Universidad Autónoma de Nuevo León

Facultad de Ingeniería Mecánica y Eléctrica
División de Estudios de Posgrado

Un Algoritmo de Localización-Asignación para el Diseño Eficiente de Planes Territoriales de Uso Comercial

Tesis presentada por

José Angel Segura Ramiro

En opción al grado de

Maestro en Ciencias en Ingeniería de Sistemas

Junio de 2008

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN

FACULTAD DE INGENIERÍA MECÁNICA Y ELÉCTRICA
DIVISIÓN DE ESTUDIOS DE POSGRADO

UN ALGORITMO DE LOCALIZACIÓN-ASIGNACIÓN PARA EL DISEÑO EFICIENTE DE PLANES TERRITORIALES DE USO COMERCIAL

TESIS PRESENTADA POR

JOSÉ ANGEL SEGURA RAMIRO

EN OPCIÓN AL GRADO DE
MAESTRO EN CIENCIAS EN INGENIERÍA EN SISTEMAS

JUNIO DE 2008

Universidad Autónoma de Nuevo León

Facultad de Ingeniería Mecánica y Eléctrica

División de Estudios de Posgrado

Los miembros del Comité de Tesis recomendamos que la Tesis "UN ALGORITMO DE LOCALIZACIÓN-ASIGNACIÓN PARA EL DISEÑO EFICIENTE DE PLANES TERRITORIALES DE USO COMERCIAL", realizada por el alumno JOSÉ ANGEL SEGURA RAMIRO, matrícula 1058159, sea aceptada para su defensa como opción al grado de Maestro en Ciencias en Ingeniería de Sistemas.

El Comité de tesis

Dr. Roger Z. Ríos Mercado
Asesor

Dra. Ada M. Álvarez Socarrás
Revisor

Dr. Karim de Alba Romenus
Revisor

Vo. Bo.

Dr. Guadalupe Alan Castillo Rodrguez
Subdirector
División de Estudios de Posgrado

DEDICATORIA

Quiero dedicar este trabajo a las personas que siempre han estado conmigo y que sin su confianza y apoyo incondicional no hubiera sido posible llegar hasta este momento.

A mis padres,

El Sr. J. Florentino Segura Rodríguez *y*

la Sra. Margarita Ramiro Santos.

Agradecimientos

Agradezco primeramente a la coordinación del PISIS por darme la oportunidad de ingresar al programa. A la Universidad Autónoma de Nuevo León y a la Facultad de Ingeniería Mecánica y Eléctrica por la beca de inscripción y colegiatura, respectivamente. Además, agradezco al Consejo Nacional de Ciencia y Tecnología (CONACYT) por otorgarme una beca de manutención. Sin estos apoyos no hubiera sido posible la realización de mis estudios de maestría.

Agradezco también los apoyos otorgados por el CONACYT, por medio del proyecto SEP-CONACYT 48499-Y, y a la Universidad Autónoma de Nuevo León, mediante el proyecto PAICYT CA1478-07, los cuales hicieron posible la exposición de este trabajo en varios foros nacionales e internacionales.

Estoy muy agradecido con el Dr. Roger Z. Ríos, mi asesor de tesis, por su apoyo, consejos y toda su ayuda durante la realización de este trabajo. Asimismo, agradezco también a mis revisores de tesis, la Dra. Ada M. Álvarez y el Dr. Karim de Alba por sus valiosos comentarios que ayudaron en el desarrollo de este trabajo.

Igualmente, deseo agradecer a mis compañeros y amigos, por brindarme su amistad y apoyo a lo largo de esta cruzada.

RESUMEN

El desarrollo del presente trabajo se enfoca en un problema de diseño territorial proveniente de una empresa de bebidas ubicada en la ciudad de Monterrey, Nuevo León. La empresa desea encontrar una partición de un conjunto de unidades geográficas dentro de un determinado número de territorios de manera que una medida de dispersidad en los territorios formados sea mínima. Además, se requiere que los territorios sean balanceados con respecto a dos diferentes medidas de actividades en las unidades básicas (número de clientes y demanda de producto) y que los territorios sean contiguos. Debido a su complejidad, este tipo de problemas no se puede resolver por algoritmos exactos en un tiempo razonable, especialmente para problemas de tamaño real.

Debido a lo anterior, se ha desarrollado una métodología heurística basada en una técnica denominada Localización-Asignación para solucionar este problema. Esta técnica consiste en un proceso iterativo donde primeramente se localizan centros que formarán territorios y después se asignan las unidades básicas a dichos centros. En la literatura revisada, se ha visto que esta técnica ha sido utilizada con éxito para problemas de diseño territorial con una restricción de balance. En el presente trabajo, se extiende esta técnica para poder manejar múltiples restricciones de balance además de restricciones de contigüidad simultáneamente. Además se aplica una técnica de búsqueda local para mejorar la calidad de las soluciones encontradas.

El análisis del problema de la empresa, así como el estudio computacional

que se realizó en este trabajo nos ha permitido lograr un perfecto entendimiento del problema. Asimismo, se han identificado los elementos del problema que más influyen en su complejidad. La metodología desarrollada ha demostrado ser capaz de obtener soluciones de buena calidad en tiempos razonables, inclusive en escenarios difíciles.

Índice general

Índice de figuras

Índice de tablas

Capítulo 1

Introducción

Actualmente las empresas requieren estar al pendiente acerca de cómo satisfacer las necesidades de sus clientes. Sin embargo, muchas veces la manera en que una empresa realiza las actividades necesarias para lograr este objetivo no es la mejor. Para el éxito de una empresa es preciso llegar a dicho objetivo de forma que los gastos asociados a ello sean mínimos.

El problema que se aborda en esta tesis proviene de una empresa embotelladora de bebidas ubicada en la ciudad de Monterrey, N. L, México. Esta empresa necesita dividir el total de sus clientes ubicados en el área de toda la ciudad dentro de un determinado número de territorios de atención comercial de acuerdo a ciertos requerimientos. Este problema, debido a sus características, se clasifica en la categoría de problemas de diseño territorial.

1.1. Descripción del Problema

La ciudad de Monterrey ha sido nombrada como uno de los consumidores más grandes de refrescos a nivel mundial [31], por lo tanto, la demanda de producto y por ende la cantidad de puntos de venta de la empresa en cuestión, es considerable. Debido a esto surge la necesidad de agrupar los puntos de venta o clientes en territorios de manera que se faciliten las tareas administrativas que realiza la empresa. La empresa busca dividir el conjunto de puntos de venta que están distribuidos en el área de la ciudad dentro de territorios utilizando criterios económicos y geográficos bien definidos. La finalidad es tener una apropiada administración de los puntos de venta, además de realizar un adecuado suministro de la mercancía.

La empresa considera una manzana geográfica como unidad básica para formar los territorios. Cada unidad básica posee una cantidad de clientes pertenecientes a la manzana representada, así como una demanda igual a la suma de las demandas de los clientes ubicados en dicha manzana. Sin embargo, debido al tamaño de la ciudad, la cantidad de manzanas geográficas a tratar es extremadamente grande, por lo tanto la empresa realiza un agrupamiento previo donde varias manzanas son consideradas como una sola unidad básica y las actividades relacionadas con la cantidad de clientes y demanda son la suma de todas las manzanas geográficas consideradas dentro de la unidad básica.

En específico, la empresa desea encontrar territorios que sean balanceados respecto a dos medidas que son asociadas a las unidades básicas: número de clientes y volumen de ventas. El principal propósito de este balance es equilibrar la carga de trabajo que tiene la gente encargada del abastecimiento de los puntos de venta así como también la administración de órdenes de compra en los distintos territorios. Otra consideración importante es la geografía de la ciudad ya que los territorios deben estar formados por unidades básicas que sean alcanzables entre sí dentro del mismo territorio. La finalidad aquí es que en una posterior fase de ruteo sea posible viajar entre unidades del mismo territorio sin salir del mismo. Esto evita situaciones perjudiciales a la empresa en las que un cliente que necesita mercancía pueda ver pasar algún camión repartidor que no lo atiende debido a que dicho cliente no pertenece al territorio asignado al camión repartidor. Además, la empresa requiere que los territorios formados sean lo más compactos posible, es decir, que las unidades básicas que pertenecen al mismo territorio se encuentren relativamente cerca entre sí. Por último, se desea construir un número específico de territorios.

Justamente, la dificultad del problema radica en encontrar el mejor agrupamiento de las unidades básicas, de tal manera que los territorios formados cumplan con todos los requerimientos antes mencionados.

1.2. Motivación

Los problemas de diseño territorial han sido ampliamente estudiados; sin embargo, en la literatura se observa que cada caso tiene sus particularidades que hacen difícil su tratamiento de forma general, por lo que en ocasiones es necesario construir herramientas especializadas para cada caso en particular. Cabe mencionar que el problema descrito en la sección anterior contiene características que no han sido tratadas previamente en la literatura especializada.

Se sabe además que debido a la naturaleza del problema, es posible formularlo como un problema lineal entero mixto que puede ser resuelto con paquetes de optimización disponibles en el mercado. Sin embargo, el tamaño de las instancias reales es tal que los algoritmos exactos no pueden dar soluciones óptimas en un período razonable de tiempo. Es por eso que para obtener soluciones satisfactorias para instancias reales es necesaria una herramienta computacional que proporcione soluciones que pueden no llegar a ser óptimas pero aún así son de alta calidad y obtenidas en un tiempo razonable.

Las razones expuestas previamente justifican la aplicación de algoritmos aproximados de solución. En la literatura se ha estudiado un método aproximado basado en una técnica denominada Localización-Asignación. Dicha técnica consiste de dos procesos iterativos donde primero son elegidos centros que representan territorios y después las unidades básicas son asignadas a estos centros.

Esta metodología ha sido utilizada con relativo éxito para problemas de diseño territorial que involucran una sola restricción de balance. En el presente trabajo se extiende esta metodología y se propone una adaptación para manejar múltiples restricciones de balance además de restricciones de conectividad simultáneamente. Cabe destacar también que este tipo de estudio no ha sido reportado en la literatura hasta donde se tiene conocimiento.

1.3. Objetivos y Alcance

El proyecto de tesis contempla diseñar y construir una herramienta computacional basada en la técnica de Localización-Asignación que permita encontrar soluciones adecuadas en un tiempo razonable para el problema descrito anteriormente. Para esto se consideran los siguientes objetivos particulares:

1. Efectuar un análisis detallado de datos, requerimientos, restricciones, condiciones y objetivos mencionados por la empresa con la finalidad de lograr un completo conocimiento del problema.

2. Estudiar la estructura matemática del problema, es decir, la formulación del modelo matemático que representa el problema, con el fin de encontrar soluciones exactas al problema y encontrar el tamaño máximo al que es posible encontrar solución exacta.

3. Desarrollar e implementar la metodología de solución basada en la técnica de Localización-Asignación que permita encontrar soluciones al problema de diseño de territorios.

4. Realizar un amplio estudio de la metodología desarrollada con el fin de lograr un conocimiento del comportamiento de dicha metodología bajo diferentes condiciones de instancias del problema.

1.4. Justificación

El desarrollo de una metodología basada en la técnica de Localización-Asignación tiene su fundamento en el éxito que ha reportado en aplicaciones similares del diseño de territorios. Sin embargo, esta técnica está diseñada para manejar una sola restricción de balance. Por otra parte, la restricción de contigüidad o conexidad tampoco es tomada en cuenta en publicaciones acerca de esta metodología. El trabajo en esta tesis propone una metodología que logra una adaptación para manejar múltiples restricciones de balance y además restricciones de contigüidad simultáneamente.

1.5. Hipótesis

En base al conocimiento adquirido a través de la revisión de literatura, conocimiento del problema y estudio de técnicas de solución a problemas similares, se ha desarrollado una metodología basada en la técnica llamada Localización-Asignación. La hipótesis es que este método propuesto es capaz de proporcionar soluciones de buena calidad en tiempos razonables y que satisfacen todos los requerimientos del problema en cuestión.

1.6. Estructura de la Tesis

En el presente capítulo se presentó la descripción general del problema de diseño de territorios tratado en la tesis. Se planteó además el objetivo de la tesis, la justificación e hipótesis de la misma. En el Capítulo 2 se realiza una revisión de los antecedentes del problema de diseño de territorios. Se explican además algunas técnicas de solución que han reportado buenos resultados en este tipo de problemas. En el Capítulo 3 se explica detalladamente el modelo matemático para representar el problema a resolver así como sus supuestos, notaciones y su complejidad computacional. En el Capítulo 4 se explica de manera detallada el algoritmo desarrollado para encontrar soluciones al problema de diseño de territorios. Se profundiza en cada fase del algoritmo explicando además su complejidad y la forma en que opera. En el Capítulo 5 se realiza el estudio empírico del algoritmo y se describe la serie de experimentos que se ejecutaron para probar el desempeño del mismo. También se describen las especificaciones de las instancias utilizadas para la evaluación, así como la forma de generarlas. Se explican también las características y parámetros utilizados en las instancias además de los aspectos a evaluar del algoritmo. Además, se presentan los resultados obtenidos a partir de los experimentos realizados. Se formula un grupo de hipótesis en relación al efecto de ciertas características del problema sobre el desempeño del algoritmo y se realiza un análisis estadístico de resultados. Para finalizar, el Capítulo 6 contiene las conclusiones, aportaciones y posible trabajo futuro.

CAPÍTULO 2

DISEÑO DE TERRITORIOS Y SUS ANTECEDENTES

El problema de diseño de territorios ha sido ampliamente estudiado tanto por gente especializada en el área de investigación de operaciones así como por diversos especialistas, tales como geógrafos, políticos, urbanistas o expertos en mercadotecnia, en áreas donde suelen surgir problemas de diseño de territorios.

La finalidad del presente capítulo es ofrecer un panorama general sobre las distintas aplicaciones que tienen los problemas de diseño de territorios en el mundo real. Se presenta el caso general de un problema de diseño de territorios y son presentadas algunas particularidades y características que sobresalen en los distintos tipos de aplicaciones.

Además, se explican algunas de las diversas técnicas que se han encontrado en la literatura y que ofrecen soluciones para este tipo de problemas.

2.1. CRITERIOS TOMADOS EN CUENTA PARA EL DISEÑO DE TERRITORIOS

Primeramente, necesitamos conocer los distintos criterios que pueden influir en la forma de diseñar territorios. A continuación se enlistan los más comunes que se han encontrado en la literatura.

2.1.1. Criterios Geográficos

Compacidad. Se busca que las unidades básicas que se encuentran en un territorio se encuentren relativamente cerca entre sí. Este criterio en la mayoría de los casos es de vital importancia. Aunque no existe la definición rigurosa de compacidad, existen diferentes medidas que ayudan a representarla. La manera más fácil de notar la importancia es pensando en recorrer un territorio. Obteniendo territorios compactos se garantizaría que los costos de transporte sean mínimos. En el caso de territorios políticos, la compacidad proviene de un requerimiento para tener una mejor organización.

Contigüidad. Los territorios deben ser geográficamente conexos. Esto para que sea posible viajar dentro de los mismos sin salir del territorio.

Accesibilidad. Este tipo de restricciones tiene que ver con garantizar la accesibilidad y evitar obstáculos como montañas, ríos, etc.

2.1.2. Criterios Demográficos

Población. Este criterio es exclusivo del diseño de territorios políticos. Se requiere la igualdad de votantes para conservar el principio de equidad. Aunque es imposible lograr la misma cantidad de votantes en cada territorio se utiliza una tolerancia de desviación. Esta tolerancia varía de país a país.

Representación Minoritaria. Este criterio es utilizado en diseño de territorios políticos. La intención de este criterio es asegurarse que los votantes minoritarios tengan la misma oportunidad de participar en el proceso político.

2.1.3. Criterios Organizacionales

Número de territorios. Este número suele ser predeterminado por la fuerza de ventas o incluso designado por algún asesor de la compañía en caso de diseño de territorios de ventas. En el caso de territorios políticos este número está dado por la cantidad de representantes políticos que sea necesario elegir. En algunos trabajos, este número de territorios a formar es considerado como una variable a optimizar.

Áreas básicas. Un área básica no siempre es un simple cliente. De hecho en la práctica lo más común es que los clientes se agrupen por colonia, códigos postales o incluso ciudades formando un área básica. Dependiendo del nivel que se elija puede dificultarse el problema o inclusive las soluciones pueden no ser representativas de la realidad.

Asignación exclusiva de áreas básicas. En la práctica algunas áreas básicas requieren ser parte de un determinado territorio ya sea por política de la empresa en el caso de territorios de ventas o por algún acuerdo en el caso de territorios políticos. También incluso hay quienes manejan restricciones de asignación conjunta, es decir, que algunas unidades básicas forzosamente necesitan ser asignadas al mismo territorio.

Localización de representantes de ventas. Este criterio es exclusivo del diseño de territorios de ventas. Debido a que los representantes de venta suelen tener que visitar seguido a sus clientes, en ocasiones es necesario diseñar los territorios pensando por ejemplo en minimizar costos de transporte.

2.1.4. Criterios basados en actividades

Balance. Este criterio es comúnmente utilizado en diseño de territorios de ventas. Muchas veces es necesario que actividades que están relacionadas con las unidades básicas tengan un balance en los distintos territorios. El motivo principal es para lograr un equilibrio en la carga de trabajo o en la cantidad de clientes que es necesario administrar por territorio.

2.1.5. Criterios Políticos

Datos políticos. Este criterio no es utilizado en un primer diseño de territorios. Sin embargo, para un posterior rediseño de territorios algunos datos políticos generados en un período de tiempo pueden ser tomados en cuenta para un rediseño de territorios.

2.2. Diseño de Territorios

El diseño de territorios puede ser visto como la acción de agrupar un conjunto de unidades básicas de una región geográfica en territorios o agrupaciones con propósitos ya sea organizacionales, administrativos o comerciales. La aplicación de este tipo de problemas es muy amplia, algunas de las más conocidas son el diseño de distritos políticos, el diseño de territorios de ventas y el diseño de zonas de servicios (emergencias, escolares, de salud, etc.)

Sin embargo, la idea general en cada caso es prácticamente la misma: el conjunto de unidades básicas son agregadas a distritos o territorios. Comúnmente primero son especificados los centros (pueden ser incluso ficticios) para representar un territorio y después se asignan las unidades básicas a estos centros (territorios) tratando de optimizar algún objetivo y cumpliendo un conjunto de restricciones preestablecidas.

El problema entonces se puede definir como encontrar una partición de n unidades básicas en p subconjuntos, donde p es el número de territorios que es necesario formar, respetando un conjunto de restricciones tales como contigüidad, compacidad, balance, etc.

El problema es generalmente reconocido como un problema de optimización combinatoria y tiene semejanzas con el problema de agrupamiento y con el de partición de conjuntos. Estos dos problemas han sido clasificados como NP-duros (Garey y Johnson [17]). Es decir, que el problema es extremadamente difícil de resolver en forma exacta para instancias suficientemente grandes. En la Sección 3.6 se realiza un estudio sobre el tamaño de las instancias que podemos resolver de forma exacta para el problema abordado en esta tesis. Por esta razón en la literatura la mayoría de las propuestas para solucionar este tipo de problema son algoritmos heurísticos que buscan soluciones de buena calidad en un tiempo razonable. En las siguientes subsecciones se describen algunos de los trabajos más destacados en las distintas categorías de diseño territorial.

