipótesis nula ( $H_0: \mu_1 = \mu_2$ ).

Todas las instancias presentes en las Tablas 5.21, 5.22 y 5.23 se resolvieron de manera óptima, excepto la instancia DU100-04 con  $p = 6$  del modelo  $C_R$ , la cual se quedó con un intervalo de optimalidad relativa (IOR) del 18.78%.

Por medio de la prueba estadística Kruskal-Wallis se obtuvo el valor de  $q$  para las instancias de los modelos  $A_R, B_R$  y  $C_R$ . El valor de  $q$  para los modelos  $A_R$  y  $C_R$  era mayor a  $\beta$ , es decir la prueba estadística no proporciona información signi-

Instancia	$p=2$	$p=3$	$p=4$	$p=5$	$p=6$	$p=7$
DU60-01	55.74	57.95	91.13	51.95	–	–
DU60-02	42.86	56.57	34.03	47.04	–	–
DU60-03	64.95	39.81	97.07	86.35	–	–
DU60-04	54.95	88.22	77.32	158.52	–	–
DU60-05	70.91	62.74	81.30	95.77	–	–
DU100-01	574.46	504.43	476.76	1000.79	1000.77	740.13
DU100-02	453.41	887.06	295.41	262.24	406.68	1026.71
DU100-03	715.23	1027.91	887.80	453.45	1019.80	1586.03
DU100-04	484.39	813.22	863.33	794.01	660.23	730.28
DU100-05	1536.80	240.04	1146.76	898.69	756.72	844.00

**Tabla 5.21:** Tiempo de ejecución en 10 instancias DS05 del Modelo  $A_R$  variando  $p$ .

Instancia	$p=2$	$p=3$	$p=4$	$p=5$	$p=6$	$p=7$
DU60-01	9.50	3.86	4.03	2.37	–	–
DU60-02	4.96	4.00	4.50	2.88	–	–
DU60-03	9.51	3.66	4.19	2.98	–	–
DU60-04	9.57	4.13	2.79	2.60	–	–
DU60-05	5.17	3.78	3.82	4.51	–	–
DU100-01	347.41	294.02	23.76	43.42	18.22	391.11
DU100-02	389.57	251.72	23.46	52.99	22.12	43.65
DU100-03	386.81	293.02	20.50	20.23	25.56	19.25
DU100-04	186.77	421.49	23.97	18.65	16.70	71.20
DU100-05	323.51	162.91	22.43	27.48	20.64	54.63

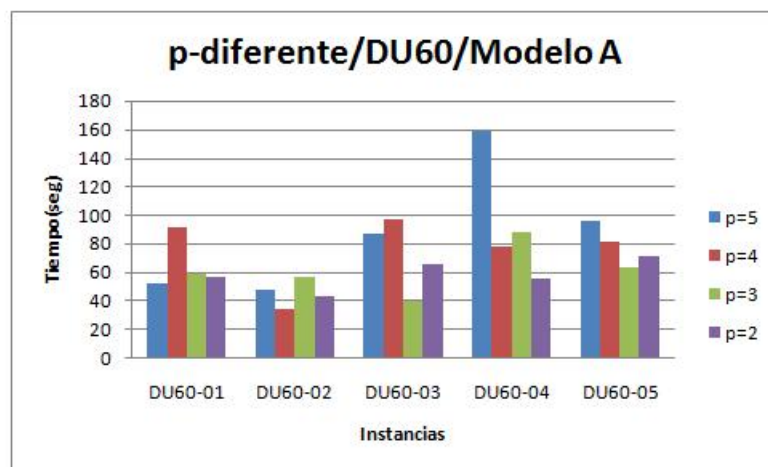
**Tabla 5.22:** Tiempo de ejecución en 10 instancias DS05 del Modelo  $B_R$  variando  $p$ .

ficativa para decir que los tratamientos son diferentes. Por tanto considero que los tratamientos son iguales. Sin embargo, al ver los tiempos en las Tablas 5.21, 5.23 y en las Figuras 5.9, 5.10, 5.13 y 5.14 se observa que la mayoría de estos tiempos

Instancia	$p=2$	$p=3$	$p=4$	$p=5$	$p=6$	$p=7$
DU60-01	64.97	33.38	45.47	30.26	–	–
DU60-02	28.27	27.52	21.78	21.93	–	–
DU60-03	63.81	37.26	50.43	52.67	–	–
DU60-04	62.18	66.81	116.25	96.00	–	–
DU60-05	22.95	46.77	12.02	18.73	–	–
DU100-01	372.64	627.38	507.01	354.44	422.74	594.80
DU100-02	319.93	1017.20	531.82	845.21	257.65	961.00
DU100-03	539.93	459.81	290.41	496.55	421.54	1223.55
DU100-04	754.23	862.69	561.76	412.57	(*)2000.21	403.22
DU100-05	355.71	547.60	605.83	621.41	548.81	349.75

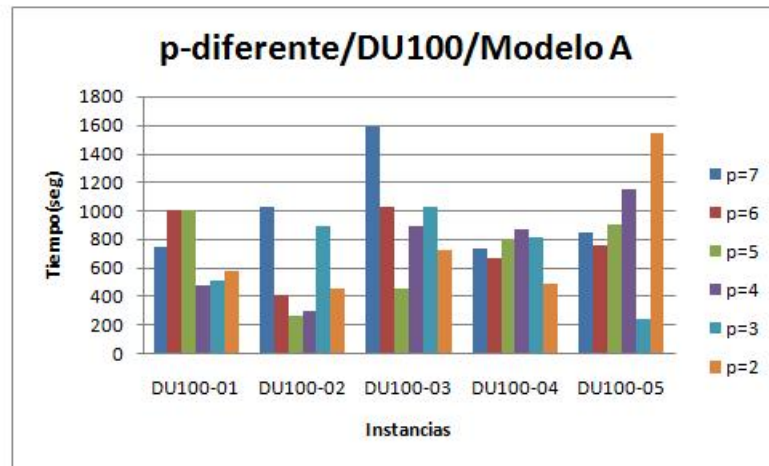
(\*)Solución que se quedó a un 18.78 % del valor óptimo

