

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN

FACULTAD DE INGENIERÍA MECÁNICA Y ELÉCTRICA

DIVISIÓN DE ESTUDIOS DE POSGRADO



DISEÑO DE TERRITORIOS COMERCIALES CON  
COSTOS DE RUTEO

POR

JUAN CARLOS SALAZAR ACOSTA

TESIS

EN OPCIÓN AL GRADO DE

MAESTRO EN CIENCIAS EN INGENIERÍA DE SISTEMAS

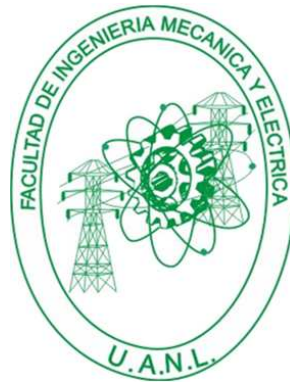
SAN NICOLÁS DE LOS GARZA, NUEVO LEÓN

JULIO 2009

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN

FACULTAD DE INGENIERÍA MECÁNICA Y ELÉCTRICA

DIVISIÓN DE ESTUDIOS DE POSGRADO



DISEÑO DE TERRITORIOS COMERCIALES CON  
COSTOS DE RUTEO

POR

JUAN CARLOS SALAZAR ACOSTA

TESIS

EN OPCIÓN AL GRADO DE

MAESTRO EN CIENCIAS EN INGENIERÍA DE SISTEMAS

SAN NICOLÁS DE LOS GARZA, NUEVO LEÓN

JULIO 2009

**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 «Diseño de Territorios Comerciales con Costos de Ruteo», realizada por el alumno Juan Carlos Salazar Acosta, con número de matrícula 1147627, 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. Satu Elisa Schaeffer

Revisor

---

Dr. Jaime Cano Belmán

Revisor

Vo. Bo.

---

Dr. Moisés Hinojosa Rivera

División de Estudios de Posgrado

San Nicolás de los Garza, Nuevo León, julio 2009

*A mis padres, hermanos y a mi princesa*

# ÍNDICE GENERAL

---

<b>Agradecimientos</b>	<b>XIII</b>
<b>Resumen</b>	<b>xv</b>
<b>1. Introducción</b>	<b>1</b>
1.1. Descripción del Problema . . . . .	1
1.2. Propuesta . . . . .	2
1.3. Motivación y Justificación . . . . .	4
1.4. Objetivos . . . . .	5
1.5. Estructura de la Tesis . . . . .	5
<b>2. Antecedentes y Marco Teórico</b>	<b>7</b>
2.1. Teoría de Grafos . . . . .	7
2.1.1. Definiciones y Conceptos . . . . .	7
2.1.2. Rutas y Métricas de Distancia . . . . .	9
2.1.3. Algoritmos para Obtener la Ruta más Corta . . . . .	11
2.1.4. Funciones de dispersión . . . . .	12
2.2. Diseño Territorial . . . . .	14

---

2.2.1. Diseño de Territorios Políticos y Escolares . . . . .	16
2.2.2. Diseño de Territorios de Ventas . . . . .	20
2.2.3. Diseño de Territorios de Servicios y Otros . . . . .	24
2.2.4. Diseño de Territorios Comerciales . . . . .	26
2.2.5. Diseño Territorial y Ruteo . . . . .	28
<b>3. Planteamiento del Problema</b>	<b>33</b>
3.1. Supuestos y Descripción del Problema . . . . .	33
3.2. Limitaciones del Modelo Euclideo . . . . .	36
3.3. Formulación del Problema . . . . .	37
3.4. Costos de Ruteo . . . . .	39
3.5. Complejidad Computacional . . . . .	42
<b>4. Método de solución</b>	<b>44</b>
4.1. Fase de Construcción . . . . .	45
4.2. Fase de Ajuste . . . . .	49
4.3. Evaluación de Costos de Ruteo . . . . .	50
4.4. Búsqueda Local . . . . .	51
<b>5. Evaluación Computacional</b>	<b>55</b>
5.1. Objetivos . . . . .	55
5.2. Escenarios y Ambiente de Experimentación . . . . .	56
5.3. Ajuste Paramétrico . . . . .	57

---

5.3.1. Experimento A: Parámetro $\alpha$ . . . . .	57
5.3.2. Experimento B: Parámetro $\lambda$ . . . . .	58
5.3.3. Experimento C: Parámetros $\sigma$ y $\gamma$ . . . . .	60
5.4. Evaluación del Algoritmo . . . . .	61
5.4.1. Experimento D: Diferentes Soluciones . . . . .	63
5.4.2. Experimento E: Variación del Parámetro $p$ . . . . .	63
5.4.3. Experimento F: Variación del Parámetro $\tau^a$ . . . . .	65
5.4.4. Experimento G: Mejoras en la Solución del Algoritmo . . . . .	66
<b>6. Conclusiones y Consideraciones Finales</b>	<b>69</b>
6.1. Conclusiones . . . . .	69
6.2. Contribuciones . . . . .	71
6.3. Trabajo a Futuro . . . . .	72
6.3.1. Modelo de Programación Entera . . . . .	72
6.3.2. Mejoras de la Metaheurística GRASP . . . . .	73
6.3.3. Búsqueda Local Especializada . . . . .	73
6.3.4. Consideraciones del Problema . . . . .	73
<b>A. Método de Ramificación y Corte Problema del Agente Viajero</b>	<b>81</b>
A.1. Formulación . . . . .	82
A.2. Desigualdades Válidas para el TSP . . . . .	83
A.2.1. Desigualdades de Eliminación de subtours . . . . .	83
A.2.2. Desigualdades de Peine . . . . .	84

---

A.2.3. Algoritmo de Separación . . . . .	85
A.3. Método de Ramificación y Corte . . . . .	87
A.4. Implementación del Algoritmo . . . . .	88



# ÍNDICE DE FIGURAS

---

1.1. Ejemplo de diseño territorial donde cada territorio es representado por un color. Pueden existir limitaciones naturales como en este caso lo son los lagos que se muestran de color azul. . . . .	2
1.2. Podemos observar a la izquierda que la dispersión con respecto a la distancia euclídeana es totalmente diferente a la dispersión con respecto a la distancia de red. A la derecha vemos que al menos una unidad de la ruta más corta entre alguna pareja de unidades puede no pertenecer al mismo grupo. . . . .	4
2.1. Ejemplo de un grafo. . . . .	8
2.2. Ejemplo de un ciclo hamiltoniano donde las aristas punteadas son los miembros del ciclo. . . . .	9
2.3. Ejemplo de la métrica de distancia de red: Las aristas punteadas representan la ruta más corta entre los nodos $i$ y $k$ . . . . .	10
2.4. Comparación entre la distancia de red y la distancia euclídeana. . . . .	10
2.5. Algoritmo de Dijkstra. . . . .	12
2.6. Algoritmo de Floyd-Warshall. . . . .	13
2.7. Comparación entre diversas funciones de dispersión. . . . .	14

---

2.8. Distancia definida por Chapleau donde el ángulo $\theta$ juega un papel importante para las diferentes ponderaciones de las distancias según la escuela. . . . .	29
2.9. Ejemplo de ciclos elementales. (a) Caras finitas de una grafo euleriano (b) Patrón de tablero de ajedrez (ciclos elementales) (c) Grafo 2-coloreable dual. . . . .	30
3.1. Ejemplo de compacidad territorial. . . . .	34
3.2. Ejemplo de conectividad territorial. . . . .	35
3.3. La distancia de red depende de la solución. . . . .	35
3.4. Casos que se presentan cuando se utiliza un método basado en distancia euclideana para resolver un enfoque basado en distancia de red. . . . .	36
3.5. Representación gráfica de las manzanas de la ciudad. . . . .	38
3.6. Ilustración del concepto de balanceo territorial. . . . .	39
3.7. Ruteo de vehículos. . . . .	40
3.8. Costos de ruteo para cada territorio. . . . .	41
4.1. Algoritmo GRASP. . . . .	46
4.2. Inserción de un nodo al territorio $V_q$ actualmente abierto. . . . .	47
4.3. Fase de Construcción. . . . .	48
4.4. Fase de Ajuste. . . . .	50
4.5. Fase de Evaluación exacta de costos de ruteo. . . . .	51
4.6. Movimiento de la búsqueda local. . . . .	52
4.7. Movimientos heurísticos para la reconstrucción de recorridos. . . . .	53

---

4.8. Fase de Postprocesamiento: Búsqueda Local. . . . .	54
5.1. Infactibilidades con respecto al parámetro $\lambda$ . . . . .	59
5.2. Dispersión con respecto al parámetro $\lambda$ . . . . .	60
5.3. Comparación de la suma de infactibilidades con respecto a los diversos valores de $p$ . . . . .	64
5.4. Comparación de los tiempos de cómputo con los diversos valores de $p$ . . . . .	64
A.1. Ejemplo de construcción de una desigualdad de eliminación de subtour. . . . .	85
A.2. Desigualdad de Peine. . . . .	86
A.3. Ejemplo de construcción de una desigualdad de peine. . . . .	87
A.4. Método de Ramificación y Corte para resolver el TSP . . . . .	89
A.5. Rutina que agrega un corte al problema $P^i$ . . . . .	90

# ÍNDICE DE TABLAS

---

3.1. Resultados al resolver un enfoque basado en distancia euclidea usando las distancias de las rutas más cortas $t_{ij}$ en lugar de las distancias euclideas $d_{ij}$ . Instancias de 500 y 1000 unidades fueron usadas.	37
5.1. Desviación relativa promedio con respecto a la mejor solución para cada valor de $\alpha$ .	58
5.2. Funcionamiento de los parámetros $\sigma$ y $\gamma$ .	61
5.3. Análisis de los tiempos de ejecución del algoritmo GRASP.	62
5.4. Comparación entre los tres tipos de soluciones.	63
5.5. Comparación de los diferentes valores de tolerancia $\tau^a$ .	65
5.6. Comparación de los algoritmos ALG1 y ALG2.	68
A.1. Resultados del Método de Ramificación y Corte en instancias de la TSPLIB.	92

# AGRADECIMIENTOS

---

Agradezco primero a mi Dios, ya que sin Él no hubiera podido realizar este trabajo. También al apoyo incondicional de mis padres Jorge y Elena, de mi familia, de mi Paulina, de mis hermanos (Ana, Migas, Gemela, Aurora, Martha, Lidia y Jorge) y amigos, y en especial el apoyo de mi pastor Mauricio Miranda por su gran ayuda en este largo proceso. Siempre estuvieron presentes, y su amor y cariño infundaron la fuerza para seguir.