2.2.1. Diseño de Territorios de Atención Comercial

El diseño de territorios de atención comercial surge en el momento en que se necesita realizar una agrupación de clientes para poder atender inteligentemente a los clientes de manera eficaz. El problema que se aborda en este trabajo de tesis cae en esta categoría y tiene algunos antecedentes que es necesario mencionar.

Es necesario destacar que los trabajos desarrollados hasta el momento en este tipo de aplicaciones han consistido de implementaciones del método GRASP, una metaheurística muy bien conocida que combina características de algoritmos voraces y procedimientos de construcción aleatoria. Esta metaheurística ha sido ampliamente utilizada en una buena cantidad de problemas de optimización combinatoria y fue introducida por Feo y Resende [11]. GRASP es un proceso iterativo en el cual cada iteración consiste básicamente en dos fases: construcción y post-procesamiento. En la fase de construcción se construye una solución factible, mientras que el post-procesamiento se encarga de mejorarla. Si una solución factible es encontrada en la fase uno, la fase dos se puede considerar como una búsqueda local simple.

La primera aproximación para este tipo de problema fue desarrollada por Vargas-Suárez [31] y Vargas-Suárez, Ríos-Mercado y López [32]. El problema estudiado en este trabajo era similar al estudiado en esta tesis. Sin embargo, en este problema no se contaba con el requerimiento de compacidad territorial. Además, en este trabajo se manejaron tres criterios de balance. Por lo tanto, el principal objetivo de la tesis fue satifacer la restricción de balance en los territorios. Se desarrolló un modelo de programación entero mixto en el cual se minimizaba el desbalance. Sin embargo, este modelo no podía solucionarse de manera óptima para instancias grandes. Así que se implementó una metodología basada en GRASP para obtener soluciones al problema. En este trabajo se reportan soluciones de buena calidad para instancias de hasta 500 unidades básicas.

La metodología de GRASP fue también utilizada por Ríos-Mercado y Fernández [28]. En este trabajo se trata un problema con prácticamente las mismas características del problema abordado en esta tesis. Sin embargo, se utiliza una medida de dispersión diferente para la función objetivo. En este trabajo se realizan dos implementaciones del método GRASP diferenciados por una estrategia de reactividad. La forma de construcción de territorios en la implementación consigue garantizar el cumplimiento de las restricciones de conexidad. La función miope o voraz utilizada consiste en una combinación de la función de dispersión con una función de infactibilidad en balanceo. Además, se emplea una búsqueda local simple para mejorar la calidad de la solución. En este trabajo se lograron encontrar soluciones de buena calidad para instancias de hasta 1000 unidades básicas. Este estudio fue posteriormente extendido [27] para instancias de gran escala logrando encontrar siempre soluciones factibles de hasta 2000 nodos y con una tolerancia de 3 %.

Más recientemente, en los trabajos de Caballero-Hernández et al. [5, 6] se aborda un problema de similares características al presentado por Ríos-Mercado y Fernández [28] con restricciones adicionales de asignación conjunta. La metodología propuesta consiste en la implementación de una metodología GRASP combinada con dos técnicas de búsqueda local. La forma de contrucción es diferente a la presentada por Ríos-Mercado y Fernández, y reporta buena calidad en las soluciones obtenidas. En este trabajo se lograron resolver instancias de hasta 2000 unidades básicas.

2.2.2. Diseño de Territorios Políticos

El problema de diseño de territorios políticos proviene de tratar de particionar algún área gubernamental como una ciudad o un estado en subáreas en las cuales son elegidos candidatos políticos para representar dicha área o subárea. El problema radica en el rediseño de territorios ya que debe ser realizado cada determinado período. Estos períodos varían según el país. Existen estudios de diseño de territorios políticos. Algunos muy completos y de los más recientes fueron publicados por Williams [33] y Grilli di Cortona et al. [9].

Entre los primeros métodos de aproximación desarrollados para resolver este tipo de problemas, se puede mencionar la estrategia de dicotomías de Forrest [15], en la cual se toman en cuenta algunas consideraciones políticas para crear distritos políticos con una cantidad similar de votantes. La idea de este tipo de métodos es realizar particiones iterativamente sobre la región en la que se desea diseñar territorios, esto para obtener cada vez subproblemas más pequeños. Cada subproblema es definido por un determinado conjunto de áreas básicas que caen dentro de las regiones particionadas. Las iteraciones del algoritmo terminan cuando se alcanza un nivel en el que el problema de diseño territorial para cada partición es fácil de resolver. Sin embargo, en trabajo no se mencionan los detalles de la forma en que se realizan estas particiones para simplificar el problema.

Otra de las técnicas utilizadas para resolver este tipo de algoritmo son los algoritmos genéticos, este tipo de algoritmos fueron aplicados a los problemas de diseño territorial por Forman and Yue [14]. Estos autores utilizan una técnica basada en una codificación y en operadores genéticos usados para resolver problemas del agente viajero. La codificación elegida es una representación de una ruta y un solo cromosoma viaja a través de cada área básica. Como las áreas son atravesadas, los territorios resultantes son formados por la secuencia de áreas básicas.

Bozkaya et al. [3] soluciona un problema de distritos políticos sujeto a restricciones de contigüidad, igualdad de población, compacidad y homogeneidad económica. Ellos desarrollaron un proceso de búsqueda tabú y lo usaron para encontrar soluciones a un caso real en Edmonton, Canadá, con 828 unidades básicas en 19 distritos. Dicha técnica integra varios de los criterios en la función objetivo. Sus resultados indican que el algoritmo produce mejores soluciones a las existentes, además se logra una notable mejora en satisfacer las restricciones.

2.2.3. Diseño de Territorios de Ventas y Servicios

La mayoría de las compañías que tienen una venta considerable y una gran área de mercado se ven en la imperiosa necesidad de diseñar territorios de ventas. Esto ya sea con fines meramente administrativos o comerciales. El problema de diseño de territorios de servicio es prácticamente muy similar, donde la finalidad es atender a los clientes de la mejor manera posible tratando de minimizar ciertos costos. En general, pueden existir una gran cantidad de razones por las que es necesario rediseñar un área de ventas o servicios. Puede ser por un incremento o decremento en el número de ventas o clientes, lo que hace necesario un ajuste en el diseño territorial. Algunas otras razones pueden ser para administrar el personal de atención a las áreas de una mejor manera o balancear la carga de trabajo en los territorios. En Zoltners y Sinha [34] se puede ver un estudio de modelos de diseño de territorios de ventas.

Fleischmann y Paraschis [13] realizan un estudio de una compañía manufacturera alemana en la cual los territorios de ventas eran muy desiguales en la carga de trabajo, por lo que llevan a cabo un rediseño de territorios. Ellos resuelven este diseño de territorios a través de una formulación de un problema entero mixto y desarrollan una aproximación basada en la técnica de Localización-Asignación.

2.2.4. Otras Aplicaciones

Aunque la mayoría de los casos de diseño territorial caen en la categoría de política o ventas, hay algunos artículos que realizan estudios sobre otros tipos de diseño territorial.

Blais et al. [2] estudian un problema de diseño de territorios de atención para la salud en donde un cierto personal de atención a pacientes, tales como enfermeras o fisioterapistas, atiende una determinada área. Cuando el área es muy grande es necesario particionarla para que sea debidamente atendida.

En el trabajo de Ferland y Guénette [12] se describe el problema del diseño de territorios escolares. Los criterios principales para la toma de decisión en cuanto a la creación de territorios viene a darse por las limitaciones de capacidad en las escuelas, la utilización igualitaria de las escuelas, las distancias de viaje de los alumnos, la buena accesibilidad y el balance racial. Caro et al. [7] realizan un estudio de aplicaciones de diseño de territorios de escuelas.

Amico et al. [1] presentan un algoritmo de recocido simulado para el problema de rediseño de territorios para policía. Ellos modelan el problema como una partición de grafos que incluye la partición de la judisdicción de policia en distritos sujetos a restricciones de contigüidad, compacidad, convexidad y tamaño. Ellos probaron su método en un caso de estudio en Departamento de Policía de Buffalo, Nueva York.

En otras aplicaciones se usa el diseño territorial para instalaciones que proporcionan servicio en una localización fija, donde es necesario que los clientes acudan a dichas instalaciones para que sean servidos. Como ejemplos de este tipo de aplicaciones están el diseño de territorios de hospitales, emergencias, bomberos, etc.

2.3. Métodos de Localización-Asignación

Hay una gran cantidad de métodos aproximados que han aparecido en la literatura. Existen incluso estudios más extensos que hacen comparaciones y evaluaciones de estos métodos. Para referencia puede consultarse Howick y Pidd [20], Ricca y Simeone [26] y Kalcsics et al. [22]. Sin embargo, como la metodología de solución que se ha desarrollado en esta tesis tiene su base en la técnica de Localización-Asignación, en esta sección se describen algunos de los antecedentes de esta técnica de solución.

La técnica de Localización-Asignación fue una de las primeras heurísticas en ser desarrollada para problemas de diseño territorial. Esta técnica fue propuesta por Hess et al. [19] para resolver un problema de diseño de territorios políticos. El problema era un caso común de diseño territorial con una restricción de balance en la que se buscaba minimizar una función objetivo que representaba compacidad en los territorios.

Dicha técnica radica en aprovechar la estructura general de un problema de diseño territorial. Esta metodología consta de dos fases: localización y asignación. En la fase de localización son elegidas ciertas instalaciones de entre la cantidad total de unidades básicas. Estas instalaciones representan centros de territorios potenciales. Después, en una fase de asignación, las unidades básicas restantes se asignan a estas instalaciones para formar así los territorios.

Sin embargo, aún separando las decisiones de localización y asignación se encontró que la asignación es difícil de resolver en instancias grandes. Por lo tanto, la fase de asignación fue resuelta de forma aproximada en vez de una manera exacta, con el fin de garantizar tiempos razonables. No obstante, los primeros resultados con algunas aproximaciones no fueron tan buenos como se esperaba. Sin embargo, tanto en George et al. [18], como en Kalcsics et al. [22] se desarrollaron nuevas técnicas en cuanto a la fase de asignación, con mejores resultados.

En este trabajo de tesis se realiza una adaptación de la técnica Localización-Asignación. En capítulos posteriores, se explicará a detalle en qué consiste cada una de las fases de esta metodología y cómo mejorar la calidad en las soluciones, así como el manejo de restricciones que no se han contemplado en la literatura.

CAPÍTULO 3

ESTRUCTURA MATEMÁTICA DEL PROBLEMA

La programación matemática utiliza modelos para representar problemas extraídos de la realidad, y así poder tratarlos con técnicas de Investigación de Operaciones. En muchos casos el uso de estos modelos se realiza de forma exitosa obteniendo soluciones que se llevan a la práctica. El tema central de este capítulo es la descripción detallada del problema tratado en esta tesis, así como su correspondiente formulación por medio de un modelo de programación matemática. En un principio se abordan las ventajas y desventajas que aporta el desarrollar un modelo para un problema. Después, se explica lo que es un modelo de programación matemática así como los diferentes tipos de modelos existentes. Posteriormente, se procede a definir detalladamente el problema, su representación como un grafo, así como los supuestos, notaciones, parámetros y variables. Después, se formula el modelo de programación entera mixta resultante y se explican detalladamente cada una de sus partes. Por último, se concluye con una discusión de este modelo así como de la alternativa comercial que permite la aplicación de este modelo para resolver pequeñas instancias de este problema.

3.1. Modelos matemáticos

El término modelo suele utilizarse en una gran cantidad de aplicaciones. En términos generales, se refiere a una estructura que posee rasgos y características de un determinado objeto. Comúnmente solo algunas de las características se plasman en el modelo dependiendo del uso que se le quiera dar a éste. En la Investigación de Operaciones los modelos son abstractos, es decir, que se utilizan símbolos algebraicos para representar las relaciones del objeto de estudio que se desea modelar. Algunos ejemplos de objetos a modelar pueden ser: un plan de trabajo, un horario de tareas, un calendario de actividades, una organización, etc.

Un modelo matemático en la Investigación de Operaciones esencialmente se compone de relaciones matemáticas (ecuaciones, desigualdades, dependencias lógicas) que corresponden a determinadas relaciones o características en el mundo real entre los objetos que se están modelando. Dichas relaciones pueden ser relaciones geográficas, restricciones de elección, restricciones de capacidad, prioridades, entre otras. Estas relaciones se utilizan para lograr una mejor modelación de las características y comportamientos que puede tener el objeto.

Las principales ventajas de construir un modelo matemático para representar un problema se enlistan a continuación:

- Permite encontrar relaciones que pueden no ser aparentes. Es decir, se puede lograr un lograr un mejor conocimiento del problema.
- Permite la experimentación al menos en lo concerniente a cuantificar valores de solución dependiendo de las variables de nuestro problema. No se puede experimentar con estos valores directamente en el problema real ya que suele implicar un gran costo.
- Permite una descripción concisa del problema lo cual hace más comprensible la estructura del mismo.

- A partir del modelo se suele ver con claridad los datos que son importantes para el problema.

- El modelo puede ser resuelto en muchos casos con el uso de paquetes computacionales al menos en instancias pequeñas. Esto puede permitir obtener una idea del comportamiento del problema bajo determinadas condiciones.

Sin embargo, existen también algunas desventajas que deben mencionarse. Un modelo es, por definición, una idealización abstracta del problema. Es decir que normalmente se requieren aproximaciones, simplificaciones y suposiciones para que el problema pueda ser modelado y resuelto. Por lo tanto se debe tener cuidado que el modelo sea una representación lo más apegada posible al problema real. Otra desventaja suele ser que algunos modelos pueden resultar demasiado difíciles de resolver. Esto suele deberse al tamaño del problema o a especificaciones del mismo. Incluso un mismo problema puede modelarse de distintas formas y hay que tratar de decidir cuál es la más conveniente según el caso. En los modelos de programación matemática, la definición de la función objetivo influye definitivamente en la respuesta lograda.

Se puede tener como conclusión que la respuesta exitosa de un modelo depende de dos factores principales: la fidelidad del modelo con respecto al objeto que representa y la exactitud de los datos. En la medida que se cuiden estos dos detalles se podrán crear modelos representativos de la realidad para proporcionar buenas soluciones a los problemas.

3.2. Programación Matemática

La programación matemática es algo distinta a la programación computacional a pesar de su nombre. La programación matemática se involucra con la programación computacional cuando se llevan a la práctica las técnicas de programación matemática y es necesario organizar grandes conjuntos de datos, así como realizar los cálculos aritméticos que sólo es posible manejar por medio de una computadora.

Los modelos de programación matemática son necesarios cuando se necesita optimizar alguna medida de desempeño. Es decir, cuando se desea minimizar o maximizar medidas tales como ganancias o costos. Esta medida a optimizar recibe el nombre de función objetivo. Para crear un modelo de programación matemática es necesario conocer: (i) variables, que representan las cantidades que se busca conocer y provocan un impacto en la medida que se desea optimizar, (ii) un conjunto de restricciones que delimitan los valores que estas variables pueden tomar y además representan las relaciones entre el objeto o situación que se está modelando, y (iii) los parámetros establecidos.

Se pueden clasificar los modelos en diferentes categorías dependiendo del tipo de valor que puedan tomar las variables (valores enteros, reales o binarios) del modelo. Además influye también el tipo de función (lineal, no lineal, convexa, etc.) que define a la función objetivo y las restricciones. Combinando el tipo de variable con el tipo de función se pueden definir una gran cantidad de tipos de modelos. Otra característica importante que define el modelo es el conocimiento de los datos. Cuando no se conocen con exactitud estos datos el modelo es considerado estocástico. Si los datos se conocen con certeza, el modelo se considera determinístico. En este caso, el problema se modela como un Programa Entero Mixto Lineal (PEML) ya que las variables de decisión son enteras (binarias) y las funciones son lineales.

3.3. Definición del problema

En la Sección 1.1 se describe el problema de diseño de territorios de ventas. Con base en esta descripción abordada previamente se pueden definir los siguientes supuestos:

- La unidad básica para construir los territorios o grupos es una manzana geográfica.
- Existen dos medidas de actividades asociadas con cada manzana o unidad básica: número de clientes atendidos y volumen de ventas. Estas mismas medidas se pueden calcular para cada territorio y su valor es simplemente la suma aritmética de los valores correspondientes a las unidades básicas que los conforman.

Además, existen restricciones o requerimientos que necesitan ser tomadas en cuenta para obtener un diseño de territorios válido para nuestro problema. Estas restricciones se enlistan a continuación:

- Cada unidad básica debe pertenecer únicamente a un territorio. Es decir que los territorios son disjuntos y la unión de todos los territorios resulta en el conjunto de unidades básicas original.
- Cada territorio debe ser conexo. Un territorio se define como conexo si para cada par de unidades básicas que pertenecen al territorio existe una ruta que las comunique y que dicha ruta solo esté compuesta por unidades básicas que pertenecen al mismo territorio.
- Se desea que las medidas de actividades sean balanceadas en los distintos territorios que se construyan para una solución, para esto se calcula una meta que se desea alcanzar en cada territorio. Esta meta es calculada a partir de la suma aritmética de todos los valores de las unidades básicas entre el

número de territorios a formar (promedio o media aritmética), dicha meta representa el balance perfecto.

- Se desea optimizar la compacidad en los territorios, es decir que el objetivo de este problema es encontrar territorios en los cuales las unidades básicas que los compongan estén relativamente cerca entre sí.

Con base en lo anterior se puede resumir que se busca que los territorios posean tres características fundamentales: deben ser conexos, compactos y balanceados respecto a las medidas de las actividades en cada unidad básica.

3.4. Formalización del modelo como grafo

A continuación se procede a formular un modelo basado en un grafo para el problema de diseño de territorios estudiado en esta tesis. La red de distribución de las unidades básicas se define por un grafo $G = (V, E)$ donde cada unidad básica es representada por un nodo $i \in V$, y una arista $(i, j) \in E$. Una arista (i, j) existe si los nodos i y j se localizan en manzanas adyacentes. Un ejemplo de esta representación se puede observar en la Figura 3.1.