**Tabla 5.23:** Tiempo de ejecución en 10 instancias DS05 del Modelo  $C_R$  variando  $p$ .

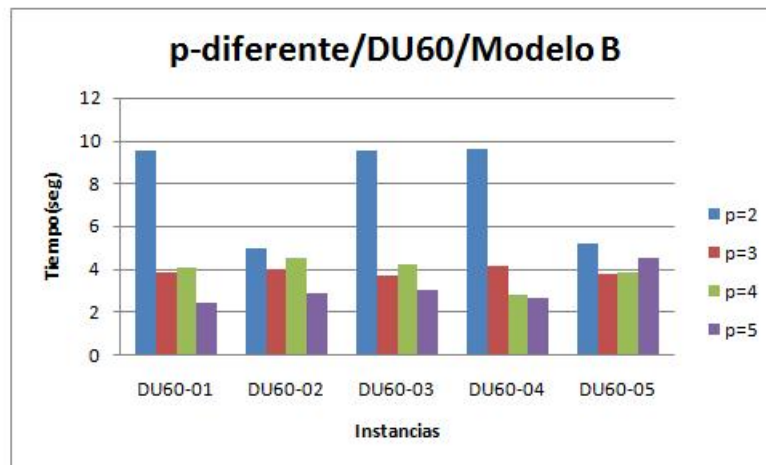


**Figura 5.9:** Tiempo de ejecución en 5 instancias DS05/(60,4) del Modelo  $A_R$  variando  $p$ .

aumentan conforme aumenta  $p$ , es decir se vuelve más difícil de resolver.

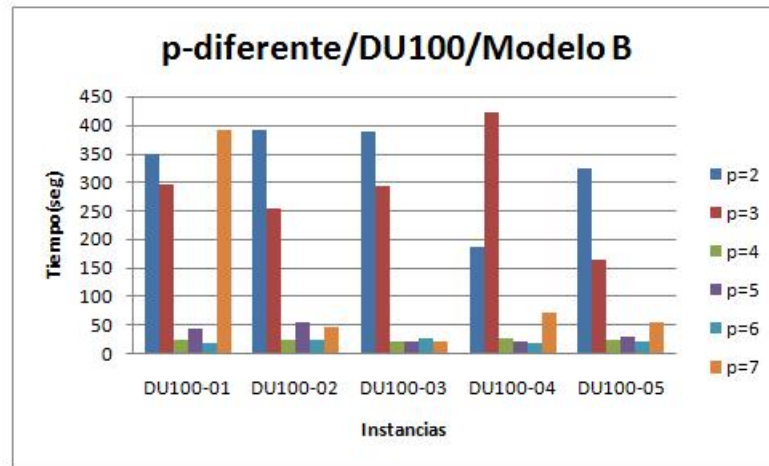


**Figura 5.10:** Tiempo de ejecución en 5 instancias DS05/(100,5) del Modelo  $A_R$  variando  $p$ .

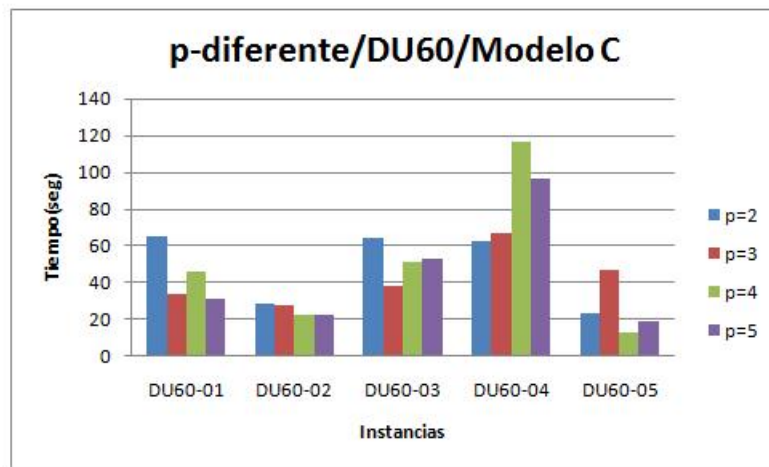


**Figura 5.11:** Tiempo de ejecución en 5 instancias DS05/(60,4) del Modelo  $B_R$  variando  $p$ .

En el modelo  $B_R$  el valor de  $q$  es menor a  $\beta$ , es decir no hay evidencia para decir que los tratamientos son iguales por tanto se procedió a analizar los datos de dos en dos con la prueba de Mann-Whitney (ver Tablas 5.24 y 5.25). Se analizaron

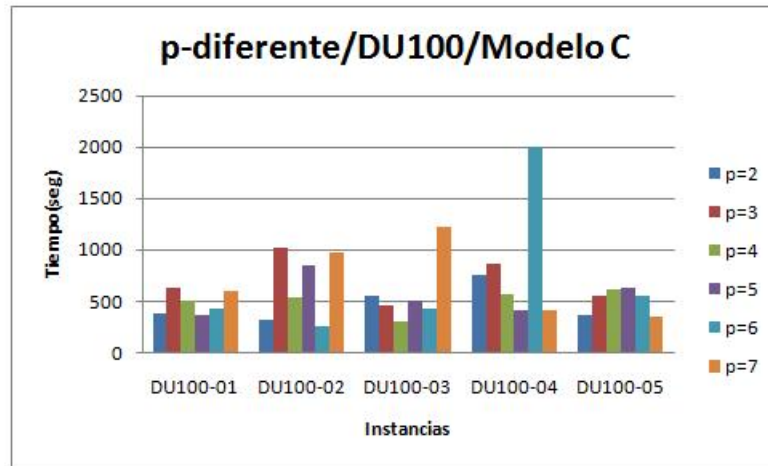


**Figura 5.12:** Tiempo de ejecución en 5 instancias DS05/(100,5) del Modelo  $B_R$  variando  $p$ .



**Figura 5.13:** Tiempo de ejecución en 5 instancias DS05/(60,4) del Modelo  $C_R$  variando  $p$ .

los tiempos presentes en la Tabla 5.22 y en las Figuras 5.11 y 5.12. Se llegó a la conclusión que para la mayoría de las instancias de este modelo ( $B_R$ ) disminuye el tiempo de ejecución conforme aumenta el tamaño de los territorios, es decir se vuelve



**Figura 5.14:** Tiempo de ejecución en 5 instancias DS05/(100,5) del Modelo  $C_R$  variando  $p$ .

Hipótesis alternativa $H_1$	Se rechaza $H_0$
$p=2 \neq p=3$	Sí
$p=2 > p=3$	Sí
$p=2 \neq p=4$	Sí
$p=2 > p=4$	Sí
$p=2 \neq p=5$	Sí
$p=2 > p=5$	Sí
$p=3 \neq p=4$	No
$p=3 \neq p=5$	No
$p=4 \neq p=5$	No

**Tabla 5.24:** Hipótesis alternativa para las 5 instancias DS05/(60,4),  $p$ -diferente en  $B_R$ .

más fácil.