De manera especial agradezco al Dr. Roger Z. Ríos Mercado que fungió como mi asesor. Gracias por el tiempo empleado en este proyecto y por la oportunidad de trabajar en su grupo de investigación. Al TDP team por que siempre estuvieron al tanto y aportaron de sus ideas y talentos. A toda la comunidad de PISIS y al plantel de maestros que siempre me apoyaron en cualquier duda y actividad.

Gracias al CONACYT por la beca de manutención económica que recibí durante mi estadía en la maestría. También, agradezco al mismo CONACYT el apoyo otorgado dentro del Programa de Becas Mixtas que me permitió llevar a cabo una estancia de investigación de tres meses en la Universidad Politécnica de Valencia, en Valencia, España. Al apoyo económico que recibí por parte de la Facultad de Ingeniería Mecánica y Eléctrica, así como de la Universidad Autónoma de Nuevo León por medio de sus becas administrativas.

También, agradezco el apoyo otorgado por el CONACYT y la SEP mediante el proyecto SEP-CONACYT 48499-Y y a la Universidad Autónoma de Nuevo León por el proyecto PAICYT CA1478-07 mediante los cuales fue posible presentar este

trabajo en diversos foros nacionales e internacionales.

Mi agradecimiento al comité de tesis, Dra. Elisa Schaeffer y Dr. Jaime Cano, quienes siempre tuvieron tiempo para aclarar mis dudas, además de su interés para el término de mi trabajo.

Un agradecimiento especial al Dr. Rubén Ruiz, al Grupo de Investigación en Sistemas de Optimización Aplicada y al Instituto Tecnológico de Informática de la Universidad Politécnica de Valencia por su apoyo y su espacio para la realización de una parte importante de este trabajo durante mi estancia académica en Valencia, España.

Finalmente, pero no menos importante, agradezco a mis compañeros de generación quienes siempre fueron un gran apoyo en todo este proceso llamado maestría. Gracias por su amistad y por sus ánimos para seguir. No hubiera sido lo mismo sin ustedes, los llevo en el corazón.

# RESUMEN

---

Juan Carlos Salazar Acosta.

Candidato para el grado de Maestro en Ciencias en Ingeniería de Sistemas.

Universidad Autónoma de Nuevo León.

Facultad de Ingeniería Mecánica y Eléctrica.

Título del estudio:

## DISEÑO DE TERRITORIOS COMERCIALES CON COSTOS DE RUTEO

Número de páginas: 93.

Este trabajo aborda un problema de diseño de territorios comerciales mediante un enfoque de distancias red y una evaluación de los costos de ruteo. Una empresa distribuidora de bebidas en la ciudad de Monterrey, México desea construir territorios para sus propósitos de venta y entrega de productos. El enfoque de distancias de red complica la construcción de estos territorios debido al requerimiento de la conectividad. También, es la primera vez que se introduce una evaluación del costo de ruteo en este problema. La evaluación del costo de ruteo fue abordado tanto de manera exacta, como de manera heurística, lo cual nos permitió obtener muy buenos resultados sin gastar grandes cantidades de tiempo.

Debido a la complejidad del problema y al tamaño de instancias que se utilizan,

no es posible resolverlo en un tiempo cómodo y razonable por medio de un algoritmo exacto. Es por esto que decidimos establecer un método de solución heurístico para encontrar soluciones de buena calidad. El método de solución desarrollado explota primero la necesidad de la empresa de obtener territorios compactos. En una segunda fase, se intenta, mediante un proceso de búsqueda local, encontrar un abanico de soluciones que considera tanto la medida de dispersión como al costo de ruteo. Este conjunto de soluciones encontrado proveen de variedad al tomador de decisiones para que éste elija la que más se ajuste a las necesidades de la empresa.

Los resultados computacionales nos dieron la certeza de que esta primera introducción de costos de ruteo a este problema es un gran avance debido a que esta medida era totalmente omitida y realizada justo después de la construcción de los territorios. Ahora, la empresa puede obtener configuraciones territoriales en las que el costo de ruteo es bueno y además, cumplan con la compacidad requerida.

Firma del asesor: \_\_\_\_\_

Dr. Roger Z. Ríos Mercado



## CAPÍTULO 1

# INTRODUCCIÓN

---

### 1.1 DESCRIPCIÓN DEL PROBLEMA

Una empresa distribuidora de bebidas tiene como centro de operaciones el área metropolitana de la ciudad de Monterrey, México. Su objetivo es realizar una distribución eficaz de sus productos de manera que sus clientes se encuentren agrupados y estén lo más cerca posible entre ellos. Cada grupo formado a partir de estas unidades es denominado *territorio*. La empresa envía un camión repartidor a cada territorio para poder realizar sus entregas. Estos grupos deben de cumplir con ciertos requerimientos que la empresa ha fijado para satisfacer sus políticas de mercadotecnia con sus clientes. Dichos requerimientos son los siguientes: el número total de territorios es fijo y dado por la empresa; el camión repartidor debe poder transportarse entre unidades de un mismo grupo sin visitar unidades de otro grupo. Es deseable que cada grupo de unidades tenga un número de clientes y una demanda similar para que la distribución de las tareas en cada territorio sea lo más equitativa posible. Finalmente, el objetivo principal es que los grupos formados tengan una forma geográfica “compacta” para que las unidades de cada grupo se encuentren lo más cerca posible entre ellas. A este problema se le denomina Diseño de Territorios Comerciales. En la Figura 1.1 se observa un ejemplo de un diseño territorial.

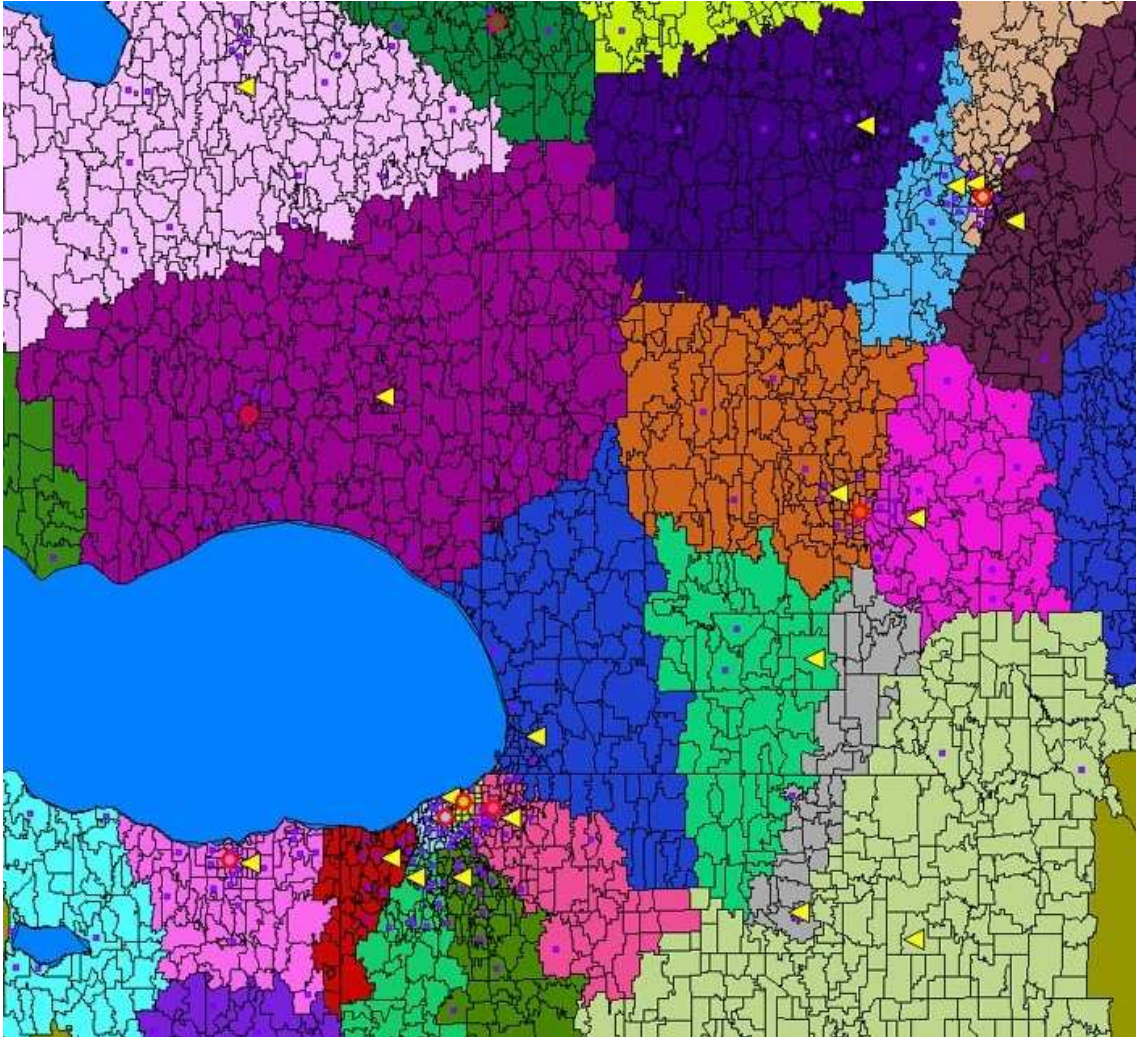


Figura 1.1: Ejemplo de diseño territorial donde cada territorio es representado por un color. Pueden existir limitaciones naturales como en este caso lo son los lagos que se muestran de color azul.