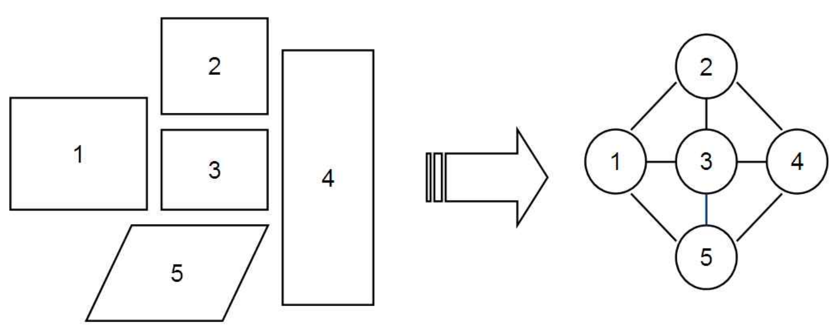

Figura 3.1: Representación de manzanas geográficas como un grafo

Cada nodo en el grafo posee las siguientes propiedades: coordenadas geográficas (c_i^x, c_i^y), y dos medidas de actividades. Esto es, w_i^a representa el valor de la actividad a en el nodo i, donde la primera actividad, $a = 1$, representa el número de clientes dentro de la unidad básica y la segunda actividad, $a = 2$, representa el total de volumen de ventas en la unidad básica. Se define también un parámetro fijo p que representa el número de territorios a formar. A partir de la definición del grafo se define un territorio como un subconjunto de nodos $V_k \subset V$ y se requiere que cada nodo sea asignado sólo a un territorio.

Para poder modelar el requerimiento de balanceo respecto a las actividades en los nodos es primero necesario definir el tamaño de un territorio respecto a la actividad a como sigue: $\omega^a(V_k) = \sum_{i \in V_k} \omega_i^a, a = 1, 2$. Debido a la estructura discreta del problema y a la restricción de asignación única de los nodos es prácticamente imposible lograr territorios perfectamente balanceados respecto a las dos actividades en cada nodo. Para manejar esta dificultad se mide el grado de balanceo calculando la desviación relativa respecto al valor promedio μ^a en la actividad a, que está dado por la fórmula: $\mu^a = \omega^a(V)/p, a = 1, 2$, donde p es, como se mencionó anteriormente, el número de territorios que se desea formar. Se permite entonces una tolerancia de desviación respecto a esta medida, que es definida por el parámetro τ^a. Esto formaría la siguiente restricción: $(1 - \tau^a)\mu^a \leq \omega^a(V_k) \leq (1 + \tau^a)\mu^a, \forall k, a$. En la Figura 3.2 se muestra un ejemplo sobre balanceo en una actividad.

La siguiente característica importante es la conexidad o contigüidad, es decir, que todas las unidades básicas asignadas a cada territorio deben estar conectadas por una ruta que se encuentre contenida totalmente dentro del territorio. En otras palabras, cada territorio V_k debe inducir un subgrafo conexo de G. Como muestra se puede observar en la Figura 3.3 una solución factible e infactible con respecto a este requerimiento. En el teritorio V_1 de la solución factible podemos notar que es posible trazar una ruta de cada nodo a los demás, mientras que en la solución infactible notamos que no podemos trazar rutas a partir del nodo 3 ya

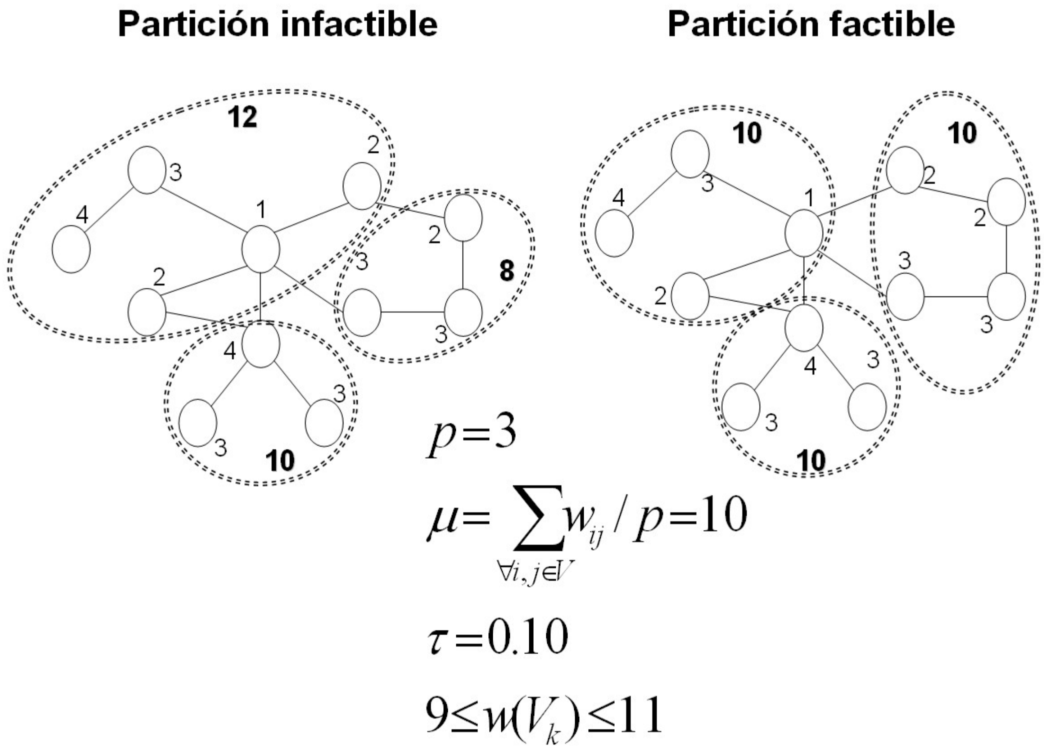

Figura 3.2: Criterio de balanceo

que se encuentra desconectado de los nodos 1 y 2.

Además, se desea que las unidades básicas en cada territorio se encuentren localizadas relativamente cerca entre sí. Una manera de tratar esta parte es definir un nodo que sirva como centro del territorio k, denotado por $c(k)$ y entonces definir la siguiente medida de dispersión: $f = \sum_{i \in V_k} \sum_{k \in K} d_{ic(k)}$, donde d_{ij} representa la distancia euclidiana entre los nodos i y j. Esta medida de dispersión es conocida en la literatura como p-mediana y representa la suma de las distancias euclidianas de cada una de las unidades básicas a su centro. Sea Π la colección de todas las posibles p-particiones de V. La formulación combinatoria del problema consiste en encontrar una partición $V = (V_1, V_2, ..., V_p) \in \Pi$ tal que se minimice f, sujeta a $(1 - \tau^a)\mu^a \leq \omega^a(V_k) \leq (1 + \tau^a)\mu^a, \forall k, a$ y $G_k = (V_k, E(V_k))$ conexo para todo $V_k \in V$.

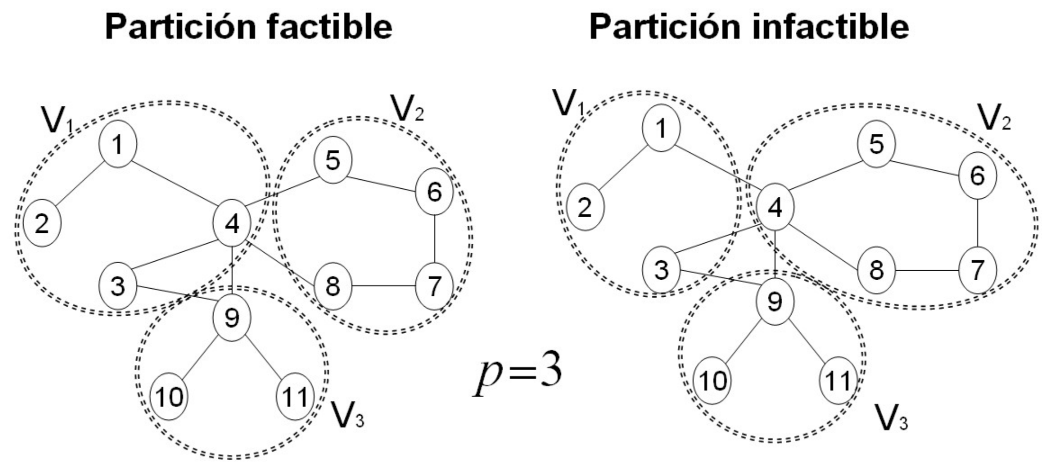

Figura 3.3: Criterio de contigüidad

3.5. Formulación Lineal Entera Mixta

A continuación se presenta una formulación del problema como un programa entero mixto lineal (PEML), este modelo que define el problema abordado en esta tesis se le denominará de aquí en adelante modelo TDP.

Índices y conjuntos

V	conjunto de unidades básicas
E	conjunto de aristas que existen entre las unidades básicas
A	conjunto de actividades en las unidades básicas
K	conjunto de territorios
n	número de manzanas (unidades básicas)
p	número de territorios
i, j	índices de las manzanas; $i, j \in V = \{1, 2, \ldots, n\}$
a	índices de las actividades; $a \in A = 1, 2$
k	índices de territorios; $k \in K = 1, 2, \ldots, p$
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 euclideana entre i y j; $i, j \in V$

τ^a tolerancia relativa 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; $a \in A$, $B \subset V$

μ^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 compacidad.

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

Modelo TDP

$$\begin{array}{lrcll}
\text{Minimizar} & f(x) & = & \sum_{i \in V} \sum_{j \in V} d_{ij} x_{ij} & & (3.1) \\
\text{sujeto a} & \sum_{i \in V} x_{ij} & = & 1 & j \in V & (3.2) \\
& \sum_{i \in V} x_{ii} & = & p & & (3.3) \\
& \sum_{j \in V} w_j^a x_{ij} & \leq & (1 + \tau^a) \mu^a x_{ii} & i \in V, a \in A & (3.4) \\
& \sum_{j \in V} w_j^a x_{ij} & \geq & (1 - \tau^a) \mu^a x_{ii} & i \in V, a \in A & (3.5) \\
& \sum_{j \in \cup_{v \in S} N^v \setminus S} x_{ij} - \sum_{j \in S} x_{ij} & \geq & 1 - |S| & i, j \in V, \forall S \subset V & (3.6) \\
& x_{ij} & \in & \{0, 1\} & i, j \in V & (3.7)
\end{array}$$

En la literatura existen reportes de varias medidas que representan la dispersión de los territorios. Las más utilizadas debido a su éxito son las que están basadas en p-centro y en p-mediana. Owen y Daskin [24] realizan un estudio sobre los modelos que resultan de estas medidas. La principal motivación para la elección de la medida en la función objetivo (3.1) es que además de representar una buena medida de dispersión, se producen soluciones que son más fáciles de manejar por el algoritmo propuesto y así se consigue un mejor desempeño del mismo. Las restricciones (3.2) garantizan la asignación única de unidades básicas a los territorios. Esto es, que cada nodo j es asignado a un territorio solamente. La restricción (3.3) se utiliza para forzar que el número de territorios formados sea igual a p. Nótese que cuando $x_{ii} = 1$ significa que el nodo i es un nodo que representa un centro. La restricciones (3.4) y (3.5) representan el balanceo de las actividades en cada territorio. Es decir que establece que el tamaño del territorio se encuentre dentro de un rango (medido por el parámetro τ^a) alrededor del tamaño promedio μ^a. Además, la restricción (3.4) garantiza también que si el nodo i no representa un centro, ninguna unidad básica puede ser asignada a este nodo i. Las restricciones (3.6) garantizan la conectividad en los territorios. Este tipo de restricciones fueron propuestas originalmente por Drexl and Haase [10] para garantizar la conectividad de rutas en problemas de ruteo, dichas restricciones fueron adaptadas a este problema de diseño territorial por Ríos-Mercado y Fernández [28]. Es importante destacar que el número de estas restricciones crece de manera exponencial respecto al tamaño del problema, lo que hace prácticamente imposible escribirlas por completo para problemas de tamaño relativamente mediano. Por ejemplo, para un problema con n=10 y p=3 se tendrían 854 restricciones de este tipo, mientras que para un problema con n=20 y p=5 se tendrían 1,834,988 restricciones de este tipo.

3.6. Complejidad del problema

En esta sección se demuestra la complejidad del problema descrito anteriormente.

Teorema 3.1. El problema TDP es NP-duro.

Demostración. Para argumentar la demostración vemos primero que para una solución dada, el verificar si la solución es o no factible puede efectuarse en tiempo polinomial. Esto es fácil de ver para las restricciones (3.2)-(3.5). En cuanto a las restricciones (3.6), aunque éstas son un número exponencial, recordemos que su equivalente en la formulación combinatoria es que $G_k = (V_k, E(V_k))$ sea conexo para todo $k \in K$. Es bien conocido que la conexidad de un grafo puede comprobarse o descartarse en tiempo polinomial, por ejemplo con métodos de búsqueda por profundidad (depth-first search, en inglés). De aquí se concluye entonces que comprobar o descartar la conexidad de TDP puede verificarse en tiempo polinomial y por consiguiente TDP$\in$ NP.

La segunda parte de la demostración consiste en encontrar un problema NP-duro que sea reducible polinomialmente al TDP. Una forma de llevar esto a cabo es tomando una instancia particular del TDP donde $\tau^a = \infty, a \in A$ y G sea un grafo completo. Para esta instancia particular, las restricciones (3.4)-(3.6) se vuelven redundantes y el problema resultante es el conocido problema de la p-mediana. Kariv y Hakimi [23] han demostrado que el problema de la p-mediana pertenece a la clase de problemas NP-duros. Debido a que una instancia del problema TDP resulta en el problema de la p-mediana se puede concluir que el problema TDP $\in$ NP-duro. □

3.7. Resolviendo el modelo en forma exacta

Si bien es cierto, como ya se mencionó anteriormente, que las restricciones (3.6) no pueden escribirse explícitamente, resolver el modelo TDP relajando dichas restricciones nos sirve para brindar información importante, particularmente en dos sentidos. El primero, es para obtener una idea del alcance del Método de Ramificación y Acotamiento (MRA) en función del tamaño del problema. El segundo, y muy importante, es que la solución del modelo relajado proporciona una cota inferior al valor óptimo de la solución del TDP. Esta cota brinda una medida para comparar la calidad de las soluciones encontradas por los métodos heurísticos.

Para resolver el TDP relajado (TDPr) se hizo uso de GAMS [4], el cual es un paquete de modelación algebraica para problemas de optimización. En este caso, se utiliza el MRA del módulo GAMS/CPLEX, con algunas recomendaciones algorítmicas reportadas en Solís García [29], como por ejemplo, la estrategia de dar prioridad a las variables x_{ii} sobre las x_{ij} en el proceso de ramificación del MRA.

El tamaño máximo de instancia para el cual se pudieron obtener soluciones del TDPr fue de 500 unidades básicas y 20 territorios. Sin embargo, no se garantiza la conectividad territorial y el tiempo en obtener dichas soluciones fue considerablemente alto. Debido a que el tamaño de las instancias reales es mucho más grande que 500 unidades básicas, queda claro que el MRA no es una herramienta de solución adecuada para la empresa. Esto hace necesario el empleo de otras técnicas de solución que aunque no garanticen soluciones óptimas, sí sean capaces de otorgar soluciones de buena calidad a problemas grandes en un tiempo de cómputo razonable. Es decir, se hace necesario el uso de heurísticas, lo que nos lleva al algoritmo propuesto en este trabajo de tesis.

3.8. Discusión

El problema de diseño de territorios se puede modelar como un problema restringido de partición de los nodos de un grafo. Inclusive algunas restricciones del modelo formulado guardan similitud con aquellas que definen al problema general de agrupamiento. Sin embargo, las restricciones de conectividad poseen un crecimiento exponencial e imponen una mayor dificultad a la formulación y solución del problema.

No obstante, a través del modelo basado en un grafo se logra la fácil comprensión del problema, y se hace además posible el manejo de las restricciones más difíciles del problema. También se obtiene como modelo resultante un modelo de programación matemática lineal entero-mixto, determinista. Su función objetivo representa una medida de dispersión basada en p-mediana la cual se desea minimizar.

Una relajación del modelo resultante de programación entera mixta se ha intentado resolver de manera exacta y se observó que para instancias de hasta 500 nodos y 20 territorios a formar es posible encontrar a través de esta vía soluciones óptimas al problema relajado. Sin embargo, para instancias de mayor tamaño no fue posible encontrar soluciones aceptables.

Aunque el proceso analizado en este capítulo no nos conduce a la solución de una instancia real del problema, contar con un modelo matemático de este tipo puede al menos permitir soluciones óptimas para instancias pequeñas y ayudar así a idear técnicas para solucionar el problema y obtener un conocimiento de la estructura del mismo.

El tamaño del problema original, sumado a la complejidad de las restricciones del problema hace necesario el empleo de técnicas aproximadas para obtener soluciones de buena calidad. Por ello se ha desarrollado una metodología de solución basada en una heurística que ha sido probada con éxito en problemas similares. Sin embargo, se hace necesaria una adaptación debido a restricciones que no han sido tratadas en la literatura. En el siguiente capítulo se explica con detalle esta metodología de solución.

Capítulo 4

METODOLOGÍA DE SOLUCIÓN

En el capítulo anterior se explicó la necesidad de desarrollar una metodología aproximada para resolver el problema abordado en esta tesis debido a que la complejidad del problema en instancias reales impide solucionar el modelo de programación entera de manera exacta.

El problema de diseño de territorios ha sido abordado en la literatura por diversos investigadores; sin embargo, existen distintas versiones del problema debido a que las características del mismo suelen variar según las necesidades de la empresa u organización que busca encontrar un buen diseño de territorios. Entre los métodos que han reportado un mayor éxito para este tipo de problemas en la literatura destacan los algoritmos de Localización-Asignación. El éxito de esta metodología radica en explotar la estructura del problema de tal forma que se divida el problema en subproblemas de una menor complejidad para que puedan ser solucionados de una manera más sencilla y con un menor esfuerzo computacional.

Cabe aclarar que ninguna de las técnicas de Localización-Asignación que existen en la literatura puede aplicarse al problema abordado en esta tesis. Las restricciones de conectividad y las múltiples restricciones de balanceo no han sido abordadas en trabajos de este tipo hasta donde se tiene conocimiento. Por ello en este trabajo de tesis el principal aporte es una adaptación de la técnica conocida como Localización-Asignación, combinada además con técnicas de búsqueda local, que permiten manejar estas restricciones antes mencionadas.

En las siguientes secciones de este capítulo se explican cada una de las fases del algoritmo desarrollado, así como el esfuerzo computacional que se requiere para llevar a cabo cada fase.