Hipótesis alternativa $H_1$	Se rechaza $H_0$
$p=2 \neq p=3$	No
$p=2 \neq p=4$	Sí
$p=2 > p=4$	Sí
$p=2 \neq p=5$	Sí
$p=2 > p=5$	Sí
$p=2 \neq p=6$	Sí
$p=2 > p=6$	Sí
$p=2 \neq p=7$	No
$p=3 \neq p=4$	Sí
$p=3 > p=4$	Sí
$p=3 \neq p=5$	Sí
$p=3 > p=5$	Sí
$p=3 \neq p=6$	Sí
$p=3 > p=6$	Sí
$p=3 \neq p=7$	No
$p=4 \neq p=5$	No
$p=4 \neq p=6$	No
$p=4 \neq p=7$	No
$p=5 \neq p=6$	No
$p=5 \neq p=7$	No
$p=6 \neq p=7$	No

**Tabla 5.25:** Hipótesis alternativa para las 5 instancias DS05/(100,5),  $p$ -diferente en  $B_R$ .

## Experimento G

En esta sección del trabajo se desea conocer el alcance del método exacto para los diferentes modelos presentados ( $A_R$ ,  $B_R$ ,  $C_R$  y  $D_R$ ). Para esto, se realizan pruebas variando el tamaño de las instancias, el límite de tiempo y en caso necesario el criterio de optimalidad relativa (COR). En los experimentos siguientes se considera

como criterio de parada 7,200 segundos, a menos que se especifique otra cosa.

Para el modelo  $A_R$  se corrió primero una instancia (DU150-01) de 150 nodos y 10 territorios con un parámetro de tolerancia  $\tau= 0.05$  variando el límite de tiempo y el gap relativo, ya que el método exacto no encontraba solución óptima para la primera hora, esto se muestra en la Tabla 5.26. En la primera columna de esta tabla se muestran los diferentes criterios de optimalidad relativa (COR) considerados en la experimentación, en la segunda, el criterio de parada de tiempo en el MRA, en la tercera, el número de iteraciones realizadas por el método, en la cuarta, el número de nodos o subproblemas en el árbol enumerativo generado por el MRA, en la quinta columna se muestra la mejor solución entera mixta (PEM) encontrada, la sexta, presenta la cota inferior proporcionada por el método y la séptima fila presenta el intervalo de optimalidad relativa (IOR) en porcentaje.

Criterios de parada						
COR	Tiempo (segundos)	Número iteraciones	Número nodos	Solución PEM	Cota inferior	IOR %
0.01	3,600	450248	1010	162.39	73.71	54.60
0.10	3,600	478919	1074	162.39	73.71	54.60
0.20	3,600	475407	1040	162.39	73.71	54.60
0.20	7,200	991799	1997	158.02	76.39	51.65

**Tabla 5.26:** Alcance del método para el modelo  $A_R$  en la instancia DU150-01 con  $n=150$  y  $p=10$ .

Como se puede ver en la Tabla 5.26 el método exacto comienza a batallar en encontrar una solución óptima en un problema con 150 nodos. La solución encontrada se queda al menos al 51.65% del valor óptimo con criterio de parada de dos horas, la cual es un margen de error bastante pobre. Se aumentó el valor del COR esperando que al darle un valor más grande el método pudiera más rápido las ramas en donde no hubiera una solución factible o mejor a la que ya se tenía y por



consiguiente que el algoritmo convergiera más rápido al óptimo, sin embargo, no mejoró considerablemente el tiempo ni la calidad de la solución.

Instancia	Tiempo (segundos)	Número iteraciones	Número nodos	Solución PEM	Cota inferior	IOR %
DU150-02	3,600	350203	1290	176.10	69.53	60.51
DU150-03	3,600	433592	1740	210.43	75.68	64.03
DU150-04	3,600	405536	914	159.90	65.31	59.15
DU150-05	3,600	842460	2838	129.14	93.44	27.64

**Tabla 5.27:** Alcance del método para el modelo  $A_R$  para cuatro instancias con  $n=150$ ,  $p=10$  y  $COR=0.01$ .

Se probaron otras cuatro instancias (ver Tabla 5.27) para el modelo  $A_R$  y se obtuvieron intervalos de optimalidad relativa (IOR) similares al de la instancia anterior (DU150-01) con un criterio de parada de una hora. De este experimento, cuatro instancias de cinco resultaron con IOR muy altos, es decir en una hora el método exacto está muy alejado del valor óptimo.

Para el modelo  $B_R$  se considera una instancia de cada tipo, una con 150 nodos y otras más con hasta 350 nodos, todas estas con un  $\tau=0.05$  y variando en caso necesario el límite de tiempo. Los resultados se muestran en la Tabla 5.28. En la primera columna de esta tabla se muestra el número de nodos  $n$  en la instancia a probar, en la segunda el nombre de esta instancia, en la tercera, el tiempo empleado por el MRA, en la cuarta, el número de iteraciones realizadas por el método, en la quinta, el número de nodos en el árbol enumerativo generado por el método, en la sexta, se muestra la mejor solución entera mixta (PEM) obtenida, en la séptima, la cota inferior determinada por el MRA y por último se presenta el intervalo de optimalidad relativa (en porcentaje) en la octava columna.