## 1.2 PROPUESTA

En sistemas territoriales, la métrica de distancia más comúnmente utilizada en la literatura para medir la distancia entre dos puntos dados de una red es la distancia euclídeana; sin embargo, una métrica más representativa sería el de la distancia de red que considera como medida la longitud de la ruta más corta entre dos puntos. Es claro que esta medida está más asociada a las distancias de *ruteo* del producto,

que será definido en el Capítulo 2.

En diversas ocasiones, es posible sustituir la distancia euclideana por la distancia de red sin mayores complicaciones. Sin embargo, en nuestro caso específico de diseño territorial, esta sustitución no es válida debido precisamente a las restricciones de conectividad territorial. En trabajos anteriores [44, 51] se ha trabajado con la métrica euclideana debido a que es mucho más tratable en los métodos de solución desarrollados.

En este trabajo, una primera contribución es precisamente hacer un tratado del problema de diseño territorial considerando la métrica de distancias de red, lo cual desde luego, representa un reto mayor desde la perspectiva tanto de modelaje como de desarrollo de técnicas eficientes de solución.

En la Figura 3.1 podemos observar un ejemplo de compacidad donde la parte izquierda muestra un grupo de unidades compacto a diferencia de la parte derecha donde vemos que el grupo tiene una forma más alargada y por ende, las unidades del mismo tienen una lejanía mayor entre ellas.

Otra contribución adicional es la de introducir en la decisión los costos de ruteo. En los trabajos desarrollados anteriormente [5, 6, 44, 46], se ha considerado únicamente el aspecto de minimizar la dispersión territorial, enfocados solamente en la fase de diseño, ignorando los costos de ruteo completamente. Aún cuando para la compañía la compacidad territorial es un criterio mucho más predominante que los costos de ruteo, la idea es proporcionar información adicional que permita evaluar o distinguir los diferentes costos de ruteo entre diferentes planes territoriales con medidas de compacidad similares. De tal forma, se propone un método que en primera instancia se enfoca en construir planes territoriales compactos y en seguida trata de mejorarlos considerando no únicamente el criterio de dispersión sino también el criterio de costo de ruteo.

### 1.3 MOTIVACIÓN Y JUSTIFICACIÓN

Como ya mencionamos, trabajos anteriores hacen uso de la distancia euclídeana en la formación de los territorios. Sin embargo, el uso de un método de solución basado en una métrica de distancia euclídeana no es totalmente efectivo para resolver el problema basado en la métrica de distancia de red.

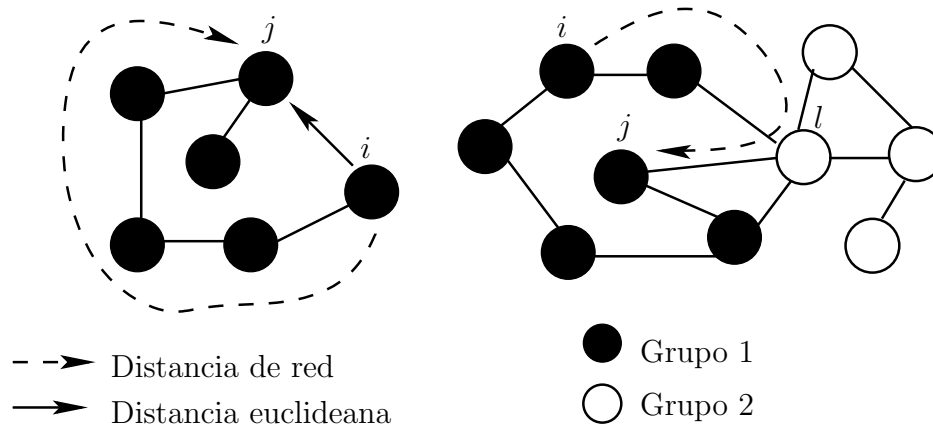


Figura 1.2: Podemos observar a la izquierda que la dispersión con respecto a la distancia euclídeana es totalmente diferente a la dispersión con respecto a la distancia de red. A la derecha vemos que al menos una unidad de la ruta más corta entre alguna pareja de unidades puede no pertenecer al mismo grupo.

La razón de esta aseveración radica en que existen dos casos que ocurren en la solución de un problema cuando han sido resueltos por un método basado en distancia euclídeana. En la Figura 1.2 podemos observar ambos casos. El primer caso se observa cuando un territorio es compacto con respecto a la distancia euclídeana, y no con respecto a la distancia de red, ya que existe una amplia diferencia entre ellas. Puede ser que una unidad se encuentra cerca de alguna(s) otra(s) con respecto a la distancia euclídeana, pero pueden estar muy alejadas entre sí con respecto a la distancia de red. El segundo caso se presenta cuando una unidad pertenece a un territorio, pero no todas las unidades de su ruta más corta, con respecto a alguna(s) otra(s) unidad(es) básica(s), se encuentran en el mismo territorio. Esto ocasionaría

un conflicto debido al requerimiento de conectividad que la empresa ha fijado.

Es de gran importancia mencionar que el uso de la distancia de red facilitará la introducción de este nuevo objetivo en el problema. Además, destacamos que para la empresa es importante tener una evaluación de costos de ruteo, ya que ese es su giro comercial. Resolver una combinación de diseño territorial con ruteo no es nada sencillo, es por esto que recurrimos al área de las heurísticas para contruir una herramienta de solución eficiente.

## 1.4 OBJETIVOS

Principalmente, el objetivo de este trabajo es construir un método de solución que resuelva eficientemente el enfoque de la distancia de red. De esta manera, se estaría contribuyendo con un enfoque más realista en el diseño de territorios comerciales y particularmente en este problema.

La introducción de costos de ruteo a nuestro problema es un mayor reto, debido al escaso trabajo que se ha encontrado en la literatura. Nuestro objetivo, en este ámbito, es introducirlos durante el proceso de decisión de nuestro método de solución.

La diversidad de soluciones es clave para un tomador de decisiones, de esta forma, este trabajo pretende llenar esa necesidad entregando tanto soluciones orientadas a la compacidad como soluciones orientadas a la reducción de costos de ruteo.

## 1.5 ESTRUCTURA DE LA TESIS

Este trabajo se encuentra dividido en seis capítulos que describiremos brevemente. El Capítulo 2 presenta el marco teórico que se encuentra detrás de la elaboración de este trabajo así como lo que hasta el momento se ha realizado en la literatura en relación a nuestro problema. En el Capítulo 3, el problema es descrito de forma más profunda y técnica para que de esta manera, podamos emplear con-

---

ceptos matemáticos que facilitan tanto el análisis como el empleo de técnicas para la resolución del problema. Nuestro método de solución se encuentra descrito totalmente en el Capítulo 4. La implementación computacional del mismo se analiza en el Capítulo 5. Finalmente, el Capítulo 6 contiene las conclusiones de nuestro estudio y las posibles ramas que pueden surgir a partir del mismo. Al final del trabajo, se encuentra un apéndice que describe detalladamente el método empleado de ramificación y corte par la resolución del Problema del Agente Viajero (TSP por sus siglas en inglés), el cual fue utilizado para calcular los costos de ruteo en un plan territorial dado. Es decir, una vez que se conoce un diseño territorial, calcular su costo de ruteo involucra la solución de un problema de agente viajero por cada territorio.

## CAPÍTULO 2

# ANTECEDENTES Y MARCO TEÓRICO

---

## 2.1 TEORÍA DE GRAFOS

La teoría de grafos es el estudio de estructuras (denominadas grafos) que pueden representar diversos tipos de redes del mundo real. Algunos ejemplos pueden ser las carreteras entre ciudades, la internet o una red eléctrica, entre otras. Representar fenómenos o formas del mundo real con grafos, puede ser muy beneficioso debido a las propiedades que poseen estas estructuras. Describiremos algunas definiciones y conceptos básicos (que pueden encontrarse en los textos de Diestel [14], Tutte [49] y Gross y Yellen [24]) que nos serán de utilidad durante el desarrollo de nuestro trabajo.

### 2.1.1 DEFINICIONES Y CONCEPTOS

Un *grafo*,  $G = (V, E)$ , es una estructura formada por dos conjuntos  $V$  y  $E$ , donde  $V$  es el conjunto de nodos y  $E$  el conjunto de aristas. La Figura 2.1 nos muestra un ejemplo de un grafo.

Un *vértice* o *nodo* es la unidad fundamental de la cual se conforma un grafo. Estas unidades pueden representar diferentes conceptos, dependiendo del enfoque que se le dé. Por ejemplo, en este trabajo cada nodo representa una unidad (una manzana de la ciudad), en cadenas de Markov, cada nodo representa un estado de transición, y así podemos describir diversos casos.

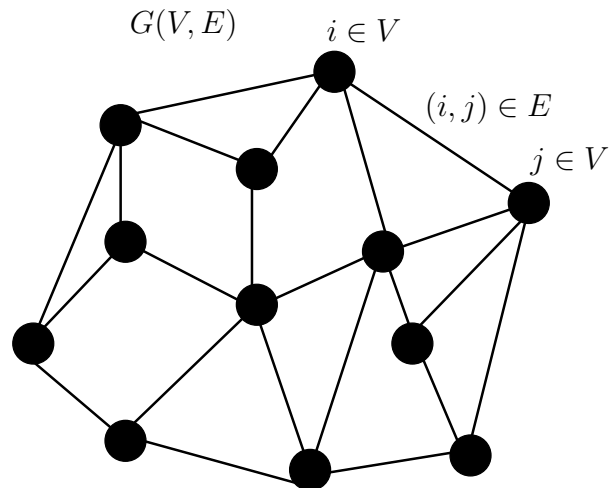


Figura 2.1: Ejemplo de un grafo.