4.1. ALGORITMO DE LOCALIZACIÓN-ASIGNACIÓN

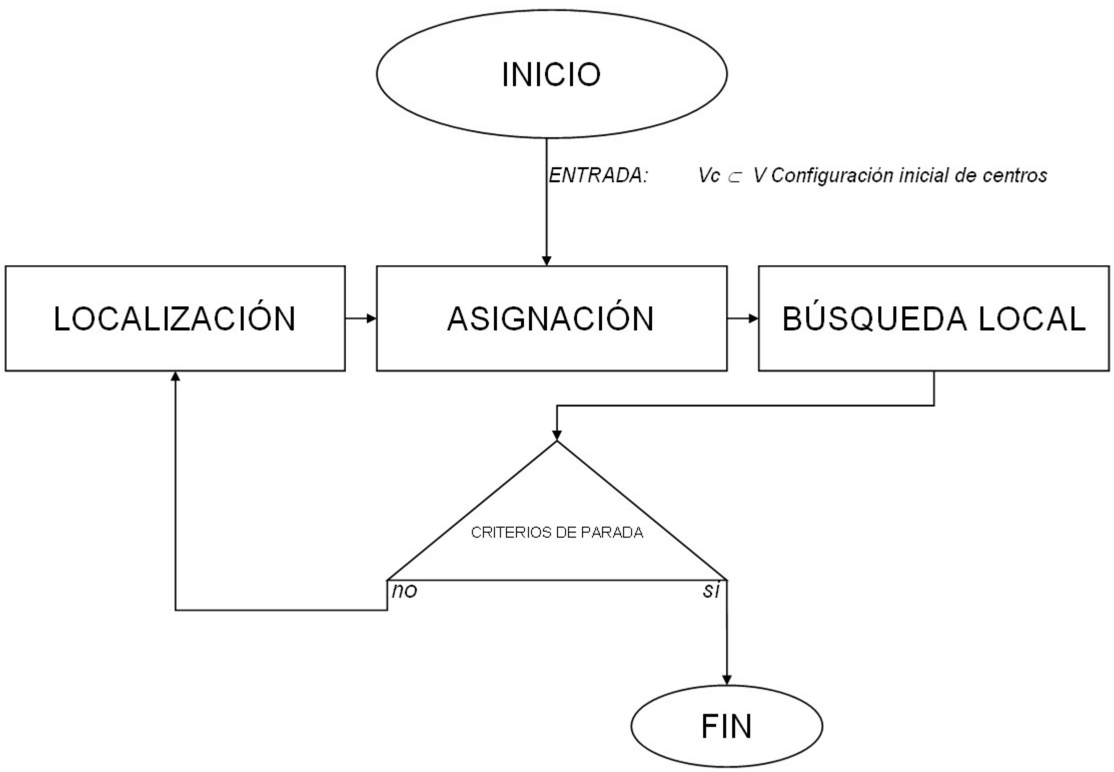

Figura 4.1: Esquema general de Localización-Asignación

La técnica de Localización-Asignación explota la estructura del problema de manera que se separan las decisiones de crear los territorios en dos fases independientes: una fase de Localización donde los centros de los territorios son elegidos, y después una fase de Asignación donde las demás áreas básicas son asignadas a estos centros. Estas fases son generalmente iterativas hasta que cierto criterio de parada es satisfecho.

Como se puede observar en la Figura 4.1, el algoritmo propuesto consta de estas dos fases de Localización-Asignación combinadas con una tercera fase de Búsqueda Local donde se intenta mejorar el valor de la solución obtenida en la última fase. Es decir que se tendrían 3 fases independientes seguidas una de otra. La Figura 4.2 muestra el pseudocódigo del método.

ENTRADA Iteraciones sin mejora: m.
SALIDA: Mejor solución encontrada S_{best}.

$S_{best} = \oslash, S_{act} = \oslash, I = 0$
MIENTRAS $I \leq m$ **HACER**
  $V_c \leftarrow$ LOCALIZACIÓN (S_{act});
  $S_{act} \leftarrow$ ASIGNACION (V_c);
  $S_{act} \leftarrow$ BUSQUEDA_LOCAL (S_act);
  SI S_{act} es mejor que S_{best} **ENTONCES**
    $S_{best} \leftarrow S_{act}$;
    $I = 0$;
  SI NO
    $I \leftarrow I$+1;
  FIN SI
FIN MIENTRAS
DEVOLVER S_{best}

Figura 4.2: Pseudocódigo del Algoritmo Localización-Asignación

En el algoritmo puede verse como estas tres fases se realizan iterativamente hasta que un criterio de parada es satisfecho. Como criterio de parada de este algoritmo se utilizó un límite de iteraciones m. El algoritmo se detiene si en una cantidad m de iteraciones consecutivas no se ha logrado mejorar la mejor solución obtenida S_{best}. Cabe mencionar que a partir de que se ejecutan las tres fases del algoritmo se obtiene una nueva solución. En las siguientes secciones se detalla cada fase del algoritmo.

4.2. Fase de Localización

La fase de Localización es la más simple del algoritmo, dicha fase consiste en obtener nuevos centros para la fase de localización a partir de la última solución encontrada. En esta sección se detalla cómo son encontrados los nuevos centros que se utilizarán en una posterior fase de asignación.

4.2.1. Configuración inicial

Al inicio del algoritmo no se cuenta con una solución, por lo tanto es necesario definir una configuración inicial de centros para poder iniciar así la fase de asignación. Es decir, es necesario definir de antemano cuáles serán los centros iniciales de nuestro problema.

Cabe mencionar que esta decisión es importante ya que la configuración de estos centros tiene un impacto en la calidad de soluciones que arroje el algoritmo. Por lo tanto, se debe tener cuidado de cómo se obtendrá esta configuración inicial. Dicha elección consistiría en elegir p unidades básicas de entre el total de n unidades básicas existentes en el problema.

A continuación se explican cómo se han obtenido estas configuraciones iniciales para el algoritmo, así como los resultados y las observaciones obtenidas con estas técnicas.

Primera solución entera encontrada

Una solución factible al problema puede proporcionar una configuración inicial. Es decir, que a partir de un diseño de territorios cualesquiera que satisfaga las restricciones impuestas se podrían obtener los centros de los cuales partiría el algoritmo.

Hay que destacar que en un principio ésta fue nuestra vía para obtener las configuraciones iniciales. A partir del modelo de programación entera mixta descrito en la Sección 3.5 se utilizó el programa de optimización CPLEX para intentar solucionar el problema. Dicho programa de optimización no es utilizado para encontrar la solución óptima ya que se ha estudiado que esto es prácticamente imposible en instancias grandes. En cambio, es utilizado solamente para encontrar la primera solución factible al problema, es decir, la primera solución que satisfaga las restricciones de balance y asignación única. Esto resulta en un diseño de territorios del cual podemos obtener una configuración de centros.

Sin embargo, al momento de la implementación se encontró un gran problema. En instancias grandes el programa de optimización tardaba demasiado en encontrar la primera solución factible, lo que hacía que se perdiera un tiempo considerable en determinar la configuración inicial. Incluso en casos extremos no se encontraba dicha configuración. Por lo tanto, se tuvo que buscar otra forma de establecer esta configuración inicial de centros.

GRASP

Debido al gran esfuerzo que ocasionaba el tratar de encontrar una solución factible a partir del modelo de programación entera mixta se pensó en utilizar técnicas heurísticas que fueran lo suficientemente rápidas para proporcionar una solución factible. GRASP es una de las técnicas que posee este tipo de características.

Para la obtención de la configuración inicial de centros se utilizó un algoritmo ya existente de GRASP para un problema de características muy similares. Dicho algoritmo fue propuesto y desarrollado por Caballero-Hernández et al. [6].

4.2.2. Localizando centros

En esta fase contamos ya con territorios formados. En el caso inicial obtenemos estos territorios a través de GRASP y en los casos posteriores se obtienen de las fases de asignación y búsqueda local que fueron ejecutadas en la iteración anterior. El objetivo ahora es recalcular nuevos centros para una siguiente fase de asignación. Esto se logra resolviendo, independientemente para cada territorio, un problema de 1-mediana de la siguiente forma: para cada territorio se selecciona como centro aquel nodo para el cual se minimice la suma de las distancias de dicho nodo a todos los demás. Es decir se selecciona $c_k = \text{argmin}_{i \in V_k}\{\sum_{j \in V_k} d_{ij}\}$, donde V_k es el conjunto de nodos en el territorio k y c_k es el centro del mismo.

Esta fase es la más sencilla del proceso ya que su complejidad computacional es de orden $O(n^2p)$; sin embargo, es necesario destacar que la elección de centros tiene un impacto considerable sobre los territorios resultantes en fases posteriores.

4.3. Fase de Asignación

Como entrada para esta fase se utiliza la configuración de los centros localizados en la fase previa. Es decir que ya se conocen los centros de los territorios, los cuales denotaremos por $V_c \subset V$. A partir de esta configuración de centros es necesario asignar cada unidad básica a un centro respetando todas las restricciones de nuestro problema.

4.3.1. Modelo de Asignación

Debido a que ya conocemos los centros de nuestro problema el número de variables y restricciones del problema original es considerablemente reducido. Así entonces podemos reformular el problema TDP de la Sección 3.5 como sigue:

Modelo TDP-Asignación (TDP-A)

$$\text{Minimizar} \quad f(x) = \sum_{i \in V_c} \sum_{j \in V} d_{ij} x_{ij} \tag{4.1}$$

$$\text{sujeto a} \quad \sum_{i \in V_c} x_{ij} = 1 \qquad j \in V \tag{4.2}$$

$$\sum_{j \in V} w_j^a x_{ij} \leq (1 + \tau^a) \mu^a x_{ii} \qquad i \in V_c, a \in A \tag{4.3}$$

$$\sum_{j \in V} w_j^a x_{ij} \geq (1 - \tau^a) \mu^a x_{ii} \qquad i \in V_c, a \in A \tag{4.4}$$

$$\sum_{j \in \cup_{v \in S} N^v \setminus S} x_{ij} - \sum_{j \in S} x_{ij} \geq 1 - |S| \qquad i \in V_c, j \in V \tag{4.5}$$

$$x_{ij} \in \{0, 1\} \qquad i \in V_c, j \in V \tag{4.6}$$

Nótese que aunque el número de variables se ha reducido de n^2 a np, y las restricciones (3.3) ya no están presentes; el TDP-A sigue siendo NP-duro. Es decir, este problema sigue siendo intratable por métodos exactos.

Relajación del Modelo de Asignación

Para determinar una formulación adecuada que permita un mejor desempeño del algoritmo iterativo se propone una nueva formulación del problema de asignación que toma ventaja de la estructura del problema además de que obtiene territorios mejor balanceados. A partir del modelo TDP-A la tolerancia es reducida a cero en una actividad para lograr un perfecto balance, con lo cual las restricciones (4.3)-(4.4) se transforman en (4.9). Además, las restricciones de conexidad (4.5) son descartadas y se relaja el tipo de variables utilizadas en el modelo. El modelo resultante, para cada $a \in A$, es básicamente un modelo de transporte:

Modelo TDP-TRANSP(a), $a \in A$

$$\begin{aligned} \text{Minimizar} \qquad f(x) &= \sum_{i \in V_c} \sum_{j \in V} d_{ij} x_{ij} && (4.7) \\ \text{sujeto a} \qquad \sum_{i \in V_c} x_{ij} &= 1 \qquad j \in V && (4.8) \\ \sum_{j \in V} w_j x_{ij} &= \mu \qquad i \in V_c && (4.9) \end{aligned}$$

$$x_{ij} \geq 0 \qquad i \in V_c, j \in V \tag{4.10}$$

RESOLUCIÓN DE RELAJACIÓN DEL MODELO DE ASIGNACIÓN

Este problema TDP-TRANSP(a) puede ser resuelto eficientemente usando el método Simplex para programación lineal. Sin embargo, es necesario enfatizar que hay dos problemas, uno por cada actividad. En la solución óptima de cada modelo los territorios están perfectamente balanceados. No obstante, es posible que la restricción de asignación única no sea satisfecha debido a la naturaleza de las variables continuas. En este caso particular se hizo uso del método Simplex implementado en CPLEX para resolver este modelo a partir de los datos del problema y una configuración dada de centros. CPLEX [21] es una biblioteca de algoritmos para problemas de programación lineal y programación entera.

Como solución se tiene una asignación de cada unidad básica a los centros. Sin embargo puede presentarse la siguiente situación: una área básica para la cual más de una variable x_{ij}, $i \in V_c$ tiene valores positivos es llamada *área división* o solamente *división*. Nótese que esto significa que esta área básica ha sido asignada a dos o más territorios. Para una solución básica (óptima) del modelo TDP-TRANSP-a es fácil probar que a lo más existen $p-1$ divisiones debido al hecho de que hay a lo más $p-1$ variables no básicas cuando el modelo es resuelto por el método Simplex [22].

Ahora es necesario redondear las variables fraccionarias de cada división a uno (una variable) o cero (las otras variables). El problema radica en como redondear estas variables y seguir manteniendo el balance lo mejor posible. Este problema es llamado *Problema de resolución de divisiones* y la metodología para resolverlo es explicada en la Sección 4.3.2.

Complejidad del modelo TDP-TRANSP(a)

Aunque se conoce que el Simplex tiene una complejidad teórica exponencial, en la práctica tiene un desempeño promedio de orden polinomial. En este caso, se ha comprobado que esta fase se resuelve muy eficientemente. En la literatura se ha demostrado que el comportamiento promedio del Simplex es polinomial [30]. De hecho, comparado con el esfuerzo computacional de las otras fases, ésta puede considerarse como muy pequeña.

4.3.2. Problema de Resolución de Divisiones

Al tratar de resolver el problema de resolución de divisiones es necesario resaltar los principales aspectos que se pueden encontrar en la solución actual.

- Pueden existir varios nodos división. Es decir que hay nodos asignados a más de un centro. Por lo que es necesario asignarlos solamente a un centro.

- El criterio de conexidad no es necesariamente satisfecho ya que la restricción de conexidad no fue tomada en cuenta. Aún así se ha encontrado que la función objetivo utilizada en este problema tiende a generar territorios conexos. Sin embargo es un punto que hay que cuidar.

- Otro punto importante es cuidar el balanceo en los territorios. Si bien en un principio se puede contar con un balanceo casi perfecto, al estar resolviendo las divisiones necesariamente hay que sacrificar el balanceo. Se debe tener cuidado de afectarlo lo menos posible.

Algoritmo de Resolución de Divisiones

Con base a las observaciones realizadas en la sección anterior se propone resolver el problema de resolución de divisiones, esto es, decidir a cuál territorio será asignado cada nodo división utilizando los siguientes criterios en el orden que se enlistan:

- Reparar conexidad: Primero, cada nodo división es insertado en un territorio que sea no conexo en caso de que la inserción de dicho nodo repare la conectividad.
- Prevenir territorios no conexos: Un nodo división que causa que el territorio en el cual se vaya a agregar se haga desconexo será descartado en ese territorio.
- Balance: Los valores de las variables en la solución son usados para seleccionar el mínimo impacto en el balance de los territorios.
- Compacidad: Se calcula una medida de impacto para seleccionar el territorio donde cause el menor impacto posible en esta medida.

Análisis de Complejidad

La complejidad de esta parte del algoritmo radica en la cantidad de nodos división que se tengan, ya que por cada nodo división hay que hacer una serie de evaluaciones en cuanto a los distintos criterios.

La principal ventaja que se tiene en cuanto a los nodos división es que la función objetivo utilizada nos garantiza que existan a lo más $p-1$ divisiones debido a la estructura de las variables básicas y no básicas generadas en el modelo. Sin embargo, esta cantidad de $p-1$ divisiones es por cada modelo, y se tienen dos modelos. Por otra parte, al combinar los resultados para los dos modelos es probable que algunos nodos puedan entrar en conflicto (ser asignados a distintos territorios en cada solución) por lo que éstos también se consideran nodos división.

Para resolver un nodo división es necesario evaluar la conexidad en los territorios donde es posible asignarlo. El algoritmo utilizado para realizar esta evaluación es el conocido algoritmo DFS (Depth-first search, por sus siglas en inglés), el cual es de orden $|V| + |E|$, donde V y E representarían los vértices y aristas del grafo que representa un territorio. A este esfuerzo se debe sumar la evaluación de la función de mérito que es de orden O(n^2).

Sin embargo, la incertidumbre en el número de nodos división hace difícil evaluar la complejidad total de esta parte del algoritmo. No obstante empíricamente podemos evaluar el algoritmo para observar su comportamiento bajo diferentes condiciones. Para esto se realizó un estudio que se abordará en el siguiente capítulo y que muestra el desempeño del algoritmo bajo diferentes tipos de instancias.

4.4. Estrategia de Búsqueda Local

Después de cada iteración de las fases de Localización-Asignación se realiza una fase de búsqueda local para mejorar la calidad de las soluciones obtenidas. Cabe mencionar que hasta el momento las técnicas de Localización-Asignación no habían reportado una combinación con una técnica de este tipo.

Primero se define un vecindario $N(S)$ que consiste de todas las soluciones alcanzables a partir de una solución S por mover una unidad i de su actual terri-

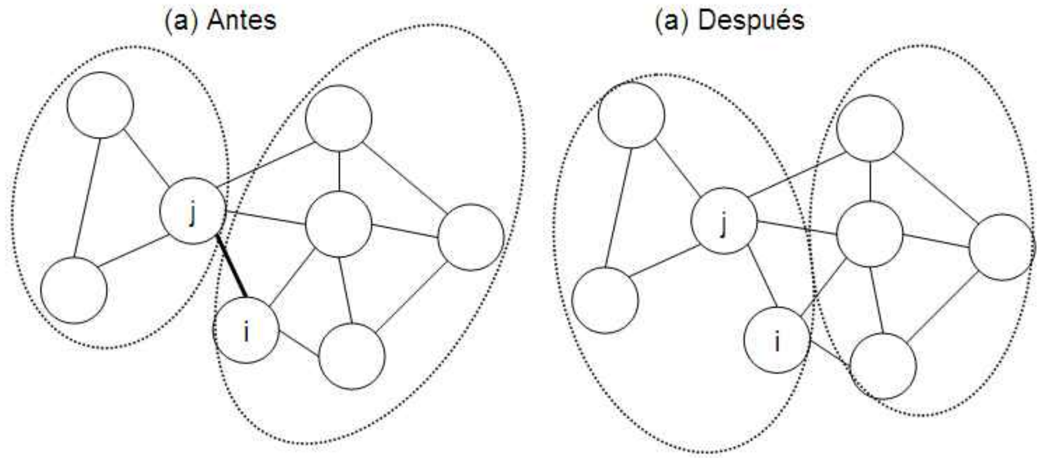

Figura 4.3: Movimiento de búsqueda local

torio $t(i)$ a un territorio vecino $t(j)$, donde j es la unidad básica correspondiente en el territorio $t(j)$ adyacente a i, esto realizándolo sin ocasionar soluciones no contiguas. Tal movimiento es denotado por $mov(i,j)$ y se ilustra en la Figura 4.3, donde $mov(i,j)$ es representado por la arista (i,j) (resaltado en negrita). Es necesario destacar que el $mov(i,j)$ es permitido solamente si los territorios involucrados en el movimiento permanecen conexos. En la práctica es necesario un criterio adicional lim_mov, y es utilizado para evitar que la búsqueda se realice por una cantidad excesiva de tiempo. Entonces el procedimiento se detiene tan pronto como se alcance un óptimo local o cuando el número de movimientos exceda el valor de lim_mov.

Para evaluar el beneficio de un movimiento se utiliza una función de mérito. Dicha función se define como: $\psi(mov(i,j)) = \lambda F(mov(i,j)) + (1-\lambda)G(mov(i,j))$, donde $F(mov(i,j))$ representa la función de dispersión y $G(mov(i,j))$ representa la función de balance. El parámetro λ es utilizado para definir la importancia que se le quiera dar ya sea a la función de dispersión o a la función de balance.