Como se puede ver en la Tabla 5.28, el método exacto puede resolver para el

n	Instancia	Tiempo (segundos)	Número iteraciones	Número nodos	Solución PEM	Cota inferior	IOR %
150	DU150-01	78.57	6974	0	8436.36	8400.83	0.42
250	DU250-01	6,686	306110	3291	14689.45	14542.87	0.99
300	DU300-01	7,200	130221	2009	17820.03	16630.11	6.67
350	DU350-01	7,200	98381	1151	22029.10	18480.12	16.11
350	DU350-02	7,200	76343	1132	20373.81	18087.62	11.22
350	DU350-03	7,200	89857	1026	21860.59	18057.50	17.39
350	DU350-04	7,200	80538	809	21588.06	18381.77	14.85
350	DU350-05	7,200	81919	637	21993.21	18493.01	15.91

**Tabla 5.28:** Alcance del método para el modelo  $B_R$  en instancias de diferentes tamaños con  $\tau=0.05$  y  $p=10$ .

modelo  $B_R$  sin ningún problema hasta instancias de 250 nodos en dos horas, la cual es una solución muy buena para este estudio. Sin embargo, el MRA no pudo resolver en dos horas instancias de más de 300 nodos.

El modelo  $C_R$  se analizó primero para una instancia (DU150-01) de 150 nodos y 10 territorios variando el límite de tiempo y el COR esperando encontrar una solución óptima. En la primera columna de la Tabla 5.29 se muestran los diferentes criterios de optimalidad relativa (COR) considerados en la experimentación, en la segunda, el criterio de parada de tiempo en el MRA, en la tercera, el número de iteraciones realizadas por el método, en la cuarta, el número de nodos o subproblemas en el árbol enumerativo generado por el MRA, en la quinta columna se muestra la mejor solución entera mixta (PEM) encontrada, la sexta, presenta la cota inferior proporcionada por el método y la séptima fila presenta el intervalo de optimalidad relativa (IOR) en porcentaje.

Como se puede ver en la Tabla 5.29, el método comienza a batallar en resolver una instancia con 150 nodos (con un COR = 0.01) ya que con un criterio de parada

Criterios de parada						
COR	Tiempo (segundos)	Número iteraciones	Número nodos	Solución PEM	Cota inferior	IOR %
0.01	3,600	683337	2657	144.74	78.6893	45.63
0.01	7,200	1662624	6651	124.82	106.26	14.86
0.10	3,600	651513	2532	144.74	78.67	45.64
0.10	7,200	1642375	6596	124.82	106.26	14.86
0.20	3,600	663075	2572	144.74	78.67	45.64
0.20	6,618	1373574	5567	131.99	105.67	19.94

**Tabla 5.29:** Alcance del método para el modelo  $C_R$  en la instancia DU150-01 con  $n=150$  y  $p=10$ .

de una hora se queda a un 45.63% del óptimo. Se varió el COR (en 0.10 y 0.20) esperando que el algoritmo termine más rápido y que al menos la solución obtenida esté a un 10 o 20% del óptimo (se renuncia al óptimo). Sin embargo, en una hora las instancias se quedaron al 45.64% del óptimo, similar a la instancia con  $COR = 0.01$ . Dando un criterio de parada más amplio (de dos horas) se encuentra una solución a un 14.86% de la óptima, la cual es relativamente buena. Al resolver la instancia con un  $COR = 0.20$  (se renuncia a un 20% del valor óptimo) y límite de dos horas, el método termina antes de utilizar este tiempo, debido a que obtiene una solución a un 19.94% del valor óptimo.

Se probaron otras cuatro instancias (ver Tabla 5.30) para el modelo  $C_R$  y se obtuvieron IOR similares al de la instancia anterior (DU150-01) con un criterio de parada de una hora, excepto para la instancia DU150-05 con  $IOR = 5.16\%$ . De este experimento, cuatro instancias de cinco resultaron con IOR muy altos, es decir en una hora el método exacto encuentra una solución muy alejada del valor óptimo.

El modelo  $D_R$  se analizó para diferentes instancias variando  $n$  desde 100 hasta 2000 nodos, la instancia de 100 nodos con  $p=5$ , las instancias de tamaño mayor

Instancia	Tiempo (segundos)	Número iteraciones	Número nodos	Solución PEM	Cota inferior	IOR %
DU150-02	3,600	844950	1847	144.47	80.67	44.15
DU150-03	3,600	560476	2116	152.93	57.94	62.11
DU150-04	3,600	573221	1931	151.74	74.14	51.13
DU150-05	3,600	1884584	8421	131.11	124.33	5.16

**Tabla 5.30:** Alcance del método para el modelo  $C_R$  en cuatro instancias con  $n=150$ ,  $p=10$  y  $COR=0.01$ .

a 100 y menor a 2000 con  $p=10$  y las instancias de 2000 nodos con  $p=30$ , todas con un  $COR = 0.01$  y  $\tau = 0.05$ . Para todas las instancias mostradas en la Tabla 5.31, el número de subproblemas generados en el nodo raíz fue cero y todas las soluciones fueron las óptimas, es decir, con un  $IOR = 0$ . El tiempo de ejecución solo se invirtió en el nodo raíz, es decir es el tiempo que el método utilizó en generar cortes (agregar restricciones) para reducir el poliedro de soluciones factibles.

$n$	$p$	Tiempo (segundos)	Número iteraciones	Solución óptima
100	5	1.67	341	48.46
200	10	10.18	857	35.21
400	10	81.83	2304	28.85
600	10	200	4297	24.91
800	10	548.68	5143	22.23
1000	10	879.14	7813	18.65
1200	10	1530.74	9717	18.71
1400	10	2098.06	11305	16.45
1500	10	2703.27	12331	17.14

**Tabla 5.31:** Alcance del método para varias instancias el modelo  $D_R$  con  $\tau=0.05$ .

Se probaron otras cinco instancias con 2000 nodos,  $p=30$  y con  $COR = 0.01$ .

El MRA no encontró solución entera en ninguna de estas instancias con criterio de parada de una hora.

Como se puede ver en la Tabla 5.31, el método exacto puede resolver sin ningún problema instancias de hasta 1500 nodos en a lo más 45 minutos para el modelo  $D_R$ , todas las soluciones fueron las óptimas.