Una *arista* representa un enlace entre dos nodos del grafo. Estos enlaces resaltan una relación que comparten dos nodos. A éstas se les puede asociar un peso dependiendo del significado del enlace. Un ejemplo sencillo es observar que una arista entre dos ciudades existe si hay una carretera que las une. El peso asociado, en este caso, es la distancia en kilómetros entre ambas ciudades. Cada arista es representada como una pareja de nodos; por ejemplo, si entre  $i$  y  $j$  existe una arista, ésta es  $(i, j)$ .

Un *grafo no dirigido* es un grafo constituido por un conjunto de aristas.

Un *subgrafo* es un grafo que está formado por un subconjunto de nodos del grafo original y por un subconjunto de aristas, también del grafo original. Sea  $G = (V, E)$  un grafo, donde  $V$  es el conjunto de nodos y  $E$  es el conjunto de aristas, se dice que  $G' = (V', E')$  es subgrafo de  $G$  si  $E' \subseteq E$  y  $V' \subseteq V$ .

Una *caminata* o *camino* de  $i$  a  $j$ , dentro de un grafo, es un conjunto de nodos que enlazan a  $i$  y a  $j$  a través de sus arcos o aristas.

Un grafo es *conexo* si para cualquiera dos nodos  $i$  y  $j$ , existe al menos un camino que lleve de  $i$  a  $j$ .

Un grafo es *plano* si puede ser dibujado en un plano sin que sus aristas se



intersequen.

Una *ruta* es un camino sin nodos ni aristas repetidas.

Una *caminata hamiltoniana* es un caminata en la que todos los nodos de un grafo son visitados exactamente una vez.

Un *ciclo hamiltoniano*, como el que se muestra en la Figura 2.2, es una caminata hamiltoniana donde las aristas forman un ciclo.

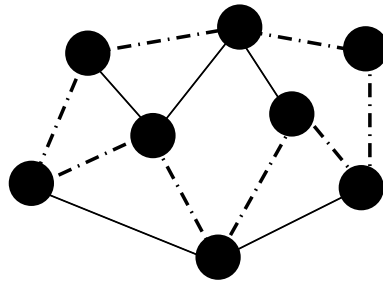


Figura 2.2: Ejemplo de un ciclo hamiltoniano donde las aristas punteadas son los miembros del ciclo.

### 2.1.2 RUTAS Y MÉTRICAS DE DISTANCIA

La relación de lejanía entre dos nodos de un grafo puede ser establecida de diversas maneras. A esto es a lo que denominamos *métrica de distancia*. La métrica más común es la *distancia euclídeana*. Ésta es calculada (dentro del plano de dos dimensiones) utilizando las coordenadas geográficas  $(x, y)$  de los puntos  $i$  y  $j$ . Es representada por la longitud de la recta que los une, de esta manera  $d_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$ . Además, se sabe que la distancia euclídeana es la distancia más corta que puede haber entre dos nodos o puntos.

La *ruta más corta* entre dos nodos,  $i$  y  $j$ , no es más que la ruta de menor costo entre ellos, representada por  $T_{ij}$ . El costo de la ruta es la suma de todos los pesos de las aristas presentes en la misma y es representado por  $t_{ij}$ . Al costo de esta ruta se le denomina *distancia de red*, la cual se muestra en la Figura 2.3.

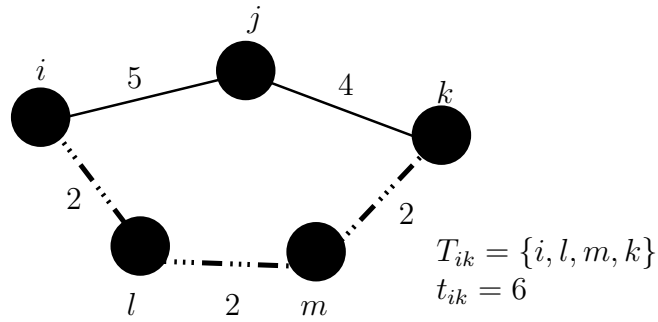


Figura 2.3: Ejemplo de la métrica de distancia de red: Las aristas punteadas representan la ruta más corta entre los nodos  $i$  y  $k$ .

Las distancias de red proveen un costo de distancia más cercano al de la distancia real (en nuestro caso, la distancia entre las manzanas de la ciudad) debido a que manejan el costo de ir de un nodo a otro a lo largo de la ruta en la red. En la distancia euclídeana, por otro lado, no se considera que a partir de un nodo se puede llegar a otro, sólo se considera el costo. En la Figura 2.4 observamos la diferencia entre la distancia euclídeana y la distancia de red.

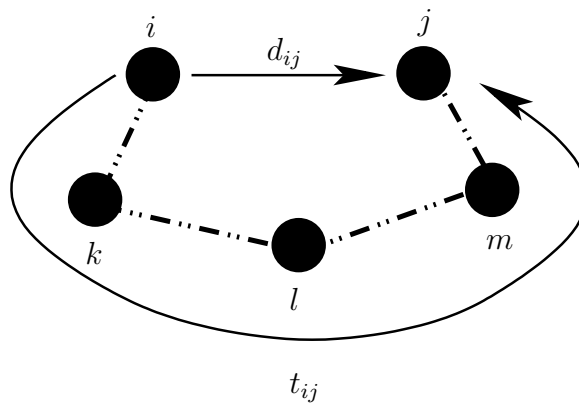


Figura 2.4: Comparación entre la distancia de red y la distancia euclídeana.

### 2.1.3 ALGORITMOS PARA OBTENER LA RUTA MÁS CORTA

Existen diversos algoritmos que permiten calcular la ruta más corta de un nodo a algún otro dentro de un grafo. En esta sección mostraremos dos: el algoritmo de Dijkstra y el algoritmo de Floyd-Warshall.

El *algoritmo de Dijkstra* [15] es un algoritmo para calcular la ruta más corta entre un nodo dado del grafo a todos los demás. La idea consiste en explorar todas las rutas partiendo del nodo fuente  $s$  y que llevan a todos los demás nodos; cuando se obtiene la ruta más corta, el algoritmo se detiene. El único inconveniente es que no trabaja con grafos que contienen aristas o arcos con peso negativo, esto es debido a que el algoritmo selecciona siempre el nodo con distancia menor al origen, y algunos nodos que podrían bajar el costo general de la ruta pueden quedar excluidos en iteraciones próximas al pasar por una arista o arco con costo negativo. Sea  $V$  el conjunto de nodos y  $E$  el conjunto de aristas del grafo  $G = (V, E)$ , sea  $c_{ij}$  el costo de la arista  $(i, j) \in E$ ,  $d_{ij}$  la distancia más corta entre  $i$  y  $j$  y  $f_{ij}$  el nodo padre de  $j$  con respecto al nodo  $i$ ,  $i, j \in V$ . Sea  $s \in V$  el nodo fuente y  $P$  la lista de prioridad de los demás nodos. El pseudo código del algoritmo de Dijkstra se muestra en la Figura 2.5.

El *algoritmo de Floyd-Warshall* [20] calcula la ruta más corta entre todos los pares de nodos del grafo. La metodología consiste en tomar una pareja de nodos y observar si la distancia entre ellos puede disminuir si se pasa a través de cualquier otro nodo convirtiéndose este último en el predecesor. De esta misma manera, se continúa analizando para todos los demás nodos. Al final se asegura la ruta más corta entre cada par de ellos. Sea  $V$  el conjunto de nodos y  $E$  el conjunto de aristas del grafo  $G = (V, E)$ , sea  $c_{ij}$  el costo de la arista  $(i, j) \in E$ ,  $d_{ij}$  la distancia más corta entre  $i$  y  $j$  y  $f_{ij}$  el nodo padre de  $j$  con respecto al nodo  $i$ ,  $i, j \in V$ . El algoritmo de Floyd-Warshall se muestra en la Figura 2.6.

---

*Dijkstra* ( $s$ )

---

**Input:** :  $G = (V, E)$ : Grafo;  $s$ : Nodo fuente**Output:** :  $R(v)$ : Ruta más corta entre  $s$  y  $v$ ,  $\forall v \in V$ 

```

1:  $d_{sv} \leftarrow \infty$ ,  $f_{sv} \leftarrow -1$ ,  $\forall v \in V$ ;
2:  $d_{ss} = 0$ ,  $P = V$ ;
3:  $a \leftarrow s$ ;
4: while ( $P \neq \emptyset$ ) do
5:   for ( $v$  adyacente a  $a$ ) do
6:     if ( $d_{sv} > d_{sa} + c_{av}$ ) then
7:        $d_{sv} \leftarrow d_{sa} + c_{av}$ ;
8:        $f_{sv} \leftarrow a$ ;
9:     end if
10:  end for
11:   $a \leftarrow$  cualquier nodo en  $P$  adyacente a  $s$ ;
12: end while
13: return  $R(v)$ ,  $\forall v \in V$ ;

```

---

Figura 2.5: Algoritmo de Dijkstra.

## 2.1.4 FUNCIONES DE DISPERSIÓN

Una *función de dispersión* mide o evalúa qué tan cerca se encuentran entre sí los nodos de un grafo. Existen diversas funciones que lo hacen de manera distinta:  $p$ -centro,  $p$ -mediana, diámetro, por mencionar las más comunes.

Para nuestro problema, como ya hemos mencionado, utilizaremos la métrica de la ruta más corta para medir la distancia entre dos nodos, así que cuando se mencione la palabra distancia realmente se refiere al costo de la ruta más corta entre dos nodos.