Para esta búsqueda local se pueden efectuar dos estrategias para explorar el vecindario:

- *Mover al mejor vecino encontrado (ME)*: Esta estrategia de búsqueda consiste en realizar la exploración completa de la búsqueda local, es decir, explorar todos los vecinos y hacer el mejor movimiento. La desventaja principal de esta estrategia radica en que el tamaño del vecindario crece de manera muy rápida con el tamaño del problema, lo cual ocasiona un esfuerzo computacional considerablemente alto.

- *Mover a la primera mejora encontrada(PE)*: Esta estrategia de búsqueda consiste en moverse al primer vecino que mejore la solución actual. Este tipo de estrategia permite realizar un mayor número de movimientos con un esfuerzo computacional menor al generado por la estrategia mencionada anteriormente.

En la práctica se realizaron las dos estrategias de búsqueda local. Sin embargo solo se utilizó una estrategia en la experimentación general. En el siguiente capítulo se muestran los resultados de esta experimentación.

Capítulo 5

Evaluación empírica

Una de las partes más importantes en el desarrollo de una metodología de solución para un determinado problema consiste en garantizar la calidad necesaria de las soluciones obtenidas por dicha metodología. Para esto, es necesario realizar un estudio computacional y así poder evaluar si el algoritmo cumple con las expectativas que se plantearon en un principio. En el caso del presente estudio, el algoritmo propuesto es un algoritmo heurístico, el cual no garantiza soluciones exactas y por lo tanto es necesario que las soluciones sean evaluadas de forma estadística o comparativa. Para esto, generalmente se establece una serie de experimentos controlados utilizando un conjunto de datos de prueba. Algunos estudios y reportes de experimentos computacionales de este tipo pueden verse en la literatura, por ejemplo en Rardin y Uzsoy [25] y Coy et al. [8].

En el presente capítulo se realiza un estudio computacional para evaluar el comportamiento del algoritmo de Localización-Asignación que se desarrolló en esta tesis. Primeramente, se plantean los objetivos que se pretenden lograr a partir de este estudio computacional. Después se explica la naturaleza de las instancias que se utilizaron para realizar las pruebas, así como cada uno de los aspectos a ser evaluados. Se explican los distintos experimentos que se realizaron, así como los resultados obtenidos y finalmente se discuten las observaciones y conclusiones para cada una de las pruebas.

5.1. Desarrollo Experimental

5.1.1. Objetivos principales

Primeramente se trazan los objetivos que se pretenden alcanzar con este estudio computacional. Éstos se enlistan a continuación.

- Establecer el valor adecuado para los parámetros λ (parámetro de la función de mérito en la búsqueda local, especificado en la Sección 4.4) y m (criterio de parada del algoritmo, especificado en la Sección 4.1) utilizados en la metodología desarrollada que nos permitan alcanzar mejor calidad en las soluciones obtenidas.

- Determinar la eficiencia del algoritmo desarrollado en términos de tiempo de cómputo, así como el número de iteraciones para alcanzar la mejor solución.

- Analizar el comportamiento del algoritmo bajo diversas condiciones con el fin de encontrar debilidades y fortalezas en nuestra metodología de solución.

- Determinar los límites de las condiciones del algoritmo bajo las cuales aún muestra un comportamiento adecuado. Es decir, evaluar los casos extremos del problema.

5.1.2. Condiciones experimentales

Los resultados de un experimento computacional dependen de un conjunto de condiciones experimentales. Éstas deben ser documentadas con la finalidad de que el experimento pueda ser llevado a cabo por otra persona que pueda estar interesada en evaluar el algoritmo. Dichas condiciones pueden deberse a factores relacionados con la naturaleza de los datos utilizados para llevar a cabo el experimento.

Otras condiciones que influyen en los resultados se deben al ambiente computacional y comprenden principalmente las herramientas que se utilizaron para llevar a cabo la metodología de solución. Para este estudio todos los algoritmos utilizados fueron codificados en el lenguaje C++. Cada proceso fue compilado por el compilador Sun C++ 8.0. Todos los experimentos fueron realizados en un ordenador SunFire V440 con el sistema operativo Solaris 9. Cabe mencionar que se utilizaron también las bibliotecas de optimización para C++ de CPLEX [21] en su versión 9.0. CPLEX es hoy en día una de las implementaciones más avazadas a nivel mundial para solucionar eficientemente problemas de programación lineal y programación entera mixta. En este caso se hizo uso del método simplex dual para solucionar el subproblema TDP-TRANSP(a) de la fase de asignación que fue especificado en la Sección 4.3.1.

5.1.3. Generación de Instancias del Problema

Una parte muy importante de la experimentación computacional recae en la naturaleza de las instancias que se utilizan para probar la metodología desarrollada para resolver un problema. Una instancia de un problema se define como un conjunto de datos de entrada que satisface dicho problema, es decir, un caso particular del problema. Es muy importante que las instancias del problema que se van a resolver representen lo más fielmente posible al problema real para que así las soluciones obtenidas puedan tener una interpretación más realista. Para lograr este objetivo, la empresa que motivó el presente estudio colaboró en proveer información importante sobre las características de los datos reales. Esto permitió identificar las características más importantes del problema y definir así el conjunto de instancias de prueba necesarias para la evaluación del algoritmo desarrollado.

Para este experimento computacional se utilizó un programa que se encarga de generar las instancias. Este programa recibe como entrada el número de unidades básicas que se desean agrupar (n), así como el número de territorios a formar (p). También es posible ajustar la tolerancia de desbalanceo permitido en las actividades ($\tau^a, a = 1, 2$). El programa generador, a partir de estos datos de entrada, se encarga de generar instancias del problema y las imprime en archivos de texto en un formato adecuado. Estas características que definen la instancia son almacenadas en un archivo de texto que es dado como entrada para el algoritmo Localización-Asignación.

A continuación se describe como son generadas las características específicas para cada instancia.

UBICACIÓN DE UNIDADES BÁSICAS

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 x 500].

NATURALEZA DEL GRAFO

Para una representación más acorde a la realidad se necesita que el grafo formado para representar una instancia del problema sea un grafo planar, es decir, que las aristas que lo conforman no deben cruzarse entre si. Además, el grafo debe ser conexo para que puedan formarse así territorios conexos. Para el desarrollo de los grafos se decidió crear tipos de grafos llamados grafos Gabriel, ya que este tipo de grafos cumple con los requerimientos anteriormente señalados. El método para obtener este tipo de grafos fue desarrollado por Gabriel y Sokal [16].

Así entonces, se implementó la metodología para generar un grafo Gabriel a través de un programa en C++. Dicho programa construye el grafo al ir agregando a los nodos las aristas necesarias para construir el grafo.

Actividades Nodales

Para generar las actividades nodales en cada unidad básica se decidió partir de una instancia real del problema proporcionada por la compañía. Dicha instancia real permite conocer cómo es la naturaleza aproximada de las actividades nodales. Se hizo notar por la empresa que la actividad 1 correspondiente a el número de clientes en cada unidad básica comprendía de 0 a 3 clientes por manzana mientras que el volumen de ventas era de 1 a 12 unidades por manzana. Para obtener la representación de las actividades se utiliza una distribución uniforme.

La instancia real proporcionada por la empresa es de un tamaño de 34000 manzanas, sin embargo, las instancias que se necesitan generar para realizar el estudio del algoritmo son de un menor tamaño. Así entonces, se realiza una trasformación para que, dependiendo del tamaño de la instancia que se desea generar, las actividades nodales sean proporcionales al tamaño de la instancia real.

Para cada unidad básica se generan los valores para las actividades nodales en base a la suma de N variables aleatorias uniformes comprendidas entre los rangos de [0,3] y [1,12] para el número de clientes y demanda de producto respectivamente. Aquí N, la cantidad de variables aleatorias que necesitan sumarse para realizar así la transformación de la instancia real a la instancia de tamaño n que se desea generar, está dada por la función piso de $N = \lfloor 34000/n \rfloor$.

5.1.4. Características y parámetros de las instancias

Para poder garantizar una mejor evaluación del algoritmo es necesario que éste sea ejecutado bajo una gran cantidad de instancias con diferentes características. Esto con el fin de detectar posibles cambios en el comportamiento del algoritmo y lograr así un conocimiento del mismo.

La generación de instancias fue dada bajo los siguientes valores en las siguientes características que definen cada instancia:

- Número de unidades básicas (n): 500, 1000 y 2000 unidades básicas.
- Número de territorios a formar (p): 20, 40 y 60
- Porcentaje de tolerancia de desviación permitida ($\tau^a, a = 1, 2$): 5 %

Esto da una combinación de 3x3x1 = 9 combinaciones en cuanto a las características de las instancias. Para cada combinación fueron generadas además 30 réplicas diferentes. Es decir, 30 instancias con las mismas características, pero diferentes datos. Teniendo en total 270 instancias sobre las cuales se realizó la experimentación de nuestra metodología.

5.1.5. Aspectos a evaluar del algoritmo

Para poder obtener una evaluación detallada del algoritmo propuesto, se toman en cuenta los siguientes aspectos:

- *Tiempo de ejecución.* Es primordial para cualquier algoritmo de solución que las soluciones obtenidas se obtengan en un tiempo razonable. Al evaluar el tiempo de ejecución se conocerá el tiempo en el cual es posible alcanzar las soluciones y así se determinará el esfuerzo computacional que se necesita para cada tamaño de instancia.

- *Número de Iteraciones.* Este dato es útil debido a que proporciona una idea de cuánto esfuerzo se necesita para que el algoritmo converja a la mejor solución posible.

- *Mejoría en búsqueda local.* A partir de una solución encontrada por una iteración de Localización-Asignación se ejecuta un proceso de búsqueda local, por lo tanto es necesario evaluar esta mejoría con el fin de conocer la calidad de las soluciones obtenidas tanto antes como después de aplicar la búsqueda local.

- *Número de nodos división.* La fase de asignación en la cual es necesario resolver los nodos división que se originan al relajar el problema de asignación es una de las partes más pesadas de nuestro algoritmo. El esfuerzo computacional en esta fase está precisamente dado por el número de nodos división que se resuelven. Por lo tanto, es necesario evaluar esta medida para conocer el esfuerzo computacional que se tiene en esta parte.

- *Factibilidad de Balanceo.* El algoritmo no tiene garantía de poder encontrar una solución factible en cuanto a balanceo. Esto puede complicarse debido a la naturaleza de las instancias o también a tolerancias muy pequeñas en esta restricción. Así entonces, es necesario que se conozca qué porcentaje de soluciones son factibles respecto a esta restricción en cada iteración, así como al final de la ejecución.

- *Factibilidad de Conexidad.* El algoritmo no garantiza el poder encontrar una solución factible en cuanto a conexidad debido a que tal restricción no es tomada en cuenta directamente en la fase de Localización-Asignación. Sin embargo, la metodología propuesta utiliza criterios que favorecen que esta restricción sea satisfecha, particularmente cuando se intenta resolver el problema de resolución de divisiones. Por lo tanto, es necesario que se evalúe el éxito de estos criterios y se mida entonces el porcentaje de soluciones que son factibles respecto a esta restricción en cada iteración, así como al final de la ejecución.

Para cada uno de estos aspectos se utilizaron contadores en el algoritmo de solución que se encargan de proporcionar estas medidas para cada instancia procesada. Dicha información es almacenada en archivos de texto para su posterior análisis.

5.1.6. Diseño de Experimentos

La experimentación comprende básicamente tres etapas:

- Etapa 1: Experimentación para establecer el valor adecuado de los parámetros utilizados en el algoritmo Localización-Asignación.
- Etapa 2: Experimentación para estudiar el comportamiento del algoritmo bajo diversas combinaciones de los factores del problema de diseño de territorios.
- Etapa 3: Experimentación para estudiar el comportamiento del algoritmo bajo condiciones extremas en los factores del problema de diseño de territorios.

5.2. Análisis de resultados

5.2.1. Etapa 1: Calibración de Parámetros de la Heurística

Debido a que el algoritmo de Localización-Asignación propuesto posee parámetros que es necesario ajustar, se necesita realizar una experimentación con el objetivo de encontrar los valores de estos parámetros con los cuales se obtengan las mejores soluciones.

Tamaño	$\lambda = 0.7$	$\lambda = 0.8$	$\lambda = 0.9$	$\lambda = 1.0$
500	2	22	6	0
1000	2	21	7	0
2000	3	23	4	0

Tabla 5.1: Calibración del parámetro λ

Cabe mencionar que no todas las instancias fueron resueltas en esta etapa. Los tamaños de instancia que se solucionaron fueron de 500, 1000 y 2000 unidades básicas. El número de territorios a formar se dejó fijo en 40 territorios. Para cada combinación de características fueron utilizadas las 30 réplicas dando un total de 3x30 = 90 instancias a resolver.

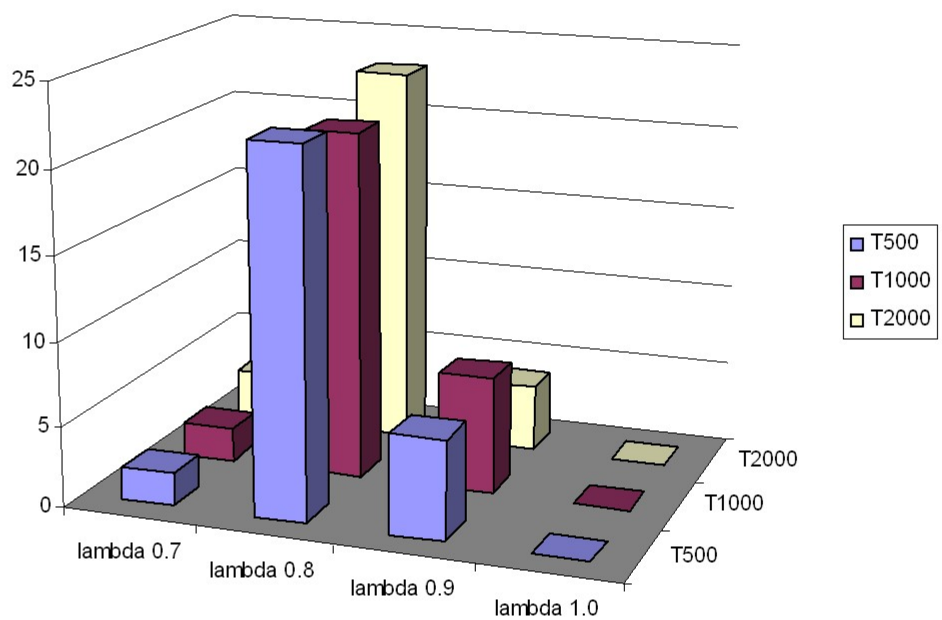

Figura 5.1: Comportamiento del parámetro λ

En la Tabla 5.1 se puede observar una comparación de los resultados obtenidos al probar distintos valores de λ para los diferentes tamaños de instancias probadas. La primera columna corresponde al tamaño de las instancias que se evaluaron. Las siguientes columnas corresponden a la cantidad de veces en las cual la mejor solución fue encontrada con el valor del parámetro λ dado por el encabezado de columna. Es necesario mencionar que las comparaciones realizadas se efectuaron conforme a la función objetivo de nuestro problema y sin tomar en cuenta el tiempo de ejecución del algoritmo, es decir, la función objetivo fue el único criterio para seleccionar la solución. Como se puede constatar en la Figura 5.1, el valor de λ=0.8 predomina sobre los demás valores. Por lo tanto, el valor de λ=0.8 es el más adecuado para ponerlo fijo en la siguiente etapa, ya que este valor fue el que obtuvo la mejor calidad en las soluciones que fueron obtenidas con los distintos valores.

Otro parámetro importante está dado por el parámetro m descrito en la Sección 4.1. Este parametro nos indica el criterio de parada del algoritmo, detenemos la ejecución del algoritmo cuando no se mejora la mejor solución encontrada hasta el momento en un número de iteraciones consecutivas m. Esto se realiza con la finalidad de reducir el esfuerzo computacional del algoritmo y se disminuya así el tiempo de ejecución.

Para calcular este parámetro se analizaron los resultados obtenidos hasta el momento y se observó el comportamiento de la función objetivo a través de cada iteración, es importante mencionar que este parámetro m se dejó fijo en 100 iteraciones durante esta fase para así observar el comportamiento del algoritmo a través de las iteraciones. Así se obtiene una gráfica como la mostrada en la Figura 5.2 donde se expone cómo se comporta la función objetivo en una instancia particular de cada tamaño.

En la Tabla 5.2 tenemos el resumen del máximo número de iteraciones en el cual se produjo una mejora para cada instancia. Como se puede observar, en ninguna de las ejecuciones del algoritmo se sobrepasaron las 40 iteraciones. De hecho, el mayor número registrado de iteraciones que se tardó en realizar una

Instancia	500	1000	2000
1	24	27	35
2	25	29	35
3	26	31	33
4	26	28	37
5	25	30	34
6	25	27	38
7	24	29	38
8	24	29	37
9	28	28	36
10	27	28	35
11	27	29	36
12	25	31	36
13	25	29	38
14	25	28	34
15	26	27	36
16	24	28	38
17	27	29	36
18	27	30	37
19	27	29	34
20	27	30	34
21	28	27	36
22	25	27	39
23	27	27	35
24	27	27	35
25	23	32	35
26	28	30	37
27	25	30	34
28	28	27	34
29	25	29	37
30	26	31	34
mínimo	23	27	33
máximo	28	32	39

Tabla 5.2: Máximo de iteraciones en que se tardó en mejorar la solución

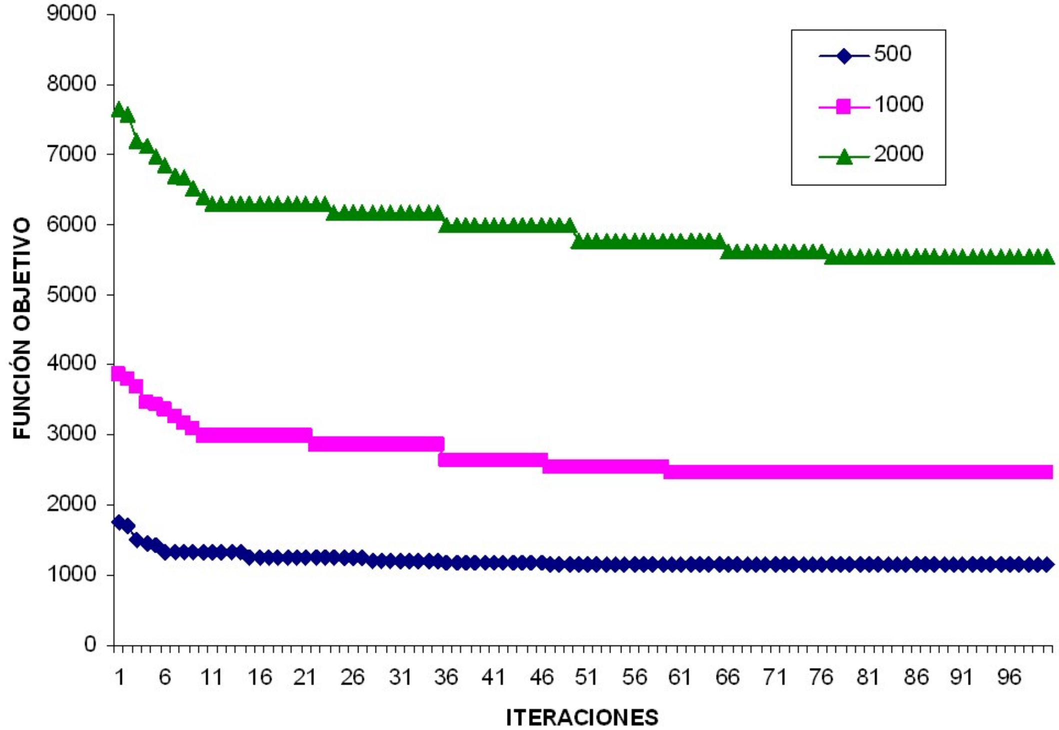

Figura 5.2: Análisis de soluciones

mejora fue de 39. Por lo tanto se decidió dejar el parámetro m fijo en 40 iteraciones para el estudio del comportamiento del algoritmo en las fases posteriores.