# Capítulo 6

## Conclusiones

En este capítulo se presentan las conclusiones en base a los resultados obtenidos en los experimentos mostrados en el capítulo anterior, las conclusiones de la tesis en general y las aportaciones que brinda este trabajo.

### 6.1. Conclusiones de los Experimentos

Como se vio en los resultados del capítulo anterior, el modificar el parámetro de las prioridades (sobre las variables a ramificar) en los modelos fue importante ya que al dar prioridades a las variables  $x_{ii}$  sobre las  $x_{ij}$  mejoraron los tiempos de solución significativamente. Esto se debe a que el fijar una variable  $x_{ii}$  en 0, por ejemplo, implica que todas las variables  $x_{ij}$  tienen que ser 0, para todo  $j \in V$ , lo cual acelera la convergencia del método.

Los conjuntos ordenados especiales (SOS) no se deben utilizar para estos modelos ya que aumentan el tiempo de solución y en algunos casos no es posible encontrar soluciones óptimas en tiempos razonables.

El modificar el parámetro de sondeos de implicaciones lógicas no presentó cambios significativos en el tiempo de solución para ningún modelo, por lo cual, se deja como opción que el método decida sobre este parámetro de acuerdo al progreso del problema.

Se encuentra que la regla, para seleccionar la variable a ramificar, que mejora los tiempos de solución en las instancias de 60 nodos del modelo  $A_R$  y  $C_R$  es la de

ramificar sobre variables con infactibilidad máxima. Para las instancias de 100 nodos del modelo  $A_R$  y  $C_R$  es mejor que el método ramifique la variable automáticamente seleccionada, al igual que para las instancias de 60 y 100 nodos del modelo  $B_R$ .

A fin de fortalecer el modelo  $A_R$ , si bien se mejoró la representación del poliedro de soluciones enteras al agregar las restricciones válidas, no se mejoró el tiempo de solución que ya se tenía del modelo  $A_R$  original.

Se encuentra que al variar el número de territorios ( $p$ ) para la mayoría de las instancias de los modelos  $A_R$  y  $C_R$  el tiempo promedio de ejecución aumenta conforme aumenta  $p$ , es decir se vuelve más difícil de resolver. En cambio, para la mayoría de las instancias del modelo  $B_R$  el tiempo de ejecución disminuye conforme aumenta  $p$ , es decir se vuelve más fácil.

Por último, se sabe que para resolver una instancia de 150 o más nodos para los modelos  $A_R$  y  $C_R$  el método comienza a batallar en encontrar una solución óptima. Para el modelo  $B_R$  se pueden resolver instancias de hasta 250 nodos en tiempos razonables (menos de 2 horas). Para el modelo  $D_R$  se pueden resolver instancias de hasta 1500 nodos de manera óptima en menos de 45 minutos.

## 6.2. Conclusiones de la Tesis

El Método de Ramificación y Acotamiento (MRA) puede dar soluciones óptimas para algunas instancias (de tamaños chico y mediano) de los cuatro modelos presentados en este trabajo.

El modificar adecuadamente algunos de los parámetros algorítmicos del método ayuda a obtener soluciones óptimas en mejores tiempos.

Si bien se puede mejorar la representación del poliedro de soluciones enteras para un problema al agregar algunas restricciones válidas, esto no asegura que el algoritmo converja más rápido a la solución óptima ya que se incrementan el número de restricciones.

El modificar algún parámetro (como  $p$  en el presente trabajo) en un modelo puede afectar el progreso del algoritmo del método o en las soluciones encontradas.

El MRA tiene un buen desempeño en la solución de algunas instancias del

problema de diseño de territorios en estudio y en algunas de sus variaciones.

En modelos como  $D_R$  el MRA no tuvo la necesidad de ramificar, es decir, resolvió el problema en el nodo raíz del árbol de enumeración invirtiendo el tiempo en generar cortes (restricciones válidas) para reducir el poliedro de soluciones factibles.

### 6.3. Aportaciones del Trabajo

Este trabajo proporciona cotas inferiores para los cuatro modelos presentados, las cuales, pueden ser utilizadas para medir la calidad de las heurísticas desarrolladas en otros estudios.

Los valores de los parámetros algorítmicos que dieron mejores resultados en este estudio pueden ser utilizados en problemas con estructura similar.

Además, se obtiene el alcance del método exacto en la solución de estos modelos. Con esta información se puede justificar plenamente el uso de las técnicas heurísticas en problemas de mayor tamaño.

Ahora bien, si se consideraran como modelos originales los mismos modelos (A, B, C, D) sin las restricciones de conexidad, entonces en este trabajo se estarían proporcionando soluciones óptimas a éstos.

Como trabajo a futuro se considera el estudio de la calidad de estas cotas inferiores encontradas, el desarrollo de otras cotas de mayor o igual calidad utilizando técnicas más sofisticadas y el integrar este conocimiento en el desarrollo de métodos para resolver en forma exacta los modelos originales (no relajados).

Por ejemplo, un trabajo reciente en donde se desarrollan cotas más sofisticadas para un problema del  $p$ -centro capacitado es presentado por Albareda Sambola, Díaz y Fernández [1]. Estudios de este tipo permiten un pronóstico optimista sobre la posible mejora de las cotas actuales para los modelos presentados aquí.



## GAMS, una herramienta en la solución de problemas.

Para dar solución al problema en estudio se utiliza una herramienta computacional muy eficiente llamada GAMS (por sus siglas en inglés, General Algebraic Modeling System). GAMS es un lenguaje de programación que permite modelar, analizar y resolver diferentes problemas de optimización [7].

Existen otros lenguajes similares a GAMS como AMPL y AIMMS, todos presentan características análogas.