- La función de  $p$ -centro mide el máximo de las distancias entre el centro y los demás nodos. Sea  $c$  el centro del  $k$ -ésimo subgrafo,  $G_k = (V_k, E_k) \subset G = (V, E)$ ,

---

*Floyd-Warshall ()*

---

**Input:**  $G = (V, E)$ : Grafo**Output:**  $R(i, j)$ : Ruta más corta entre  $i$  y  $j$ ,  $\forall i, j \in V$ 

```

1:  $d_{ij} \leftarrow \infty$ ,  $f_{ij} \leftarrow -1$ ,  $\forall i, j \in V$ ;
2:  $d_{ii} \leftarrow 0$ ;
3:  $d_{ij} \leftarrow c_{ij}$ ;  $f_{ij} \leftarrow i$ ,  $\forall (i, j) \in E$ ;
4: for ( $k \leftarrow 1, \dots, |V|$ ) do
5:   for ( $i \leftarrow 1, \dots, |V|$ ) do
6:     for ( $j \leftarrow 1, \dots, |V|$ ) do
7:       if ( $d_{ij} > d_{ik} + d_{kj}$ ) then
8:          $d_{ij} \leftarrow d_{ik} + d_{kj}$ ;
9:          $f_{ij} \leftarrow k$ ;
10:      end if
11:    end for
12:  end for
13: end for
14: return  $R(i, j)$ ,  $\forall i, j \in V$ ;

```

---

Figura 2.6: Algoritmo de Floyd-Warshall.

entonces la dispersión del territorio  $V_k$  está dada por  $:= \max_{i \in V_p} \{t_{ic}\}$ .

- La *p-mediana* es muy similar al *p-centro*, solo que mide la suma de las distancias entre el centro y los demás nodos. De esta manera, sea  $c$  el centro del  $k$ -ésimo subgrafo,  $G_k = (V_k, E_k) \subset G = (V, E)$ , la dispersión del territorio  $V_k$  está dada por  $:= \sum_{i \in V_p} t_{ic}$ .
- El *diámetro* no necesita un centro, simplemente mide la distancia máxima entre todos los pares de nodos. Así,  $\text{diámetro} := \max_{i, j \in V} \{t_{ij}\}$ .

La Figura 2.7 muestra una comparación entre cada función de dispersión antes descritas.

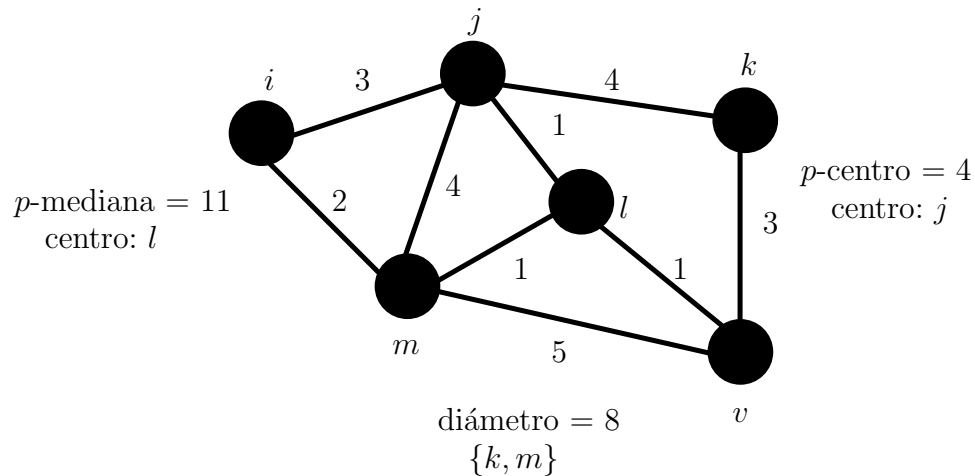


Figura 2.7: Comparación entre diversas funciones de dispersión.

## 2.2 DISEÑO TERRITORIAL

El *diseño territorial* o *diseño de territorios* consiste en agrupar unidades geográficas en segmentos geográficos de mayor tamaño denominados territorios. Los territorios deben estar diseñados de acuerdo a las necesidades y requerimientos del tomador de decisiones. Kalcsics, Nickel y Schröder [33] describen y desarrollan una herramienta con diversos algoritmos que resuelvan problemas de diseño territorial de manera general y básica. También presentan “bloques básicos” para la modelación básica de los problemas de diseño territorial. Generalmente, la unidad básica o unidad de cobertura es la parte esencial de la que están compuestos los territorios. Ejemplos de estas unidades son condados, áreas postales, manzanas, etc. Estas unidades tienen atributos cuantificables asignados denominados actividades.

En algunos casos, los territorios son formados a partir de un centro, el cual puede ser una unidad básica asignada al vendedor en un diseño de territorios de ventas, aunque simplemente puede ser utilizado como una estrategia para construir el territorio sin que tenga un significado trascendente. Generalmente se requiere un número  $p$  de territorios designado por el tomador de decisiones, además de una asignación única de unidades básicas a cada uno.

El problema de diseño territorial puede encontrarse en diversas áreas y aplicaciones: diseño de territorios políticos [22, 31], diseño de territorios escolares [7, 28], diseño de territorios de servicios [4], diseño de territorios de ventas [19, 32], etc.

Cada problema en particular trae consigo objetivos y requerimientos diferentes. Por ejemplo, para un problema de diseño de territorios de ventas la compacidad representa un objetivo importante, pero además, específicamente se desearían territorios en los cuales se maximizará la ganancia. En un problema de territorios comerciales, el balanceo con respecto a diversas actividades es un requerimiento indispensable. La conexidad es importante para los problemas anteriores pero incluso para formar distritos de energía, donde obviamente la red de distribución es sumamente indispensable para poder transportar la energía en cada territorio.

Diversos métodos de solución han sido utilizados para resolver algunos de estos problemas. De éstos destacamos dos, que han jugado un papel muy importante en la historia de los problemas de diseño territorial:

1. Método de localización y asignación. Hess y Samuels [29] propusieron este método de solución heurístico a un problema de diseño de territorios políticos el cual modelaron como un problema  $p$ -mediana. Para este problema, existen  $p$  instalaciones (centros) que deben ser ubicadas a las cuales se les asignará las demás instalaciones (unidades básicas). La fase de localización se refiere a la configuración de los centros, mientras que la fase de asignación puede ser vista como un problema de asignación capacitado. La ventaja de este método radica en que ambas fases son independientes y son implementadas simultánea e iterativamente hasta que se obtenga un resultado satisfactorio. Generalmente, al finalizar la fase de asignación existen unidades básicas que se encuentran asignadas en dos o más centros, llamadas unidades división, y esto genera un problema de separación para asignar estas unidades solamente a un centro. En esta aplicación, se propuso una regla simple de “desempate” en la cual se asignaba la unidad al centro que “poseía” la mayor parte de la misma. Finalmente, encontraron que una tasa de  $m/p \geq 20$  (donde  $m$  es el número promedio

de unidades división por territorio) era suficiente para obtener territorios que no excedieran el 10 % de su tamaño ideal.

2. Métodos divisionales. Podemos mencionar el método de *dicotomías sucesivas* de Forrest [21] y el método de cuñas cortantes de Chance [9]. En el primero, la idea es particionar iterativamente la región en pequeños y cada vez más pequeños subproblemas, donde cada subproblema es definido por un conjunto de unidades básicas y el número de territorios en que este subconjunto debe ser particionado. En el segundo, cada territorio tiene forma de una rebanada de pastel y por ende, toca el centro y la frontera completa del territorio. Observando la forma de los territorios es obvio que este enfoque no toma en cuenta la compacidad de los mismos.

Kalcsiks, Nickel y Schröder [33] también proponen dos heurísticas para la resolución de un problema de diseño territorial. Una de ellas es basada en localización-asignación, en la cual hacen una contribución para resolver el problema de separación mediante programación dinámica, basado en el hecho de que no existen ciclos (se asume que el grafo de separación es un árbol). La segunda, es particionar el problema en subproblemas hasta que cada uno de éstos pueda ser resuelto eficientemente. Este es el método de las dicotomías sucesivas. El hecho de que estos métodos se traten de integrar para resolver cualquier tipo de problema de diseño territorial, provee una gran contribución al área, pero definitivamente existe aún mucho por hacer para esta gran problemática. En el resto de la sección se describirán algunas de las diversas áreas del diseño territorial, además de los diferentes problemas que se tratan tanto en el área comercial, como en el área industrial y gubernamental.

### 2.2.1 DISEÑO DE TERRITORIOS POLÍTICOS Y ESCOLARES

El problema de diseño territorios políticos es visto como el problema de dividir una zona geográfica política, como un estado o ciudad, en zonas de menor tamaño donde se elige a un candidato político. Si bien existen muchos intereses dentro de



la formación de estos territorios, lo que se pretende es tener una partición con una cantidad de población casi equitativa en cada uno. En el pasado, la formación de estos territorios ha sido manipulada por los políticos que se encontraban en el poder para favorecer a algún candidato y de esta manera, seguir oprimiendo a las minorías que surgen dentro de la población del lugar.

En la historia de este problema existe un caso particular muy famoso, en el que el líder político del momento, Elbridge Gerry, gobernador del estado de Massachusetts en los Estados Unidos, propone territorios de forma de salamandra para ganar una ventaja electoral. Debido al nombre del gobernador en aquel tiempo se le denominó a esta práctica como *gerrymandering* [33].

Para evitar la práctica de *gerrymandering*, se han instituido comisiones que actúan independientemente del gobierno. Los criterios con los que una zona es dividida dependen de las especificaciones y leyes de cada región. En la literatura podemos ver las diferencias de los criterios que se utilizan para la formación de territorios, así como de las metodologías de solución utilizadas.