5.2.2. Etapa 2: Estudio general del comportamiento del algoritmo

El estudio más extenso es el que se realiza en esta etapa. El principal objetivo de este estudio es conocer el comportamiento del algoritmo de Localización-Asignación bajo diversas condiciones respecto a las características de las instancias. Así entonces se pueden detectar los puntos fuertes y débiles del algoritmo. Además, a través de este estudio podemos conocer en qué casos es posible garantizar un mejor comportamiento del algoritmo.

Debido a la gran cantidad de información que se obtiene en esta etapa fue necesario el uso de gráficas y resúmenes de información para lograr con mayor facilidad la explicación de los resultados. A continuación se exponen los resultados para cada una de los aspectos del algoritmo que se evaluaron.

TIEMPO DE EJECUCIÓN

Un aspecto importante, además de la calidad, es el tiempo de ejecución del algoritmo. Este tiempo debe ser razonable para garantizar a la empresa la obtención de soluciones en el momento que se requiera.

Como hipótesis en este experimento se tiene que los principales factores que pueden influir en el tiempo de ejecución son el tamaño de la instancia (n) y el número de territorios a formar (p).

Para verificar en cuánto influyen estos factores en el tiempo de ejecución podemos ver la Figura 5.3 en donde se observa el tiempo bajo el factor p. Cada punto de la gráfica representa un promedio de las 30 instancias para cada combinación de factores. En el eje y se tiene el tiempo de cómputo y en el eje x se tiene el número de territorios a formar (p). Cada serie Tn representa el tiempo promedio en un grupo de instancias de tamaño n y se puede observar una línea que lo representa. Como era de esperarse, cada serie de distinto tamaño n en unidades básicas tiene diferente tiempo de ejecución. Para la serie de mayor tamaño n se obtiene el mayor tiempo de ejecución. Sin embargo, observando la Tabla 5.3 que nos muestra el resumen de la información de los tiempos de ejecución podemos ver que el crecimiento en el tiempo de ejecución es razonable conforme n crece. Incluso al aumentar al doble el tamaño de instancia, el tiempo de cómputo no muestra un crecimiento más allá del 50 %.

Sin embargo, también se puede observar en la Tabla 5.3 que otro factor im-

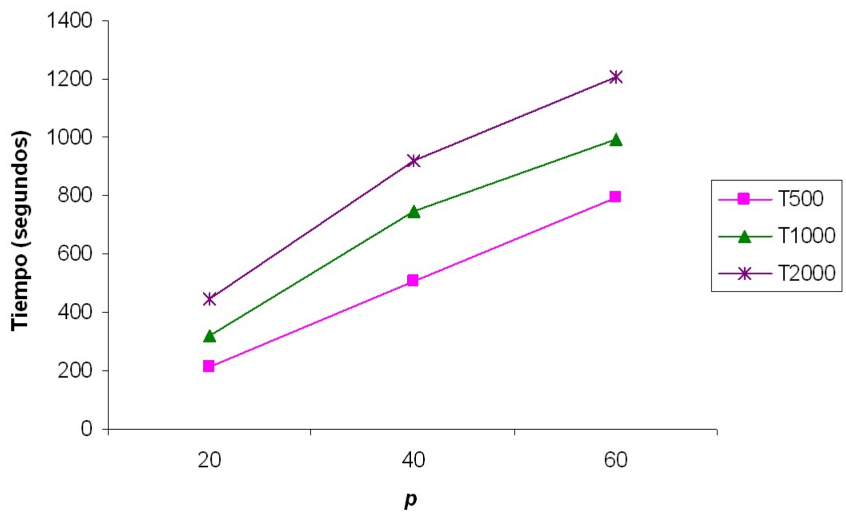

Figura 5.3: Promedio Tiempo de Ejecución

		Tiempo (segundos)		
		T500	**T1000**	**T2000**
	Mejor	201.09	298.10	412.75
p = 20	Promedio	214.53	321.31	447.85
	Peor	236.48	343.54	475.53
	Mejor	489.75	723.81	887.37
p = 40	Promedio	504.43	745.81	922.87
	Peor	523.71	767.12	953.85
	Mejor	770.98	966.03	1167.73
p = 60	Promedio	789.23	996.31	1209.50
	Peor	811.02	1025.67	1253.19

Tabla 5.3: Resumen de tiempo de ejecución

		Fase Localización (segundos - % de tiempo total)					
		T500		**T1000**		**T2000**	
	Mejor	24.93	12.40 %	40.84	13.70 %	63.15	15.30 %
p = 20	Promedio	28.96	13.50 %	46.27	14.40 %	73.45	16.40 %
	Peor	36.66	15.50 %	58.06	16.90 %	89.40	18.80 %
	Mejor	28.89	5.90 %	54.29	7.50 %	79.86	9.00 %
p = 40	Promedio	33.80	6.70 %	61.90	8.30 %	107.05	11.60 %
	Peor	37.71	7.20 %	70.58	9.20 %	127.82	13.40 %
	Mejor	32.38	4.20 %	68.59	7.10 %	147.13	12.60 %
p = 60	Promedio	37.88	4.80 %	83.69	8.40 %	183.84	15.20 %
	Peor	41.36	5.10 %	91.28	8.90 %	224.32	17.90 %

Tabla 5.4: Resumen de tiempos: Fase de Localización

portante que influye en gran cantidad sobre el tiempo de ejecución es el número de territorios a formar (p). De hecho el crecimiento respecto a este factor es grande conforme se aumenta el número de territorios a formar. Esto se puede deber a la fase de asignación que depende del número de nodos división que es necesario resolver. Este número de nodos está muy ligado a este número p. Por lo tanto también se puede concluir que el aumento en el número de territorios a formar p provoca un crecimiento en el tiempo de ejecución. Hay que recalcar que el número de territorios que la empresa busca en el caso real es de 40 a 50 territorios, por lo que se puede decir que los tiempos en los que se ejecuta el algoritmo son razonables.

Para lograr un mejor entendimiento de los factores que influyen en el tiempo de cómputo se expone el esfuerzo computacional en cada fase del algoritmo. Este estudio puede observarse en las tablas 5.4, 5.5 y 5.6. Se muestra el tiempo promedio, en segundos, que el algoritmo tardó en cada fase, así como el porcentaje que representa este tiempo sobre el total de la ejecución. Además se han registrado el mejor y peor caso en cada grupo de instancias con las mismas características.

Una primera observación es que la fase de Localización es la que tiene el

		Fase Asignación (segundos - % de tiempo total)					
		T500		**T1000**		**T2000**	
	Mejor	44.04	21.90 %	45.01	15.10 %	48.70	11.80 %
p = 20	Promedio	50.20	23.40 %	54.94	17.10 %	60.01	13.40 %
	Peor	59.59	25.20 %	67.33	19.60 %	71.80	15.10 %
	Mejor	235.57	48.10 %	268.53	37.10 %	283.07	31.90 %
p = 40	Promedio	255.24	50.60 %	293.10	39.30 %	307.32	33.30 %
	Peor	269.71	51.50 %	312.99	40.80 %	339.57	35.60 %
	Mejor	441.00	57.20 %	511.03	52.90 %	521.97	44.70 %
p = 60	Promedio	470.38	59.60 %	540.00	54.20 %	558.79	46.20 %
	Peor	502.02	61.90 %	572.32	55.80 %	602.79	48.10 %

Tabla 5.5: Resumen de tiempos: Fase de Asignación

menor tiempo de cómputo. Puede también apreciarse, de la Tabla 5.4, que para esta fase el tiempo de cómputo crece más con n que con p. Aún así, el crecimiento del tiempo de cómputo es bastante razonable.

En la Tabla 5.5 correspondiente a la fase de Asignación podemos observar un caso completamente opuesto al de la fase de Localización. En este caso, al incrementar el tamaño de las instancias, el tiempo de cómputo no se ve grandemente afectado. Sin embargo, bajo diferentes valores del parámetro p, se observa un crecimiento notable en el tiempo de ejecución. Este comportamiento puede ser atribuido a la cantidad de nodos división que necesitan ser resueltos en esta fase de Asignación, este número de nodos depende directamente del parámetro p.

Por último, en la Tabla 5.6 puede observarse el esfuerzo computacional en la fase de Búsqueda Local. En este caso, la influencia tanto del número de unidades básicas n, como el número de territorios a formar p es notable. Cuando se incrementa cualquiera de estos dos valores se ocasiona un aumento en el esfuerzo computacional que realiza el algoritmo.

		Fase BL (segundos - % de tiempo total)					
		T500		**T1000**		**T2000**	
	Mejor	124.07	61.70 %	202.41	67.90 %	283.97	68.80 %
p = 20	Promedio	135.37	63.10 %	220.10	68.50 %	314.39	70.20 %
	Peor	152.53	64.50 %	244.26	71.10 %	342.38	72.00 %
	Mejor	197.37	40.30 %	363.35	50.20 %	455.22	51.30 %
p = 40	Promedio	215.39	42.70 %	390.80	52.40 %	508.50	55.10 %
	Peor	236.72	45.20 %	415.78	54.20 %	552.28	57.90 %
	Mejor	245.94	31.90 %	330.38	34.20 %	421.55	36.10 %
p = 60	Promedio	280.97	35.60 %	372.62	37.40 %	466.87	38.60 %
	Peor	303.32	37.40 %	411.29	40.10 %	507.54	40.50 %

Tabla 5.6: Resumen de tiempos: Fase de Búsqueda Local

NÚMERO DE ITERACIONES

El siguiente aspecto a evaluar corresponde a la cantidad de iteraciones necesarias para llegar a la mejor solución encontrada. Es de esperarse que este número de iteraciones esté muy relacionado al tiempo de ejecución.

Como hipótesis en este experimento también se tiene que los principales factores que pueden influir en el tiempo de ejecución son el tamaño de la instancia (n) y el número de territorios a formar (p).

Para verificar en cuánto influyen estos factores en el tiempo de ejecución podemos ver la Figura 5.4 en donde se observa el número de iteraciones promedio bajo las diferentes combinaciones de factores. Cada punto de la gráfica representa un promedio de iteraciones de las 30 instancias para cada combinación de factores. En el eje y se tiene el número de iteraciones promedio y en el eje x se tiene el número de territorios a formar. Cada serie Tn representa representa un tamaño de instancia n y es observada por una línea que la representa.

Como era de esperarse se puede observar en la Tabla 5.7 que el resumen nos arroja un comportamiento similar al mostrado en la gráfica de tiempo de

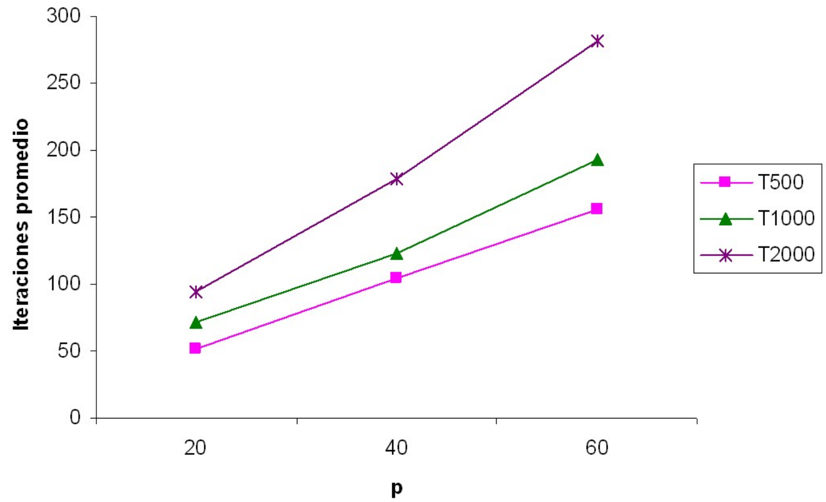

Figura 5.4: Promedio Iteraciones

		Número de iteraciones		
		T500	**T1000**	**T2000**
	Mínimo	43.00	58.00	79.00
p = 20	Promedio	50.00	72.07	95.27
	Máximo	58.00	92.00	117.00
	Mínimo	87.00	105.00	148.00
p = 40	Promedio	105.33	122.50	178.40
	Máximo	114.00	133.00	214.00
	Mínimo	144.00	175.00	252.00
p = 60	Promedio	156.47	193.10	280.80
	Máximo	170.00	211.00	309.00

Tabla 5.7: Resumen de iteraciones

cómputo. Por lo tanto, podemos obtener conclusiones similares a partir de esta gráfica. Los valores de n y p influyen de manera similar en el número de iteraciones como en el tiempo de ejecución.

Selección de búsqueda local

Como se vió en la Sección 4.4 se implementaron dos estrategias de búsqueda local, la primera estrategia es la de movernos al primer movimiento de mejora (Primero Encontrado, PE), mientras la segunda estrategia consiste en explorar todos los vecinos y movernos al mejor (Mejor Encontrado, ME). En esta sección se muestran los resultados previos en los cuales se seleccionó la estrategia de búsqueda local que reportó mejores resultados.

Para esta parte de la experimentación se utilizaron instancias con los parámetros fijos en número de territorios a formar (p=40) y tolerancia en desbalanceo (τ^a=0.05, a=1,2). Se variaron los tamaños de las instancias para n=500, 1000 y 2000, y además se generaron 30 réplicas para cada tamaño de instancia.

Para cada instancia se utilizaron las dos estrategias de búsqueda local, utilizando un límite de 100 movimientos para cada estrategia. Los principales factores que se evaluaron en este experimento son:

- *Mejora relativa.* Esta medida se refiere a la mejora promedio que se logró para cada instancia. Es calculada a partir de la solución inicial de la que parte la búsqueda local. La función objetivo de dicha solución inicial es comparada con el óptimo local encontrado por la estrategia de búsqueda.
- *Tiempo de ejecución.* Esta medida se refiere al tiempo total de cómputo que se utilizó en la fase de la búsqueda local para cada instancia. Es decir, se calcula la suma de los tiempos de búsqueda local de todas las iteraciones que realizó el algoritmo.

- *Infactibilidad en balanceo.* Esta medida es calculada con la suma de las infactibilidades que se generan al violar las restricciones de balanceo.

Los resultados de este experimento se pueden ver en la Tabla 5.8. La información obtenida se encuentra agrupada por tamaño de instancia. Se calcularon los promedios para cada medida y además se muestran los mejores y peores valores encontrados de dichas medidas.

Como se puede ver en los resultados obtenidos, la estrategia de búsqueda local que parece mostrar un mejor comportamiento fue la que consiste en moverse al primer vecino en el cual se encuentra una mejora. Respecto a las mediciones de mejora relativa y tiempo de ejecución en cada caso se muestra superior a la estrategia de mejor vecino. En la medida en la que no muestran diferencia es en la factibilidad, ya que ambas estrategias lograron encontrar soluciones factibles respecto al balanceo en cada instancia.

A partir de este experimento podemos concluir que la estrategia que consiste en moverse a la primera mejora es la que genera un mejor comportamiento del algoritmo. Así que se utilizó esta estrategia de búsqueda local en los posteriores experimentos para evaluar el algoritmo.

MEJORÍA EN BÚSQUEDA LOCAL

La implementación de la búsqueda local combinada en el algoritmo de Localización-Asignación como una fase independiente no se ha reportado en la literatura anteriormente, aún en los casos de restricciones de balance únicas. Esto hace necesario el evaluar si realmente este proceso influye en la calidad de las soluciones obtenidas por el algoritmo.

Para evaluar la mejoría que se obtiene con la búsqueda local se llevó a cabo

			PE	ME
n=500	Mejora relativa (%)	Mejor	50.23	47.72
		Promedio	48.72	47.26
		Peor	46.09	41.48
	Tiempo de ejecución (seg)	Mejor	269.36	284.05
		Promedio	277.43	287.52
		Peor	288.04	293.28
	Infactibilidad relativa	Mejor	0.00	0.00
		Promedio	0.00	0.00
		Peor	0.00	0.00
n=1000	Mejora relativa (%)	Mejor	58.81	56.46
		Promedio	55.77	54.09
		Peor	51.85	50.82
	Tiempo de ejecución (seg)	Mejor	361.91	383.62
		Promedio	372.90	387.82
		Peor	383.56	391.23
	Infactibilidad relativa	Mejor	0.00	0.00
		Promedio	0.00	0.00
		Peor	0.00	0.00
n=2000	Mejora relativa (%)	Mejor	71.33	66.34
		Promedio	64.10	60.25
		Peor	58.73	55.79
	Tiempo de ejecución (seg)	Mejor	399.31	417.06
		Promedio	415.29	424.52
		Peor	429.23	434.00
	Infactibilidad relativa	Mejor	0.00	0.00
		Promedio	0.00	0.00
		Peor	0.00	0.00

Tabla 5.8: Comparación de estrategias de búsqueda local

		Mejora relativa con la BL (%)		
		T500	**T1000**	**T2000**
p = 20	Mínimo	45.60	47.99	50.63
	Promedio	47.87	51.58	57.21
	Máximo	49.70	55.92	62.76
p = 40	Mínimo	46.09	51.85	58.73
	Promedio	48.72	55.77	64.10
	Máximo	50.23	58.81	71.33
p = 60	Mínimo	48.70	54.85	63.44
	Promedio	50.18	60.19	70.38
	Máximo	51.77	63.98	76.07

Tabla 5.9: Mejora con la búsqueda local

la medición de la mejora relativa en cada solución obtenida aplicando este procedimiento. Al final se promedia dicha mejora para reportarla.

El resumen de dichos resultados se puede ver en la Figura 5.5. Cada punto de la gráfica representa el promedio del porcentaje de mejora en todas las iteraciones de las 30 instancias para cada combinación de factores. En el eje y se tiene el porcentaje de mejora promedio y en el eje x se tiene el número de territorios a formar. Cada línea representa un distinto tamaño de instancia.