GAMS tiene muchas características y ventajas a la vez, las más importantes son:

1. Tiene la capacidad para pasar de resolver un problema de pequeña dimensión a problemas más grandes sin hacer una gran variación al código.
2. Puede separar el proceso de modelado del proceso de resolución del problema. El usuario sólo se preocupa por formular el problema correctamente. Una vez expresado el modelo en la notación de GAMS, éste manda llamar a uno de los optimizadores disponibles para obtener una solución. Lo anterior permite cambiar el modelo para mejorarlo o completarlo cómodamente.
3. La representación de un problema de optimización coincide, prácticamente, con la descripción matemática de este problema. Esto hace que el modelo en

GAMS sea sencillo de comprender para aquellos que están familiarizados con la optimización. Más adelante, al ver la formulación matemática y el código GAMS del problema a estudiar, se pondrá en evidencia lo anterior.

GAMS es el lenguaje algebraico de modelado más ampliamente difundido comercialmente.

Un lenguaje de modelado (por ejemplo GAMS) permite utilizar diferentes optimizadores para la solución de un mismo problema. Esta característica representa una gran ventaja por la flexibilidad que aporta en la selección del optimizador más adecuado a las características del problema. El mejor método para un problema concreto depende de las características del problema, en este caso, se utiliza el módulo GAMS/CPLEX (ver Apéndice B).

En la Tabla A.1 se presenta el código fuente del programa en GAMS para una instancia con  $n = 5$ ,  $p = 2$ , tipo DS10 y modelo  $A_R$ .

```

$title districting
$ontext
    Territory design problem model
References:
[1] R.Z. Rios-Mercado and E. Fernandez. A reactive GRASP for a
    sales territory design problem with multiple balancing
    requirements. Technical Report PISIS-2006-12, Graduate Program
    in Systems Engineering, UANL, Mexico, September 2006.
$offtext

Sets
    I   "Basic areas (BAs)" / 0*5 /
    A   "Activities" / customers, demand, workload /
    IC  "Co-ordinates" / cx,cy /
    K   "Territories" / 1*2 /

```

Alias (I,J);

Alias (I,J2);

Scalars

p "Number of territories" / 2 /

n "Number of BAs" / 5 /;

Parameters

t(a) "Tolerance for activity a" / customers 0.100000

demand 0.100000

workload 0.100000 /;

Table w(i,a) "Value of activity a in BA i"

	customers	demand	workload
0	854	2822	2674
1	799	2774	2707
2	841	2597	2616
3	846	2775	2702
4	839	2745	2658

Table coor(i,ic) "X-Y coordinates for basic area i"

	cx	cy
0	304.230560	203.060913
1	241.006790	101.112946
2	150.634567	215.043869
3	108.182716	226.939133
4	198.321381	187.434433

Parameter d(i,j) "Euclidean distance between basic areas i and j";

d(i,j) =

sqr(sqrt(coor(i,'cx')-coor(j,'cx'))+sqrt(coor(i,'cy')-coor(j,'cy')));

Parameter m(a) "Average value of activity a";

m(a) = sum(i,w(i,a))/p;

Free variables

z

zd "max dispersion objective"

Binary variables

x(i,i) " =1 if center is located at BA i; 0 otherwise"

x(i,j) " =1 if i is assigned to territory centered in j; 0 otherwise";

Equations

max\_dispersity

dispersity(j)

assignment(j)

location

balance\_lb1(i)

balance\_ub1(i)

balance\_lb2(i)

balance\_ub2(i)

max\_dispersity.. z =e= zd;

dispersity(j).. zd =g= sum(i,d(j,i)\*x(j,i));

assignment(j).. sum(i,x(j,i)) =e= 1;

location.. sum(j,x(j,j)) =e= p;

balance\_lb1(i).. sum(j,w(j,'customers')\*x(j,i))  
=g= (1-t('customers'))\*m('customers')\*x(i,i);

balance\_ub1(i).. sum(j,w(j,'customers')\*x(j,i))  
=l= (1+t('customers'))\*m('customers')\*x(i,i);

balance\_lb2(i).. sum(j,w(j,'demand')\*x(j,i))  
=g= (1-t('demand'))\*m('demand')\*x(i,i);

balance\_ub2(i).. sum(j,w(j,'demand')\*x(j,i))  
=l= (1+t('demand'))\*m('demand')\*x(i,i);

\*Modelo A: TDP with two balancing constraint

model tdp\_b2 / all /;

```

tdp_b2.OptFile = 1;
x.prior(i,j) = 10;
x.prior(i,i) = 1;
tdp_b2.prioropt = 1;
option limcol = 0;
option limrow = 0;
option solprint = off;
option mip = cplex;

Solve tdp_b2 using mip minimizing z;
Display x.l

Para imprimir los resultados en un archivo
file resultadosDU10-01.sol;
put resultados;
resultados.nd = 0;
put n;
loop(i, put i.tl,loop[j, if{x.l(i,j)=1,put j.tl;}]);
putclose resultados;

```

**Tabla A.1:** Modelo en GAMS para una instancia con  $n = 5$ ,  $p = 2$ , tipo DS10 y modelo  $A_R$

Las ecuaciones declaradas bajo la palabra Equations (ver Tabla A.1) no son mas que las ya presentadas en la Sección 3.3. Es decir, la ecuacion max\_dispersity corresponde a (A.11), dispersity(j) a (A.12), assignment(j) a (A.13), location a (A.14), balance\_lb1(i) y balance\_lb2(i) a (A.15) para las actividades 1 y 2 respectivamente, balance\_ub1(i) y balance\_ub2(i) a (A.16) para las actividades 1 y 2 respectivamente.

# Apéndice B

## CPLEX

CPLEX una biblioteca de algoritmos para resolver algunos tipos especiales de problemas de optimización. GAMS/CPLEX permite a los usuarios combinar las capacidades de alto nivel de modelado en GAMS con el poder de los optimizadores de CPLEX [7].

Los algoritmos de CPLEX son diseñados para solucionar problemas de manera rápida y con la mínima intervención del usuario. Se requiere una licencia para que CPLEX pueda resolver los diferentes algoritmos para problemas de Programación Lineal (PL), programación con restricciones cuadráticas (PQ) y PEM.

La siguiente declaración puede ser usada dentro del programa de GAMS para especificar que se desea usar CPLEX:

```
Option LP = Cplex; {o QCP, MIP, MIQCP, RMIP o RMIQCP},
```

Esto debe aparecer antes de que se llame a resolver el problema (antes de la declaración `Solve`). Si CPLEX fue especificado como el optimizador por default durante la instalación de GAMS, no es necesaria la declaración anterior.