En 1965, Hess et al. [30] abordan un problema de redistribución que por orden de la Suprema Corte, en el estado de Delaware se realizaron tareas de recolectar datos de redistribuciones de otros estados para definir los criterios que se tomarían en cuenta para formar los nuevos distritos. Al final consideraron tres: igualdad de población, conexidad y compacidad. Identificaron a la localización de almacenes y asignación de clientes a ellos como análoga al problema de construir distritos, donde para medir la compacidad utilizaron la suma de los cuadrados de las distancias entre cada persona y su centro, además de que las capacidades de los almacenes deben ser casi iguales (igual población en cada distrito). Para resolver este problema emplearon una heurística que resuelve la localización de almacenes y la asignación de clientes. Básicamente la heurística asigna centros aleatorios y resuelve un problema de transporte para asignar los clientes a cada centro (almacén). Al final se rechazan las soluciones no conexas y se escoge la de mejor compacidad. La aplicación de este método para la construcción de estos distritos en Delaware fue totalmente aceptada por la

Corte debido a que la compacidad fue menor en comparación a sus distritos actuales, además de tener distritos con un número similar de personas.

Un trabajo similar es presentado por Garfinkel y Nemhauser [22] quienes utilizan las unidades poblacionales como elemento indivisible. Su método de solución emplea dos fases: en la primera se generan soluciones factibles con respecto a la conexidad, compacidad y un balanceo de población. En la segunda se encuentra un conjunto de  $M$  distritos factibles que cubren solamente una unidad de población. El objetivo principal es el balance de los territorios, y es en la segunda fase donde se combinan los distritos generados en la primera para minimizar la máxima desviación de población. Para medir la compacidad utilizan una medida de distancia excluyente; de esta manera si la distancia entre los centros de dos unidades  $i$  y  $k$ ,  $d(i, k)$  es mayor que una distancia, ya definida como excluyente,  $e(i, k)$  entonces deben estar asignados en distritos diferentes. El método fue aplicado en tres problemas diferentes en donde el primero fue muy fácil de resolver y no hubo necesidad de aplicar la distancia excluyente. En el segundo caso no obtuvieron una respuesta exitosa debido a que se introdujo un número muy grande de distancias excluyentes dando como consecuencia un tiempo de cómputo muy alto o se obtenían soluciones que no eran factibles. Finalmente, en el tercer problema se resolvió un caso de 37 condados y se obtuvieron 4 distritos.

Como vemos, los criterios de conexidad, compacidad y de balance poblacional son comunes para este tipo de problemas. Al momento de asignar clientes a distritos éstos pueden ser asignados en más de uno (a esto se le denomina comúnmente como nodos división), Hojati [31] tiene como uno de sus objetivos obtener el mínimo número de nodos división. La metodología empleada es de tres fases: en la fase 1 utiliza la relajación lagrangiana de un modelo de localización de almacenes para determinar los centros de los distritos, en la fase 2 utiliza el problema de transporte para asignar las unidades de población a los centros y en la fase 3 resuelve el problema de asignar los nodos división a un solo distrito. Lo que se destaca de este trabajo, además del uso de la relajación lagrangiana para encontrar centros, es el empleo de

una secuencia de problemas de transporte capacitado para resolver la asignación de unidades división.

Ricca y Simeone [43] presentan un modelo de particionamiento de grafos conexos con múltiples objetivos. Tales criterios son: igualdad de población, compacidad y conformidad de fronteras administrativas (tomar en cuenta fronteras administrativas que actualmente existen). Presentan una comparación de cuatro metaheurísticas: descenso, búsqueda tabú, recocido simulado y la aceptación del solterón. De todas éstas, sólo el método del descenso es el que no arroja resultados favorables.

El problema de diseño de territorios escolares es definido como la especificación del grupo de estudiantes que asistirán a una escuela. Los principales criterios que se toman en cuenta para este tipo de problemas generalmente son: capacidad de las instalaciones escolares, igualdad en la usanza de las escuelas, la distancia de viaje de entre los estudiantes y las escuelas, buena accesibilidad y un balanceo racial.

Ferland y Guénette [18] formulan un modelo donde utilizan un grafo con diversas escuelas en él, tomando en cuenta las calles de la ciudad como aristas, donde los nodos representan las intersecciones entre ellas. Así, el problema de formación de distritos escolares es asignar aristas a las escuelas. Los criterios que se deben satisfacer son los siguientes: cada estudiante debe ser asignado a una escuela, la capacidad de cada clase, cada estudiante debe ir a la misma escuela año con año, los sectores escolares deben ser contiguos para que los estudiantes del mismo barrio puedan ser asignados a la misma escuela, además de facilitar la asignación de autobuses escolares. Su aportación es un sistema que pueda ser utilizado año con año. Este sistema recibe de entrada los nodos, arcos y calles, los cuales fueron obtenidos del departamento de la junta escolar de transporte.

Otro problema que se presenta con los distritos escolares es el redistribiteo. Caro et al. [7] lo definen como al ajuste de las fronteras de un sistema escolar. Es muy necesario debido al crecimiento que se presenta en la población. Los autores mezclan la programación entera con las herramientas de los Sistemas Geográficos de Infor-

mación (GIS por sus siglas en inglés) para facilitar su uso a las escuelas debido a los diversos y cambiantes criterios que se pueden encontrar durante un redistrito. Este sistema fue implementado en dos regiones escolares de la ciudad de Philadelphia, EUA: Fels y Olney. Para la primera región se logró asignar a casi el 100% de los estudiantes a su primer o segunda escuela más cercana. Para la región Olney se logró asignar del 78% al 93% de los estudiantes a su primera o segunda escuela más cercana.

### 2.2.2 DISEÑO DE TERRITORIOS DE VENTAS

Los territorios de ventas generalmente son territorios a los cuales se les asigna un vendedor que tiene que recorrerlos por completo. Esta asignación implica que diversos criterios deban ser considerados: la compacidad, el número de clientes, la demanda total, el esfuerzo de cada vendedor, territorios potenciales de ventas, disminuir la distancia que un vendedor necesita recorrer, maximizar la ganancia, obtener territorios balanceados con respecto a una o más actividades, por mencionar algunos. Este problema se ha visto en la literatura incluso desde los años 60's cuando la necesidad de crear dichos territorios para cubrir la demanda surgía, además de reducir los grandes costos que se generaban en responder a una venta.

El diseño de territorios de ventas también puede ser establecido para vendedores que hacen llamadas para realizar sus ventas. Shanker, Turners y Zoltners [47] abordan este tipo de enfoque e introdujeron un procedimiento para resolver dos problemas que surgían en la época: la definición de fronteras, determinar cuales clientes son asignados a cada vendedor, y la frecuencia de llamadas, qué tan seguido un vendedor debe llamar a un cliente. Como se observa, estos problemas son muy diferentes, mientras uno es más compatible para las decisiones administrativas, el segundo depende de las habilidades del vendedor. El procedimiento consiste en construir primero, una matriz de mercado  $F$  de dimensiones  $m \times p$ , clientes por vendedores, donde cada posición  $f_{ij}$  es el esfuerzo que hace el vendedor  $j$  para hacer un venta con el cliente  $i$ . Asociada a esta matriz de esfuerzo  $F$  otra matriz se le suma,

$B$ , en la cual el término  $b_{ij}$  representa si el vendedor  $j$  llama al cliente  $i$ . La columna de  $F$  que define el territorio del vendedor  $j$  no es más que la intersección de  $B$  con otra matriz,  $B^j$ , en la cual se encuentran muchas más asignaciones territoriales de clientes candidatas para el vendedor  $j$ . De esta manera  $b_k^j$  es un vector columna, donde cada una define un territorio potencial para el vendedor  $j$ . Cada columna de la matriz  $B$  puede ser caracterizada por  $R^j$ , la respuesta total esperada de ventas obtenida del vendedor  $j$  con el esfuerzo descrito por la matriz  $F$  para sus clientes. A partir de este concepto, también podemos describir  $R_k^j$  como la respuesta total esperada de ventas para cada territorio candidato en la matriz  $B^j$  del vendedor  $j$ . A partir de aquí, la metodología sigue los siguientes pasos:

1. Generar un conjunto  $N_j$  de asignaciones territoriales  $B^j$  para el vendedor  $j$ . Las restricciones de carga de trabajo son introducidas aquí, (a) utilizando un juicio administrativo ó (b) usando un objetivo aproximado para la demanada potencial.
2. Evaluar la respuesta de venta  $R_k^j$  para cada territorio candidato y determinar el nivel óptimo de llamadas para cada candidato.
3. Repetir los pasos anteriores para  $j = 1, 2, \dots, p$
4. Seleccionar los  $p$  territorios de la matriz de mercado que maximizan el total de ventas para cada vendedor.

Los modelos de partición de conjuntos son comúnmente utilizados para representar y resolver un problema de territorios de ventas. Kalcsics et al. [32] proponen un modelo orientado a localización de instalaciones, formulado como un problema  $m$ -mediana capacitado. Para cada vendedor es asignada una unidad de cobertura de ventas (SCU) y así mismo, cada territorio es formado por un subconjunto de SCU's. El objetivo es encontrar un número dado  $m$  de nuevos territorios de ventas minimizando la suma de distancias de los SCU's hacia la ubicación correspondiente del vendedor dentro del territorio al cual pertenecen. La medida de los territorios

está limitada tanto inferior como superiormente con respecto a una actividad. Para resolver problemas a gran escala desarrollaron una heurística donde la posición inicial (SP) de los vendedores es obtenida resolviendo un problema continuo de localización de  $m$ -instalaciones. El procedimiento heurístico está basado en un intercambio combinado y en una búsqueda de entornos variables (VNS): después de haber obtenido la configuración inicial de SP's, se comienza a intercambiar SCU's del mismo territorio con el SP actual para tratar de mejorar la solución (se utilizan SCU's del mismo territorio ya que la configuración inicial ya es muy buena). La nueva solución es aceptada si aunque sea infactible es más balanceada, o ambas es factible y la suma de las distancias es menor. Los resultados computacionales muestran buenas soluciones cuando se emplea la VNS para problemas de gran escala.