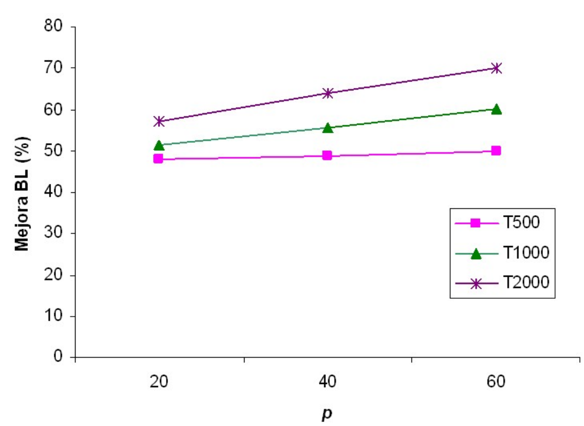

Figura 5.5: Mejora en Búsqueda Local

Se puede notar que hay una menor influencia por parte del número de territorios a formar en la mejora de la búsqueda local. El único factor que parece influir en una mayor medida es el tamaño de la instancia. Como puede verse en la Tabla 5.9, la mejora provocada por la búsqueda local está entre un 40 % y 70 %, por lo tanto se puede concluir a partir de esto que la búsqueda local es una parte primordial del algoritmo para obtener soluciones de mejor calidad. Además es importante destacar que en todos los casos se obtuvo la factibilidad respecto al balanceo.

Nodos división

Es importante conocer el número de nodos división ya que este número está relacionado con el esfuerzo computacional en la fase de asignación del algoritmo. Conociendo este número para cada instancia da una idea del esfuerzo computacional requerido al resolver dicha instancia. Sin embargo, este número no se puede calcular exactamente para cada instancia ya que depende de la naturaleza de la solución de los modelos relajados de asignación. El número de nodos división máximo que se puede obtener a partir de la solución de un problema de asignación relajado está dado por $p-1$ para cada modelo. Sin embargo, al combinar las dos soluciones de diferentes modelos se tiene una incertidumbre en este número de nodos división, ya que pueden existir nodos que en la solución de un modelo estén asignados a un territorio y en la solución del otro modelo se asignen a otro territorio. Estos nodos también son considerados nodos división

Para calcular el número de nodos división se utilizaron contadores que registran el número de nodos división en cada fase de asignación. Al final se calcula un promedio del número de nodos división obtenidos a través de las distintas iteraciones. En la Tabla 5.10 se muestra un resumen con los resultados respecto a los nodos división.

		Nodos División		
		T500	**T1000**	**T2000**
p = 20	Mínimo	24.00	35.50	56.88
	Promedio	34.81	49.33	77.64
	Máximo	44.11	66.02	98.02
p = 40	Mínimo	79.37	96.45	117.84
	Promedio	86.95	105.86	140.73
	Máximo	96.19	117.09	157.28
p = 60	Mínimo	123.87	141.25	159.25
	Promedio	131.51	156.26	180.11
	Máximo	143.39	170.19	195.24

Tabla 5.10: Promedio de nodos división

Para identificar en cuánto influyen las distintas características en el número de nodos división podemos ver la Figura 5.6 en donde se observa el promedio de nodos división bajo diferentes factores. Cada punto de la gráfica representa un promedio de las 30 instancias para cada combinación de factores. En el eje y se tiene el promedio de número de nodos división y en el eje x se tiene el número de territorios a formar p. Cada línea representa un tamaño de instancia.

En este caso se puede observar que los resultados muestran un comportamiento similar a los resultados de límite de tiempo o número de iteraciones. Como era de esperar, el principal factor que influye en estos resultados es el número de territorios formados p, mientras se incremente este número se incrementará por lo tanto el número de nodos división.

También se puede observar que el tamaño de la instancia n influye en el número de nodos división que genera el algoritmo. Sin embargo, el incremento no es tan grande como lo es con respecto al parámetro p.

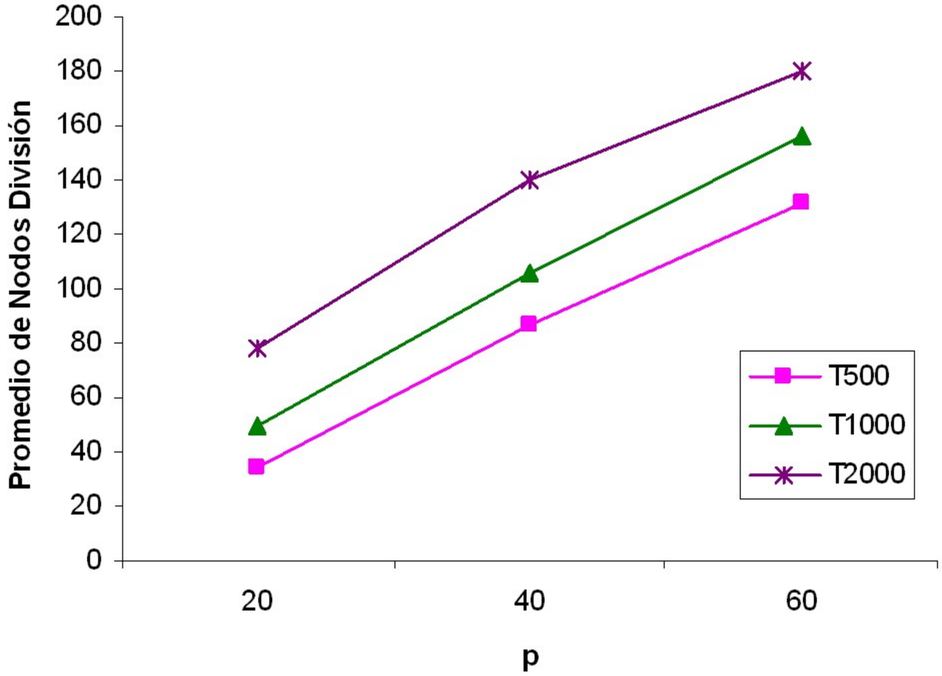

Figura 5.6: Promedio de Nodos División

p	T500	T1000	T2000
20	100 %	100 %	100 %
40	100 %	100 %	100 %
60	100 %	100 %	100 %

Tabla 5.11: Promedio de balanceo satisfecho en iteraciones

Factibilidad de Balanceo

Como se ha explicado anteriormente, el algoritmo puede generar soluciones no factibles respecto al balanceo. Sin embargo, como puede verse en la Tabla 5.11 en las instancias que se generaron para este experimento en particular siempre se encontraron soluciones factibles respecto al balanceo con una tolerancia de τ^a=0.05. Este resultado se puede atribuir principalmente a la forma de construir las soluciones en la fase de asignación, ya que se parte de un balance perfecto y no se llega a violar este porcentaje de tolerancia al resolver los nodos división.

p	T500	T1000	T2000
20	94.81 %	91.66 %	89.33 %
40	94.93 %	91.67 %	89.28 %
60	95.38 %	91.67 %	89.27 %

Tabla 5.12: Promedio de balanceo satisfecho en cada iteración

FACTIBILIDAD DE CONEXIDAD

Es importante mencionar que la metodología de solución desarrollada para resolver este problema no garantiza, en forma teórica, que se satisfaga el criterio de conexidad en cada territorio. Es decir, existe la posibilidad de que se puedan obtener soluciones no conexas como resultado.

La Tabla 5.12 muestra el porcentaje de veces que la restricción de conexidad se satisfizo en cada iteración del algoritmo para las diferentes combinaciones de n y p. Como se aprecia, si bien es cierto que no siempre se logra satisfacer la conexidad, el porcentaje es bastante alto. El peor caso lo exhiben las instancias con n=2000 con un porcentaje de éxito de casi 90 %. Si q representa la probabilidad de éxito y l el número total de iteraciones que emplea el algoritmo en toda su ejecución entonces la probabilidad de que el algoritmo falle en encontrar una solución que satisfaga la restricción de conexidad está dada por (1 - q)^l. Tomando en cuenta que empíricamente $q \geq 0.89$ y $l \geq 30$, se concluye que la probabilidad de éxito es prácticamente 1. Esto se corroboró empíricamente ya que de todas las instancias probadas con todo tipo de condiciones diferentes, fue posible encontrar 100 % de éxito, es decir, siempre se encontraron soluciones conexas, lo cual es un mágnifico resultado. Dicho éxito se debe a la forma de resolver los nodos división, ya que es tomada en cuenta la conexidad como criterio. Además, en la literatura se ha mencionado que la medida que se utilizó en la función objetivo tiende a generar territorios conexos.

Calidad de Soluciones

Para lograr un conocimiento de la calidad de las soluciones que obtiene una metaheurística es necesario comparar las soluciones obtenidas por el algoritmo contra la solución óptima del problema. Sin embargo, en la mayoría de los casos, la solución óptima de un problema es prácticamente imposible de obtener para instancias grandes que se suelen resolver con un algoritmo heurístico. En nuestro caso particular, se ha probado en la Secciones 3.6 y 3.7 que la complejidad del problema no nos permite obtener soluciones óptimas para instancias grandes como las que se han tratado con el algoritmo.

Sin embargo, otra forma de comparación es obtener cotas para el problema. Una cota, a pesar de que no garantiza ser la solución óptima al problema, puede darnos una idea de la calidad de soluciones que obtenemos con el algoritmo heurístico. En este caso particular, por medio de programación lineal podemos obtener cotas inferiores de nuestro problema resolviendo el modelo que se ha descrito en la sección 3.5. Las soluciones obtenidas por el algoritmo así podrían ser comparadas con estas cotas y podríamos decir algo de la calidad de las soluciones.

Para obtener las cotas se hizo uso del modelador algebraico GAMS [4]. Sin embargo, el problema no fue resuelto de forma entera, solo se resolvió la relajación lineal para obtener las cotas para las diferentes instancias. A partir de estas cotas se calcula un intervalo de optimalidad relativo (IOR), es decir, un porcentaje que nos indicaría que tan lejanos podríamos estar de alcanzar el óptimo. La forma de calcular este intervalo está dado por: IOR = (cota_superior - cota_inferior)/cota_inferior.

En la Tabla 5.13 podemos observar un resumen acerca de la calidad que se pudo evaluar con las cotas obtenidas por la relajación lineal del modelo. Se obtuvo un promedio para cada grupo de 30 instancias con las mismas características.

		IOR		
		T500	**T1000**	**T2000**
	Mínimo	32.38 %	34.35 %	38.95 %
p = 20	Promedio	35.30 %	38.81 %	45.32 %
	Máximo	38.06 %	43.08 %	55.52 %
	Mínimo	34.13 %	32.61 %	38.86 %
p = 40	Promedio	36.40 %	40.97 %	45.72 %
	Máximo	39.42 %	46.82 %	52.31 %
	Mínimo	34.65 %	34.93 %	40.01 %
p = 60	Promedio	37.96 %	41.84 %	46.54 %
	Máximo	41.42 %	45.97 %	53.49 %

Tabla 5.13: Evaluación de calidad de soluciones

Además se registró el mínimo y máximo IOR conforme a las cotas obtenidas.

Como puede apreciarse, el IOR varía de 32 % a 55 %, lo cual es considerablemente alto, ya que estaríamos entre 32 % y 55 % del óptimo en caso de que las cotas obtenidas fueran de excelente calidad. Sin embargo, no podemos garantizar la calidad de las cotas generadas por la relajación lineal. Por lo tanto, poco podemos concluir de la calidad de las soluciones a través de las cotas de la relajación lineal. Para un estudio más completo de la calidad de las soluciones es necesario realizar comparaciones contra mejores cotas.

5.2.3. Etapa 3: Estudio del comportamiento del algoritmo bajo casos extremos

En esta etapa el objetivo principal es probar el algoritmo bajo condiciones extremas para obtener el conocimiento de los límites a los cuales el algoritmo se comporta aún de manera adecuada.

Para este estudio se generaron instancias particulares elevando gradualmente el grado de dificultad dependiendo de la característica en cuestión que se desea

		Tiempo (segundos)					
		T500	**T1000**	**T2000**	**T3000**	**T4000**	**T5000**
	Mejor	201.09	298.10	412.75	946.65	1209.43	1733.45
p = 20	Promedio	214.53	321.31	447.85	958.78	1224.20	1742.73
	Peor	236.48	343.54	475.53	981.11	1237.04	1758.33
	Mejor	489.75	723.81	887.37	1785.80	2159.26	2518.28
p = 40	Promedio	504.43	745.81	922.87	1804.15	2173.51	2532.31
	Peor	523.71	767.12	953.85	1821.04	2190.42	2555.16
	Mejor	770.98	966.03	1167.73	2467.15	3003.03	3545.81
p = 60	Promedio	789.23	996.31	1209.50	2493.40	3034.38	3569.10
	Peor	811.02	1025.67	1253.19	2525.09	3077.82	3588.07

Tabla 5.14: Caso Extremo: Tamaño de Instancia

evaluar. El algoritmo fue evaluado para las características de tamaño de instancia (n), número de territorios a formar (p) y la tolerancia respecto al balanceo (τ).

Tamaño de Instancia

La importancia del tamaño de las instancias a resolver es primordial, ya que las instancias reales son de gran tamaño. Por lo tanto es necesario conocer el mayor tamaño al que podemos resolver una instancia del problema en un tiempo razonable.

En la Tabla 5.14 se hace un resumen de los resultados encontrados, así como también un gráfico que nos muestra el crecimiento en el tiempo respecto al tamaño de instancia. Es importante destacar que aún en las instancias de mayor tamaño se logró cumplir con la factibilidad tanto de conexidad como de balanceo.

En la Figura 5.7 se puede notar el crecimiento en el tiempo con respecto al tamaño de instancia. Es de destacar que para instancias de tamaño 5000 el tiempo de cómputo va desde los 30 hasta los 60 minutos. Sin embargo, este tiempo aún puede ser considerado como adecuado ya que para realizar un diseño de territorios, la empresa tiene un tiempo considerable debido a que el rediseño de territorios lo efectúan cada varios meses.

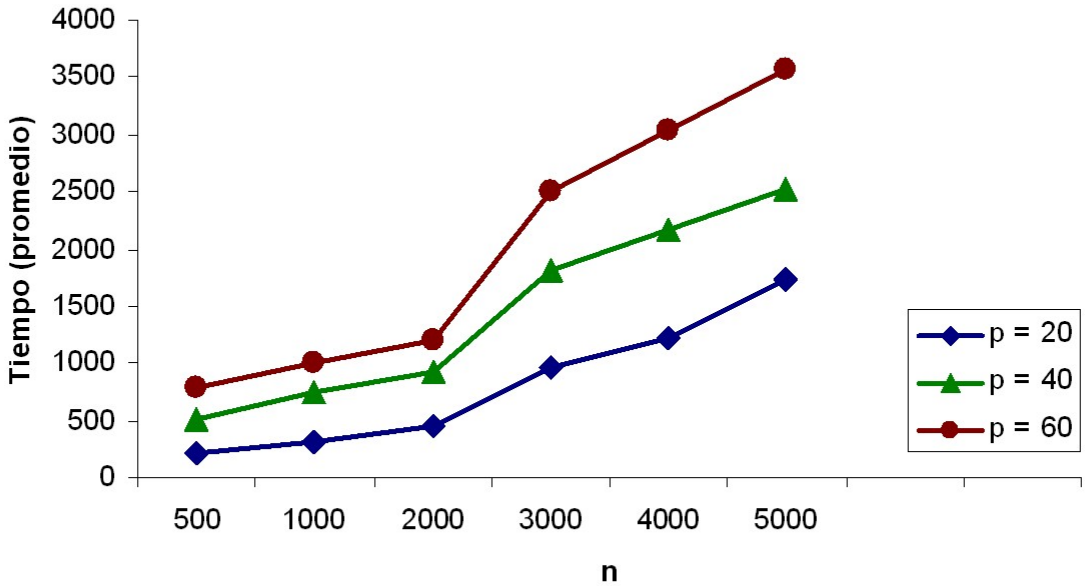

Figura 5.7: Caso extremo: Tamaño de instancia

Además, es importante mencionar que los promedios en los casos extremos se realizaron sobre un menor número de réplicas debido al elevado tiempo que era necesario para resolver cada réplica. En lugar de obtener promedios sobre 30 réplicas. Se promediaron sólo sobre 10 réplicas en los tamaños mayores a 3000 áreas básicas.

Número de Territorios a formar

Como se ha podido constatar en los experimentos anteriores, el número de territorios a formar (p) tiene una gran influencia sobre el desempeño del algoritmo. Por lo tanto es necesario también conocer el límite hasta el cual es posible resolver instancias en un tiempo razonable.

Para esto se construyeron instancias con un mayor número de territorios a formar. Los resultados se muestran en la figura. En el eje y se encuentra el tiempo promedio y en el eje x se encuentran los distintos valores de número de territorios a formar. Cada serie representa un tamaño de instancia con la tolerancia de 5 % en desbalanceo.

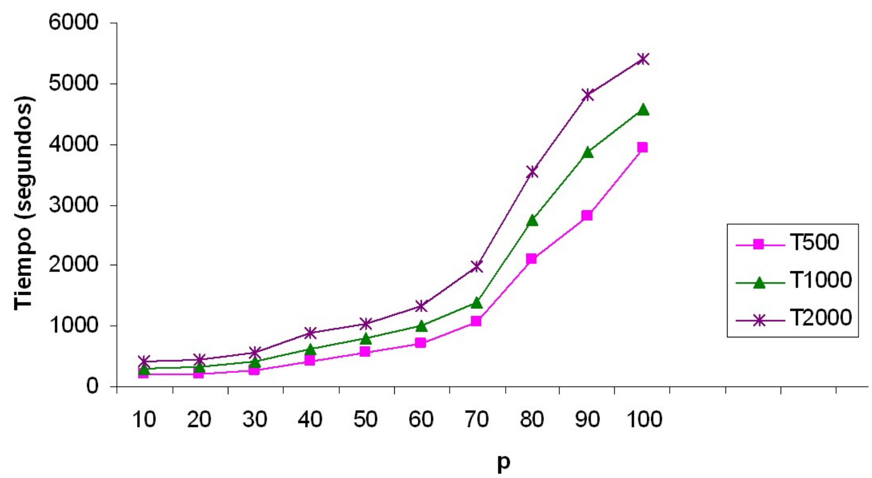

Figura 5.8: Caso extremo: territorios a formar

Como puede observarse en la Figura 5.8, el crecimiento del tiempo al variar p en el rango de 10 a 70 se mantiene relativamente estable. El crecimiento toma una pendiente mayor a partir de 70. A pesar de esto, en ningún caso empleó más de 2 horas de tiempo, lo cual es también una situación bastante manejable en la práctica. Sin embargo, el número de territorios a formar que busca la empresa va de 40 a 50 territorios, por lo que la metodología utilizada muestra un excelente desempeño, aún en instancias grandes. Además, hay que recalcar que se siguen obteniendo soluciones factibles en conexidad y balanceo en todos los casos hasta el momento.