Mientras existen numerosas opciones disponibles, CPLEX toma automáticamente ciertos valores para algunas opciones (parámetros de control) que son generalmente adecuadas para un problema estándar de optimización. Sin embargo, cuando se trata de problemas difíciles, como pueden ser los de PL de muy gran tamaño, los

de programación no lineal o los de PEM, son convenientes pruebas específicas de ajuste con algunos parámetros. En particular, algunos relacionados con la eficiencia y estabilidad numérica del algoritmo.

En CPLEX existen varias opciones a las cuales se les puede modificar su valor para satisfacer las necesidades del problema. En base al conocimiento del problema en estudio se pueden modificar el valor de los parámetros siguientes: `epgap`, `varsel`, `probe`, `mipordind`, entre otras. Algunas de las opciones de CPLEX que guardan una relación con la forma de trabajar del MRA y con el problema en estudio se presentan en la Tabla B.1. Éstas deben ser declaradas en un fichero llamado `cplex.opt`. La siguiente instrucción debe ser declarada en el archivo GAMS para indicar a GAMS/CPLEX que lea el archivo `cplex.opt`: `tdp_b2.OptFile = 1;`

Opción	Descripción
<code>epgap</code>	Tolerancia relativa sobre el gap entre el mejor entero objetivo y el mejor nodo restante objetivo. Cuando el valor cae abajo del valor fijo de <code>epgap</code> , la optimización entera mixta es parada. Esta opción anula la opción <code>OptCR</code> de GAMS que proporciona estos valores iniciales. Rango: (0.0, 1.0) (default = GAMS <code>OptCR</code> )
<code>mipordind</code>	Las prioridades se deben asignar basadas en el conocimiento del problema. Las variables con prioridades más altas serán ramificadas antes de las variables de prioridades más bajas. Esta dirección de la búsqueda del árbol puede a menudo reducir dramáticamente el número de nodos buscados. Cuando fija a 0 (default), la opción <code>mipordind</code> de las instrucciones de CPLEX no usa prioridades para ramificar. Cuando se fija a 1, el orden de las prioridades son utilizadas. <sup>1</sup> (default =1)

<sup>1</sup>Las prioridades se asignan a las variables discretas usando el sufijo `.prior` en el modelo GAMS. Bajos valores `.prior` significan mayor prioridad. El sufijo `.prioropt` tiene que ser utilizado en GAMS para señalar a GAMS que debe exportar las prioridades al solver (ver Tabla A.1).

	<p><b>0</b> no usar prioridades para ramificación.</p> <p><b>1</b> utilizar el orden de las prioridades.</p>
probe	<p>Determina la cantidad de sondeos de implicación lógica realizados sobre un PEM. Estos sondeos pueden ser tan poderosos como consumir mucho tiempo.</p> <p>El ajuste del valor a 1 puede causar reducciones dramáticas o aumentos dramáticos en el tiempo de solución dependiendo del modelo particular. (default =0).</p> <p>–<b>1</b> Ningún sondeo de implicaciones lógicas</p> <p><b>0</b> Automático</p> <p><b>1</b> Sondeo limitado</p> <p><b>2</b> Más sondeo</p> <p><b>3</b> Sondeo lleno</p>
SOS1	<p>(Por sus siglas en inglés Special Order Set) Esta opción puede dar o no a cada conjunto un orden especial, permite ramificar sobre una restricción, por lo cual es de esperarse que al activarla el problema converga más rápido a la solución deseada.</p> <p><b>0</b> No utilizar conjuntos ordenados especiales.</p> <p><b>1</b> Utilizar conjuntos ordenados especiales.</p>
varsel	<p>Esta opción se utiliza para fijar la regla de selección de la variable de ramificación en el nodo que se ha seleccionado para ramificar. El valor por default de 0 permite que CPLEX seleccione la mejor regla basada en el problema y su progreso.</p> <p>–<b>1</b> Ramificar sobre variables con infactibilidad mínima. Esta regla puede conducir más rápidamente a una primera solución factible entera, pero usualmente es lento para alcanzar la solución óptima entera.</p> <p><b>0</b> Ramificación de la variable automáticamente seleccionada.</p> <p><b>1</b> Ramificación sobre variables con infactibilidad máxima. Esta regla obliga cambios más grandes en el árbol anterior, para alcanzar en tiempos rápidos la solución óptima.</p>



	<p><b>2</b> Ramificación basada sobre pseudo-costos. Generalmente, el pseudo-costo es más efectivo cuando el problema contiene desventajas complejas y el valor dual tiene una interpretación económica.</p> <p><b>3</b> Ramificación fuerte. Este ajuste causa la selección de variable basada en resolver parcialmente un número de sub-problemas con ramas tentativas para ver que rama es la más prometedora. Esto es a menudo eficaz en problemas grandes y difíciles.</p> <p><b>4</b> Ramificación basada en pseudo-costos reducidos. Computacionalmente menos intensiva que 2</p>
--	--

**Tabla B.1:** Algunas opciones de CPLEX

Para el estudio se procedió a crear un archivo llamado `cplex.opt` (ver Tabla B.2), en donde están algunos parámetros (con su respectiva descripción y valor) que se desean analizar. En el código fuente de GAMS debe ir la instrucción `nombremodelo.OptFile = 1;` para indicarle a CPLEX que debe leer el archivo `cplex.opt` antes de que el método comience a trabajar, por ejemplo, `tdp_b2.OptFile = 1;` (ver Tabla A.1).