Hess y Samuels [29] realizan una comparación entre las diferentes medidas de actividad debido a que construyeron una heurística (GEOLINE) que recibe de entrada el tipo de actividad y los centros iniciales. La complejidad de este algoritmo depende del cálculo que se realiza con las diferentes actividades. Mencionan que su algoritmo GEOLINE no construye territorios geográficamente compactos, pero si los construye compactos en relación a su medida de actividad. Entre las actividades que compararon se encuentran: el número de llamadas de ventas, el potencial de ventas, las ventas y algunas combinaciones ponderadas entre ellas. Las implementaciones de su heurística permitió a las compañías encontrar beneficios sin necesidad de realizar un estudio completo, además de darles la oportunidad de utilizar diferentes actividades.

Zoltners y Sinha [53] revisaron algunas formulaciones de este problema hasta 1983 y proponen una formulación donde cada territorio tiene la estructura de un árbol jerárquico de adyacencia de SCU's y donde el centro es el nodo raíz. Las aristas representan vías reales factibles. Para crear estas estructuras, los autores desarrollaron una base de datos que puede leer las carreteras interestatales de los Estados Unidos. También, hacen uso de la ruta más cortas entre las SCU's para utilizar con eficiencia la base de datos. El uso de las rutas más cortas establece un

rígido acceso y uso restricciones de contigüidad. Los posibles objetivos pueden ser: minimizar el tiempo de viaje, las interrupciones o maximizar las ganancias.

Un problema de territorios de ventas a gran escala es abordado por Fleischmann y Paraschis [19] en un problema muy particular en Alemania a finales de los 80's. Los territorios deben cumplir tres requerimientos: ser lo más uniforme posible entre ellos, ser de forma compacta y disjuntos entre sí y finalmente, el procedimiento de solución debe ser lo más automático posible, esto es, sin necesidad de un gran esfuerzo para adaptaciones posteriores. Utilizan un enfoque de localización-asignación, lo cual apunta a que el último requerimiento se convierte en la resolución eficiente de nodos división después de la fase de asignación. Para resolver eficientemente esta fase, primero realizan asignaciones parciales hasta que el criterio de balance sea roto. Después, implementan un teorema que permite fijar la mayoría de las asignaciones parciales. Este procedimiento se repite hasta que todos los nodos división hayan sido resueltos.

Finalmente, Drexl y Haase [16] trabajan un problema muy interesante sobre la ubicación estratégica del equipo de ventas que a su vez, involucra la resolución simultánea de cuatro subproblemas: determinación del número de vendedores, localización de los vendedores, diseño de territorios de ventas y la asignación de recursos de ventas. En su trabajo modelan el problema cubriendo los cuatro subproblemas e implementan métodos de aproximación capaces de resolver instancias del mundo real a gran escala. Estos métodos de aproximación se resumen en una heurística desarrollada por ellos: primero realizan una localización inicial de vendedores para un número dado de ellos. Después intentan alternar estas localizaciones para obtener mejores soluciones. La construcción de territorios de ventas se realiza a partir de aquí, lo hacen de manera voraz asignando SCU's al vendedor al cual le resulte menos costoso en la función objetivo. Este procedimiento se repite para un número diferente de vendedores. Finalizada la construcción, realizan intercambios de SCU's para mejorar la solución. La asignación de recursos de ventas se realiza una vez que los territorios hayan sido establecidos.

### 2.2.3 DISEÑO DE TERRITORIOS DE SERVICIOS Y OTROS

El diseño de territorios de servicios generalmente considera los servicios médicos, de bomberos, de la policía, etc. Los criterios varían según cada caso en particular, pero la cantidad de población es un factor determinante y común en estos casos.

La cuestión de asignar poblaciones a los hospitales es de importancia trascendental, ya que la vida de millones de personas puede depender de la distancia donde se encuentran con respecto al hospital. Pezzella y Nicoletti [41] enfrentan un problema de construir distritos de salud. Para representar la región, se utiliza un grafo  $G = (V, E)$ , donde las unidades básicas que serán combinadas, los condados, son representadas por el conjunto de nodos  $N$ . Los arcos entre dos nodos son el camino más corto, si existe, sin pasar a través de otros condados. De estos  $n$  condados,  $m$  tienen un hospital. De esta manera, cada distrito hospitalario posee al menos un condado que contiene un hospital. Para formar los distritos de salud, utilizan dos pasos: en el primero, a través de programación matemática, determinan los distritos hospitalarios óptimos tomando en cuenta la demandada y la capacidad de camas de los hospitales. El segundo paso es formar distritos de salud agregando distritos hospitalarios tomando en cuenta cotas inferiores y superiores de población de cada uno.

Bertolazzi, Bianco y Ricciardelli [4] tratan con la construcción de distritos de emergencias (el departamento de bomberos y las ambulancias) en la ciudad de Roma. Lo que se desea es asignar unidades de emergencia a zonas (distritos) de manera que puedan responder efectivamente cuando se les necesiten a menos que se encuentren ocupadas. Las unidades básicas son llamadas átomos y tienen asociado una fracción de llamadas de alarma generadas en la región considerada. Las unidades de emergencia no suelen estar en un solo lugar así que se hace uso de una matriz de localización para describir la distribución de las unidades. Finalmente, lo que se busca es asignar cada llamada de alarma a cada estación y de esta manera, se forman los distritos de emergencia. Proponen un modelo de programación entera el



cual resuelven mediante un algoritmo iterativo.

Para los departamentos de policía es importante particionar las ciudades en territorios denominados comunmente jurisdicciones. D'Amico et al. [12] trabajan con un caso de estudio para fijar territorios (jurisdicciones) donde para cada uno, un óptimo número de patrullas es asignado, teniendo en cuenta diversos criterios que convierten a cada territorio en “bueno”.

Un problema muy particular es el denominado diseño territorial del sistema de energía eléctrica (EPDP por sus siglas en inglés). Bergey, Ragsdale y Hoskote [3] abordan este problema debido a la creciente demanda y competencia de este sector en los países occidentales. Existen dos factores muy importantes dentro de este particular problema: la energía fluye a través de la red física y no existe manera de almacenar la energía eléctrica. Esto trae como consecuencia la existencia de una coordinación para satisfacer la demanda en tiempo real. Es por esto que debe haber algunas consideraciones con que lidiar:

1. Coordinar el envío y entrega para la red centralizada con los proveedores y compradores descentralizados.
2. Definir derechos de propiedad que sean divisibles para una red común indivisible de transportación.
3. Establecer políticas de precio (precio nodal o precio por zona).
4. Facilitar la competencia en el nivel local de distribución.

Los puntos de suministro en la red eléctrica se les representa como nodos; a partir de estos puntos se distribuye la energía a los vecindarios, comercios y zonas industriales. Debido a los altos costos de distribución, mantenimiento y demás gastos relacionados es requerido que las zonas a las que se les distribuye la energía sean administradas por una sola entidad. Este EPDP está modelado como un problema bi-objetivo: la desviación de la ganancia total del diseño territorial y la compacidad.

Un requerimiento necesario es la conexidad de los territorios, esto para la distribución de la energía eléctrica. El método de solución que emplean es un algoritmo genético donde el operador de mutación utiliza una serie de movimientos nodales (un nodo es movido a otro territorio vecino). Esto limita el número de movimientos a nodos frontera (nodos que se encuentra en la frontera de un territorio con otro). La aplicación del método se realizó en el sistema eléctrico de Ghana donde se consideraron cuatro criterios: ganancia potencial, dispersión del área geográfica, dispersión de los clientes y claridad en la fijación de los límites de los distritos.

#### 2.2.4 DISEÑO DE TERRITORIOS COMERCIALES

Los problemas de diseño de territorios comerciales son similares a los de territorios de ventas. En este caso no es asignado un centro para algún vendedor sino simplemente es un medio utilizado para construir los territorios y tener una medida de compacidad. Específicamente este problema ha sido tratado para construir territorios para una empresa distribuidora de bebidas. Diversos enfoques con diferentes requerimientos son descritos en este apartado.

Una empresa embotelladora tiene la necesidad de realizar un diseño territorial para la distribución de sus productos, Vargas-Suárez, Ríos-Mercado y López [51] abordan este problema. Este problema consta de tres medidas de actividad asociadas a cada unidad básica: número de clientes, demanda y carga de trabajo, además de asignaciones forzosas de algunas unidades a ciertos territorios. No se considera la compacidad, además de que el número de territorios no es fijo, sino que se mantiene como una variable del problema. Para resolver este problema implementaron una metaheurística GRASP donde la finalidad es minimizar la desviación con respecto a metas determinadas de cada una de las actividades.

Ríos-Mercado y Fernández [44] estudian una extensión de este problema y proponen una metaheurística GRASP como método de solución. Como necesidad, la empresa desea que los territorios que se construyan tengan un balance con respecto

a tres medidas de actividad: La demanda, el número de clientes y la carga de trabajo. Para poder distribuir los productos, se requiere además que los territorios sean conexos y en gran medida compactos. El número total de territorios es dado por la empresa. Cabe señalar que este enfoque es basado en distancia euclídeana y la medida de dispersión utilizada es la de  $p$ -centro. El método GRASP consta de la fase de construcción y la fase de post-procesamiento. En la primera, dada una iteración y a partir de un territorio parcial, se decide si se añade una unidad básica no asignada o se cierra el territorio y se abre uno nuevo. Al final de la construcción puede que se hayan abierto más de los territorios deseados o incluso en menor cantidad. Si este es el caso, se aplica una fase de ajuste donde se dividen o se mezclan los territorios hasta llegar al número deseado. La fase de postprocesamiento describe una búsqueda local en la cual una unidad básica  $i$  es movida al territorio de la unidad  $j$ , donde  $(i, j)$  representa un arco del grafo original, y de esta manera se mantiene la conexidad.