Es necesario también destacar que los promedios en los casos extremos se realizaron sobre un menor número de réplicas debido al elevado tiempo que era

	p	T500	T1000	T2000
$\tau = 0.5$	20	100.0 %	100.0 %	100.0 %
	40	100.0 %	100.0 %	100.0 %
	60	100.0 %	100.0 %	100.0 %
$\tau = 0.4$	20	100.0 %	100.0 %	100.0 %
	40	100.0 %	100.0 %	100.0 %
	60	100.0 %	100.0 %	100.0 %
$\tau = 0.3$	20	100.0 %	100.0 %	100.0 %
	40	100.0 %	100.0 %	100.0 %
	60	100.0 %	100.0 %	100.0 %
$\tau = 0.2$	20	93.3 %	90.0 %	40.0 %
	40	100.0 %	93.3 %	53.3 %
	60	100.0 %	100.0 %	83.3 %
$\tau = 0.1$	20	0.0 %	0.0 %	0.0 %
	40	3.3 %	6.7 %	6.7 %
	60	6.7 %	10.0 %	10.0 %

Tabla 5.15: Caso Extremo: Tolerancia en balanceo

necesario para resolver cada réplica. En lugar de obtener promedios sobre 20 réplicas. Se promediaron solo sobre 3 réplicas cuando el número de territorios a formar fue mayor a 50.

Tolerancia en balanceo

El siguiente aspecto a evaluar en forma extrema es la factibilidad con respecto al desbalanceo cuando el parámetro de tolerancia τ^a se hace más pequeño. Se desea conocer hasta dónde le es posible al algoritmo encontrar soluciones factibles en cuanto a balanceo. Para llevar a cabo este experimento se probaron instancias para distintos valores de tolerancia y se tomó registro sobre si fueron encontradas soluciones factibles por el algoritmo. Se probaron tolerancias para los siguientes valores: τ^a = 1 %, 2 %, 3 %, 4 %, 5 %, con a = 1, 2.

En la Tabla 5.15 se muestran los porcentajes de instancias para las cuales se encontró solución factible en el balanceo para un determinado $\tau = \tau^a, a = 1, 2$. Por ejemplo, para instancias de tamaño 500 y con 20 territorios a formar, se pudieron encontrar soluciones factibles para el 100 % de las instancias hasta para un 3 % de tolerancia, mientras que para un 2 % de tolerancia no se encuentran soluciones factibles para todas las instancias, inclusive para una tolerancia de 1 % no se encontró solución factible para ninguna instancia.

Puede concluirse el excelente desempeño del algorimo ya que es capaz de encontrar al 100 % de soluciones factibles para todas las instancias con un $\tau^a \geq$0.03. Inclusive, su desempeño con τ^a = 0.02 es bastante bueno.

5.3. CONCLUSIONES

A partir de la evaluación computacional fue posible lograr un entendimiento completo del algoritmo desarrollado así como del problema tratado. El algoritmo de Localización-Asignación al ser adaptado para cumplir con múltiples restricciones de balance y además restricciones de conexidad logró cumplir con los objetivos al mismo tiempo que se proporcionaron soluciones de calidad en tiempos razonables.

Se encontró que el desempeño del algoritmo está completamente ligado al número de territorios a formar. Por lo tanto, esta metodología de solución muestra un mejor comportamiento y es recomendable cuando el número de territorios deseados es pequeño comparado con el número total de áreas básicas, inclusive en instancias grandes. Se pudo constatar también que las restricciones de conexidad y balanceo pueden ser fácilmente satisfechas por el algoritmo. Y además se encontró que la búsqueda local jugó un rol importante en la producción de soluciones de buena calidad ya que se notó una considerable mejora de la función objetivo después de ser aplicada.

También se realizaron estudios de casos extremos encontrando interesantes resultados que permitieron incrementar el conocimiento del problema. Se encontró que el número de territorios a formar es un factor importantísimo en el comportamiento del algoritmo de Localización-Asignación, especialmente en la fase de Localización. Además, se determinaron los parámetros adecuados para lograr un mejor desempeño del algoritmo.

Capítulo 6

Conclusiones, contribuciones y trabajo futuro

Después de realizar un estudio muy detallado sobre el problema abordado, además de investigar lo mejor forma de explotar la estructura del problema y metodologías de solución, así como codificar la metodología propuesta y realizar su experimentación correspondiente, es necesario recapacitar sobre lo que se ha logrado y como se puede continuar. En este capítulo final se expresan las conclusiones a las que se llega después del trabajo realizado. Además se presentan las contribuciones y el posible trabajo futuro que se puede desarrollar más adelante a partir de este trabajo.

6.1. Conclusiones

En el presente trabajo de tesis se ha estudiado un problema de la familia de los problemas de diseño territorial correspondiente a las necesidades reales de una empresa embotelladora ubicada en la ciudad de Monterrey, N. L., México. Es necesario resolver este problema para solucionar los requerimientos de logística que tiene la empresa. Además con el pasar del tiempo es necesario que se resuelva nuevamente debido a la incorporación o baja de nuevos puntos de venta. Los requerimientos de la empresa entran en conflicto en este problema y no es fácil encontrar una solución que satisfaga estos requerimientos en forma simultánea.

Para solucionar este problema, se realizó la formulación del mismo como un problema lineal entero-mixto determinista. Sin embargo, debido al tamaño y características de las instancias reales no se pudo resolver de forma exacta lo que llevó a la implementación de una técnica heurística para encontrar soluciones a este problema.

Los objetivos planteados en la Sección 1.3 han sido completamente satisfechos. Se ha logrado un conocimiento de cada uno de los requerimientos, restricciones y características que la empresa presentó acerca de sus necesidades reales. A partir de este conocimiento fue desarrollado el modelo matemático con el cual fue posible obtener soluciones para instancias pequeñas del problema mediante GAMS. Se desarrolló una metodología basada en la técnica de Localización-Asignación que fue adaptada para manejar las múltiples restricciones de balanceo y además restricciones de conexidad. Tal estudio no ha sido reportado en la literatura hasta donde se tiene conocimiento. Fueron generadas instancias a partir del conocimiento de la empresa por medio de un programa que fue implementado. Estas instancias fueron utilizadas para lograr una completa evaluación de la metodología de solución desarrollada.

El algoritmo desarrollado es simple, eficiente, fácil de entender y es capaz de proporcionar soluciones que cumplen con las propiedades necesarias para formar un diseño de territorios válido para la empresa (conexidad, balanceo, compacidad). Dichas soluciones han sido obtenidas en tiempos razonables a pesar de que el tamaño de las instancias utilizadas es grande. Sin embargo, el algoritmo propuesto es un algoritmo *ad hoc* para un problema de diseño de territorios con características muy específicas. Es decir que sólo podría ser utilizado como herramienta para problemas con este tipo de características. Sin embargo el objetivo de esta tesis no era desarrollar una metodología general, sino proporcionar una herramienta eficaz que permitiera trabajar el problema con todas las restricciones y consideraciones impuestas por la empresa.

6.2. Contribuciones

Las contribuciones más importantes de este trabajo se enlistan a continuación:

- Se logró un entendimiento detallado del problema abordado, planteándolo así como un problema de diseño de territorios.
- Se realizó la formulación de un modelo matemático que define completamente al problema estudiado.
- Se demostró la complejidad computacional del problema.
- Se realizaron pruebas sobre una relajación de dicho modelo matemático para determinar el tamaño máximo al cual es posible resolver el problema de manera óptima.
- Se diseño y construyó una metodología de solución basada en la técnica Localización-Asignación. Dicha técnica ha sido estudiada para este tipo de problemas, sin embargo su adaptación para tratar restricciones de conexidad y múltiples restricciones de balanceo no ha sido tratada hasta el momento.
- Fue realizado un completo estudio que probó que la metodología de solución propuesta logró obtener soluciones que cumplen con los requerimientos establecidos.

- Se logró crear una herramienta para solucionar el problema real de la empresa, como resultado de aplicar en la industria los conocimientos adquiridos a lo largo de los estudios de maestría.

- Se logró la publicación de un artículo derivado de esta tesis:

  - A location-allocation heuristic for a territory design problem in a beverage distribution firm. J. A. Segura-Ramiro, R. Z. Ríos-Mercado, A. M. Alvarez-Socarras y K. de Alba Romenus. En J.E. Fernández, S. Noriega, A. Mital, S.E. Butt y T.K. Fredericks (editores), *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.

- Se realizaron las siguientes presentaciones relacionadas con el presente trabajo de tesis:

  - "A Location-Allocation Heuristic for a Territory Design Problem in a Beverage Distribution Firm". J. A. Segura Ramiro, R. Z. Ríos-Mercado, A. M. Álvarez-Socarrás y K. de Alba Romenus. INFORMS Southwest Regional Conference, Collage Station, Texas, Abril 2008
  - "A location-allocation heuristic for a territory design problem in a beverage distribution firm". J. A. Segura Ramiro, R. Z. Ríos-Mercado, A. M. Álvarez-Socarrás y K. de Alba Romenus. 12th Annual International Conference on Industrial Engineering Theory, Applications, and Practice (IJIE'07), Cancún, México, Noviembre 2007.
  - "Un procedimiento metaheurístico basado en localización-asignación para un problema de diseño de territorios de ventas". J. A. Segura Ramiro. Ciclo de seminarios del PISIS, FIME, UANL, Octubre 2007.
  - "Un algoritmo de localización-asignación para un problema de diseño territorial". J. A. Segura Ramiro y R. Z. Ríos-Mercado. XL Congreso de

la Sociedad Matemática Mexicana, UANL, San Nicolas de los Garza, NL, Octubre 2007.

- "Un procedimiento metaheurístico basado en localización-asignación para un problema de diseño de territorios de ventas". J. A. Segura Ramiro. Ciclo de seminarios del PISIS, FIME, UANL, Mayo 2007.

6.3. Recomendaciones y Trabajo Futuro

Como áreas de oportunidad que continuarían este trabajo de tesis se consideran dos importantes áreas:

6.3.1. Generación de cotas

Otra forma de evaluar la calidad de las soluciones que son obtenidas por la presente metodología de solución es la obtención de cotas que permitan medir directamente la calidad de las soluciones. Como ya se mostró en este trabajo, la cota obtenida a partir de la relajación lineal del modelo es bastante pobre. Sin embargo, la generación de cotas para un problema con las características que se presentaron es un trabajo que implicaría un esfuerzo para otro trabajo de tesis futuro debido a la complejidad combinatorial del problema.

6.3.2. Búsqueda Local

Como se pudo observar en la experimentación computacional, la búsqueda local fue de vital importancia para obtener buena calidad en las soluciones de nuestro algoritmo. En este sentido, existe un mundo de posibilidades que pueden ser exploradas. Desde intentar desarrollar distintos vecindarios hasta derivar métodos más sofisticados como Búsqueda Tabú, Búsqueda por Entornos Variables o Búsqueda Local Iterada.

Bibliografía

[1] S. J. Amico, S. J. Wang, R. Batta y C. M. Rump. A simulated annealing approach to police district design. *Computers & Operations Research*, 29(6):667–684, 2002.

[2] M. Blais, S. D. Lapierre y G. Laporte. Solving a home-care districting problem in an urban setting. *Journal of the Operational Research Society*, 54(11):1141–1147, 2003.

[3] B. Bozkaya, E. Erkut y G. Laporte. A tabu search heuristic and adaptive memory procedure for political districting. *European Journal of Operational Research*, 144(1):12–26, 2003.

[4] A. Brooke, D. Kendrick y A. Meeraus. *GAMS: A User's Guide, Release 2.25*. The Scientific Press, San Francisco, E.U.A., 1992.

[5] 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 F. Chicano E. Alba, F. Herrera, F. Luna, G. Luque y A. J. Nebro, editores, *Actas de las I Jornadas sobre Algoritmos Evolutivos y Metaheurmsticas (JAEM'07)*, pp. 145–153, Zaragoza, España, Septiembre 2007.

[6] S. I. Caballero-Hernández, R. Z. Ríos-Mercado, F. López y S. E. Schaeffer. Empirical evaluation of a metaheuristic for commercial territory design with joint assignment constraints. En J. E. Fernández, S. Noriega, A. Mital, S. E. Butt y T. K. Fredericks, editores, *Proceedings of the 12th Annual International*

Conference on Industrial Engineering Theory, Applications, and Practice (IJIE 07), pp. 422–427. ISBN: 978-0-9654506-3-8, Cancún, México, Noviembre 2007.

[7] F. Caro, T. Shirabe, M. Guignard y A. Weintraub. School redistricting: Embedding GIS tools with integer programming. *Journal of the Operational Research Society*, 55(8):836–849, 2004.

[8] S. P. Coy, B. L. Golden, G. C. Runger y E. A. Wasil. Using experimental design to find effective parameter settings for heuristics. *Journal of Heuristics*, 7(1):77–97, 2001.

[9] P. Grilli di Cortona, C. Manzi, A. Pennisi, F. Ricca y B. Simeone. *Evaluation and Optimization of Electoral Systems.* SIAM, Philadelphia, E.U.A., 1999.

[10] A. Drexl y K. Haase. Fast approximation methods for sales force deployment. *Management Science*, 45(10):1307–1323, 1999.

[11] T. A. Feo y M. G. C. Resende. A probabilistic heuristic for a computationally difficult set covering problem. *Operations Research Letters*, 8(1):67–71, 1989.

[12] J. A. Ferland y G. Guénette. Decision support system for a school districting problem. *Operations Research*, 38(1):15–21, 1990.

[13] B. Fleischmann y J. N. Paraschis. Solving a large scale districting problem: A case report. *Computers and Operations Research*, 15(6):521–533, 1988.

[14] S. L. Forman y Y. Yue. Congressional districting using a TSP-based genetic algorithm. En E. Cantú-Paz, , J. A. Foster, K. Deb, L. David y R. Rajkumar, editores, *Genetic and Evolutionary Computation-GECCO 2003*, Lecture Notes in Computer Science, Vol. 2723, pp. 2072-2083. Springer-Verlag, Berlín, Alemania, 2003.

[15] E. Forrest. Apportionment by computer. *American Behavioral Scientist*, 23(1):23–35, 1964.

[16] K. R. Gabriel y R. R. Sokal. A new statistical approach to geographic variation analysis. *Systematic Zoology*, 18(3):259–278, 1969.

[17] M. R. Garey y D. S. Johnson. *Computers and Intractability. A Guide to the Theory of NP-Completeness*. Freeman, New York, E.U.A., 1979.

[18] J. A. George, B. W. Lamar y C. A. Wallace. Political district determination using large-scale network optimization. *Socio-Economic Planning Sciences*, 31(1):11–28, 1997.

[19] S. W. Hess, J. B. Weaver, H. J. Siegfeldt, J. N. Whelan y P. A. Zitlau. Nonpartisan political redistricting by computer. *Operations Research*, 13(1):998–1008, 1965.

[20] R. S. Howick y M. Pidd. Sales force deployment models. *European Journal of the Operational Research Society*, 48(3):295–310, 1990.

[21] ILOG, S.A. *CPLEX 9.0 Online Documentation*, 2003. http://yalma.fime.uanl.mx/cplex-manual/.

[22] J. Kalcsics, S. Nickel y M. Schröder. Toward a unified territorial design approach: Applications, algorithms, and GIS integration. *Top*, 13(1):1–56, 2005.

[23] O. Kariv y S. L. Hakimi. An algorithmic approach to network location problems, part 2: The p-medians. *SIAM Journal of Applied Mathematics*, 37(1):539–560, 1979.

[24] S. H. Owen y M. S. Daskin. Strategic facility location: A review. *European Journal of Operational Research*, 111(3):423–447, 1998.

[25] R.L. Rardin y R. Uzsoy. Experimental evaluation of heuristic optimization algorithms: A tutorial. *Journal of Heuristics*, 7(1):261–304, 2001.

[26] F. Ricca y B. Simeone. Political districting: Traps, criteria, algorithms and tradeoffs. *Ricerca Operativa AIRO*, 27(1):81–119, 1997.

[27] R. Z. Ríos-Mercado. Computational experience with a reactive GRASP for a large scale commercial territory design problem. En M. J. Geiger y W. Habenicht, editores, *Proceedings of EU/ME 2007 Metaheuristics in the Service Industry*, pp. 72–79, ISBN: 978-3-00-022976-3, Stuttgart, Alemania, Octubre 2007.

[28] R. Z. Ríos-Mercado y E. A. Fernández. A reactive GRASP for a commercial territory design problem with multiple balancing requirements. *Computers & Operations Research*. Aceptado.

[29] N. Solís García. *Evaluación de la Calidad de Métodos de Optimización Exacta para Modelos de Diseño Territorial*. Tesis de Licenciatura, Facultad de Ciencias Físico-Matemáticas, U.A.N.L., San Nicolás de los Garza, N. L., México, Junio 2008.

[30] Daniel A. Spielman y Shang-Hua Teng. Smoothed analysis of algorithms: Why the simplex algorithm usually takes polynomial time. *Journal of the ACM*, 51(3):385–463, 2004.

[31] L. Vargas-Suárez. *Un Procedimiento de Búsqueda Miope Adaptativa y Aleatorizada para la Definición de Territorios de Atención Comercial*. Tesis de Maestría, Facultad de Ingeniería Mecánica y Eléctrica, U.A.N.L., San Nicolás de los Garza, N. L., México, Junio 2005.

[32] L. Vargas-Suárez, R. Z. Ríos-Mercado y F. López. Usando GRASP para resolver un problema de definición de territorios de atención comercial. En M. G. Arenas, F. Herrera, M. Lozano, J. J. Merelo, G. Romero y A. M. Sánchez, editores, *Proceedings of the IV Spanish Conference on Metaheuristics, Evolutionary and Bioinspired Algorithms (MAEB)*, pp. 609–617, Granada, España, Septiembre 2005.

[33] J. C. Williams. Political redistricting: A review. *Papers in Regional Science*, 74(1):13–40, 1995.

[34] A. A. Zoltners y P. Sinha. Toward a unified territory alignment: A review and model. *Management Science*, 29(11):1237–1256, 1983.

AUTOBIOGRAFÍA

JOSÉ ÁNGEL SEGURA RAMIRO

Candidato para el grado de Maestro en Ciencias en Ingeniería de Sistemas
Universidad Autónoma de Nuevo León
Facultad de Ingeniería Mecánica y Eléctrica

Tesis

UN ALGORITMO DE LOCALIZACIÓN-ASIGNACIÓN PARA EL DISEÑO EFICIENTE DE PLANES TERRITORIALES DE USO COMERCIAL

Nacido en Monterrey, Nuevo León. Hijo primogénito del Sr. J. Florentino Segura Rodríguez y la Sra. Margarita Ramiro Santos. Graduado en la Facultad de Ciencias Físico Matemáticas como Licenciado en Ciencias Computacionales (2000-2004). Inició sus estudios de Maestría en Ciencias en Ingeniería de Sistemas en Enero de 2006 con el apoyo del PISIS y una beca de manutención otorgada por el CONACYT.