Un ejemplo de un archivo `cplex.opt` se presenta a continuación. Los renglones que comienzan con `*` se consideran comentarios y al momento de que CPLEX lo lee no los considera.

```

*epgap es la tolerancia relativa sobre el gap entre el mejor objetivo
*entero y el objetivo del mejor nodo restante. Cuando el valor cae abajo
*de el valor fijo de epgap, la optimización entera mixta es parada.
*Anula la opción OptCR de GAMS que proporciona estos valores iniciales.
*(default = GAMS OptCR)

epgap .01

```

\*mipordind se activa para poder utilizar prioridades, estas se deben  
\*asignar basadas en el conocimiento del problema. Las prioridades se  
\*asignan a las variables discretas usando el sufijo .prior en el  
\*modelo GAMS, bajos valores .prior significan mayor prioridad. El sufijo  
\*.prioropt tiene que ser utilizado para señalar a GAMS y exportar  
\*las prioridades al solver.(default = 1)  
\*0 no usar prioridades para ramificacion  
\*1 utilizar el orden de las prioridades

mipordind 1

\*probe determina la cantidad de sondeos de implicaciones lógicas (probing)  
\*realizados sobre un PEM. El sondeo puede ser muy poderoso y consumir  
\*mucho tiempo. Fijar a 1 puede causar reducciones dramáticas o aumentos  
\*dramáticos en el tiempo de solución dependiendo del modelo particular.  
\*(default = 0)  
\* -1 ningún sondeo  
\* 0 automático  
\* 1 sondeo limitado  
\* 2 más sondeo  
\* 3 sondeo lleno

probe 0

\*varsel fija la regla para seleccionar la variable de ramificación  
\*en el nodo que se ha seleccionado para ramificar. El valor default  
\*es 0 permite que CPLEX seleccione la mejor regla basado sobre el  
\*problema y su progreso.(default = 0)  
\*-1 Ramificar sobre variables con infactibilidad mínima  
\* 0 Ramificación de la variable automáticamente seleccionada

<ul style="list-style-type: none"><li>* 1 Ramificación sobre variables con infeasibilidad máxima</li><li>* 2 Ramificación basada sobre pseudo-costos</li><li>* 3 Ramificación fuerte</li><li>* 4 Ramificación basada en pseudo-costos reducidos</li></ul> <p>varsel 3</p>
---

**Tabla B.2:** Archivo cplex.opt

# Bibliografía

- [1] M. Albareda Sambola, J. A. Díaz y E. Fernández. Lagrangean duals and exact solution to the capacitated  $p$ -center problem. Reporte técnico, Departamento de Estadística e Investigación Operativa, Universidad Politécnica de Cataluña, Barcelona, España, Junio 2006.
- [2] M. S. Daskin. *Network and Discrete Location. Models, Algorithms and Applications*. Wiley, New York, EUA, 1995.
- [3] A. Díaz, F. Glover, H. M. Ghaziri, J. L. González, M. Laguna, P. Moscato y F. T. Tseng. *Optimización Heurística y Redes Neuronales*. Paraninfo, Madrid, España, 1996.
- [4] R. Diestel. *Graph Theory*. Springer-Verlag, Heidelberg, Alemania, 2005.
- [5] B. Fleischmann y J. N. Paraschis. Solving a large scale districting problem: A case report. *Computers & Operations Research*, 15(6):521–533, 1988.
- [6] J. E. Freund, I. Miller y M. Miller. *Estadística Matemática con Aplicaciones*. Prentice Hall, Naucalpan de Juárez, Estado de México, sexta edición, 2000.
- [7] GAMS Development Corporation, Washington, EUA. *GAMS. The Solver Manuals*, 2005.
- [8] S. I. Caballero Hernández, R. Z. Ríos-Mercado y F. López. Solución heurística a un problema de diseño de territorios comerciales con restricciones de asignación

- conjunta mediante grasp. En *Actas de las I Jornadas sobre Algoritmos Evolutivos y Metaheurísticas (JAEM'07)*, pp. 145–153, Zaragoza, España, Septiembre 2007.
- [9] S. W. Hess y S. A. Samuels. Experiences with a sales districting model: Criteria and implementation. *Management Science*, 18(4):998–1006, 1971.
- [10] F. S. Hillier y G. J. Lieberman. *Introducción a la Investigación de Operaciones*. McGraw-Hill, México, DF, 1997.
- [11] J. Kalcsics, S. Nickel y M. Schröder. Toward a unified territorial design approach: Applications, algorithms, and GIS integration. *Top*, 13(1):1–56, 2005.
- [12] O. Kariv y S. L. Hakimi. An algorithmic approach to network location problems. I: The  $p$ -centers. *SIAM Journal on Applied Mathematics*, 37(3):513–538, 1979.
- [13] R. Martí. Algoritmos heurísticos en optimización combinatoria. Departamento de Estadística e Investigación Operativa, Universidad de Valencia. Valencia, España, Junio 2005.
- [14] Minitab Inc., EUA. *Meet MINITAB. Release 14 for Windows*, Septiembre 2003.
- [15] P. B. Mirchandani y R. L. Francis, editores. *Discrete Location Theory*. Wiley, New York, EUA, 1990.
- [16] R. Z. Ríos-Mercado y E. Fernández. A reactive GRASP for a commercial territory design problem with multiple balancing requirements. *Computers & Operations Research*. Aceptado.
- [17] M. W. P. Savelsbergh. Preprocessing and probing techniques for mixed integer programming problems. *ORSA Journal on Computing*, 6(4):445–454, 1994.
- [18] J. A. Segura-Ramiro, R. Z. Ríos-Mercado, A. M. Álvarez-Socarrás y Karim de Alba Romenus. A location-allocation heuristic for a territory design problem in a beverage distribution firm. En *Proceedings of the 12th Annual International Conference on Industrial Engineering Theory, Applications, and Practice (IJIE'07)*, pp. 428–434, ISBN: 978-0-9654506-3-8, Cancún, México, Noviembre 2007.

- 
- [19] L. Vargas-Suárez, R. Z. Ríos-Mercado y F. López. Usando GRASP para resolver un problema de definición de territorios de atención comercial. M. G. Arenas, F. Herrera, M. Lozano, J. J. Merelo, G. Romero y A. M. Sánchez, editores. En *Proceedings of the IV Spanish Conference on Metaheuristics, Evolutionary and Bioinspired Algorithms (MAEB)*, pp. 609–617, Granada, España, Septiembre 2005.
- [20] L. G. Vargas Suárez. Un procedimiento de búsqueda miope, adaptativa y aleatorizada para la definición de territorios de atención comercial. Universidad Autónoma de Nuevo León, San Nicolás de los Garza, NL, Junio 2005.