Caballero-Hernández, Ríos-Mercado y López [6] y Caballero-Hernández et al. [5] introducen las restricciones de asignación conjunta. Estas restricciones aseguran que ciertas parejas de unidades básicas deben ser asignadas en un mismo territorio. Proponen un método GRASP que debido a las restricciones de asignación conjunta, emplean una fase de preproceso para calcular las rutas más cortas entre las unidades básicas que deben ser asignadas en un mismo territorio y así elegir una de ellas aleatoriamente. La fase de construcción comienza con estas rutas calculadas como territorios en sí. Los nodos que no fueron asignados a estas rutas representan un “territorio unitario”. Con esto se pretende mantener la conexidad en el desarrollo del algoritmo. En la fase de construcción los territorios se van fusionando hasta obtener el número deseado de ellos. Emplean una búsqueda local similar a Ríos-Mercado y Fernández [44], pero también introducen otra de intercambio de parejas de nodales.

A diferencia del enfoque de  $p$ -centro, Segura-Ramiro et al. [46] introducen la modelación del problema como  $p$ -mediana. Otra diferencia muy significativa es el método de solución empleado: una heurística basada en localización y asignación de

instalaciones. En la fase de localización se encuentra una configuración de centros, esto proporciona que los territorios que se encuentren sean de buena calidad. Esta fase es aplicada nuevamente después de la fase de asignación hasta que no haya mejora. En la fase de asignación, se reformula el modelo y se relaja la integralidad de las variables, de esta manera lo que se obtiene es simplemente un problema de transporte. Debido a la continuidad de las variables, una unidad básica pudo haber sido asignada a dos o más territorios, a esto se le conoce como problema de separación. Para resolver este problema toman dos aspectos a seguir: proponen un procedimiento heurístico que intenta en primera instancia reparar conexidad si acaso hubiera alguna violación, y posteriormente se fija en violaciones de balanceo y dispersión para determinar la asignación de las unidades divididas. Una asignación que haga un territorio no conexo es descartada inmediatamente y el objetivo es obtener territorios compactos.

### 2.2.5 DISEÑO TERRITORIAL Y RUTEO

El *ruteo de vehículos* es la asignación de rutas de clientes a una flotilla de vehículos repartidores donde cada ruta solo es servida por un vehículo y además, comienza y termina en el centro de distribución [48]. El diseño territorial y el ruteo tienen mucho en común, pero existe una diferencia fundamental: el diseño territorial envuelve criterios más globales, los cuáles lo relacionan a servir objetivos a niveles administrativos y gerenciales. Además, el diseño territorial proporciona un marco de referencia para el ruteo. En sí, el ruteo proporciona un diseño territorial, pero debido a su naturaleza muchos requerimientos del diseño territorial quedan fuera de su alcance.

Los problemas de diseño de territorios escolares pueden tener muy cercana relación con los problemas de ruteo debido a la necesidad de trazar las rutas de los autobuses escolares. Chapleau, Ferland y Rousseau [10] determinan primero segmentos con el número de estudiantes y después una ruta y las paradas de los autobuses son establecidas. En la red, los estudiantes se localizan en las aristas (calles) y cada

nodo representa una parada de autobús. Todos los estudiantes de una arista son asignados al mismo nodo más cercano. Los empates son rotos al azar. El objetivo principal es minimizar el número de paradas de autobús y la distancia recorrida. Para esto, definen una métrica de distancia entre dos nodos (2.1) diferente a la euclideana (ver Figura 2.8) con la cual comienzan a construir segmentos elípticos con el eje mayor en dirección a la ubicación de la escuela. El número de segmentos generados fue obtenido dividiendo el número de estudiantes con la capacidad de la ruta. Al principio permitieron que los segmentos violaran la capacidad de la ruta hasta cierto punto, pero en el siguiente paso aplicaron algoritmos de intercambio para obtener una solución factible y en segundo lugar para obtener segmentos compactos. La medida de compacidad utilizada fue la  $p$ -mediana.

$$dis(x, y) = \begin{cases} \overline{xy}^2 \left\{ 1 + \lambda \left[ 1 - \frac{(\overline{xy}^2 - \overline{yz}^2 - \overline{xz}^2)}{2 \cdot \overline{xy} \cdot \overline{xz}} \right] \right\}, & \text{si } \overline{xy} \leq \overline{xz} \\ \overline{xy}^2 \left( 1 + 2\lambda \frac{\overline{yz}}{\overline{xy}} \right), & \text{si } \overline{xy} > \overline{xz} \end{cases} \quad (2.1)$$

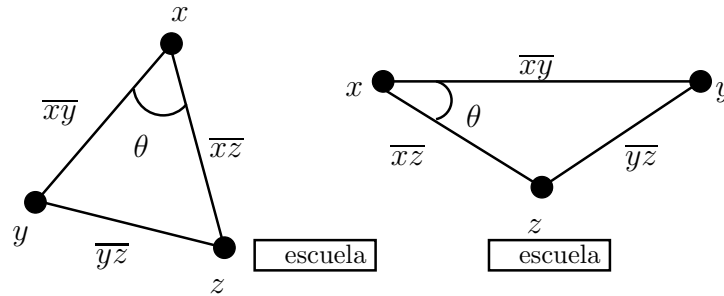


Figura 2.8: Distancia definida por Chapleau donde el ángulo  $\theta$  juega un papel importante para las diferentes ponderaciones de las distancias según la escuela.

Muyldermans et al. [39] abordan un problema de esparcimiento de sal en combinación con diseño territorial. El problema de esparcimiento de sal puede verse como un problema de ruteo con arcos capacitado, debido a que los camiones que distribuyen la sal tienen una capacidad limitada. Este problema surge en las ciudades en las que el hielo se acumula en las calles y carreteras. Las rutas para el esparcimiento

de la sal son organizadas típicamente por distritos o territorios y son categorizadas en dos tipos: cobertura preventiva y cobertura sanitaria. La cobertura preventiva se realiza antes de que las calles estén cubiertas de hielo y la cobertura sanitaria por ende, se realiza cuando las calles están ya cubiertas por hielo. Es por esto que se requiere el doble de la cantidad de sal. Cattrysse, Van Oudheusden y Lotan [8] y Van Oudheusden, Cattrysse y Lotan [50] demostraron que para el problema de esparcimiento de sal, el diseño territorial provee un mejor desenvolvimiento para el ruteo. La metodología de solución propuesta por Muyldermans está basada en el enfoque de ciclos elementales. Para formar los ciclos elementales se debe tomar un grafo plano euleriano [34] y después aplicar el *patrón del tablero de ajedrez*, asignar con un color a un ciclo elemental y a los ciclos que tienen una arista en común con éste se les asigna otro color, y de esta manera formar una partición. Esta partición del grafo es caracterizada por una gran número de ciclos muy pequeños en donde cada arista pertenece solo a un ciclo (solo tomando en cuenta un color) como indica la Figura 2.9. Se etiqueta cada cara (partición) formada en el grafo y se selecciona un

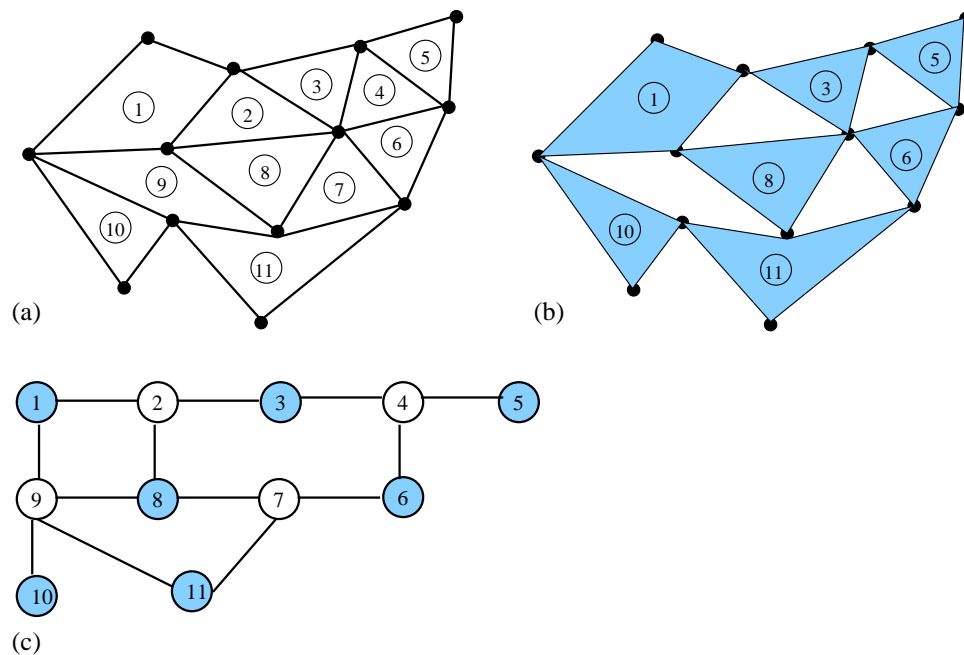


Figura 2.9: Ejemplo de ciclos elementales. (a) Caras finitas de una grafo euleriano (b) Patrón de tablero de ajedrez (ciclos elementales) (c) Grafo 2-coloreable dual.

conjunto de caras de tal manera que cada arista pertenezca solo a un ciclo. A estos ciclos seleccionados (caras) se les denomina ciclos elementales. A partir de aquí, se puede construir un grafo dual en el cual cada cara del primal corresponde a un nodo en el dual. Una arista en el dual implica que dos caras en el primal tienen al menos una arista en común. El grafo dual es por ende planar y debido a que el primal tiene todos sus nodos de grado par, no tiene circuitos con un número de aristas impar. Es a partir de este grafo dual que se aplican cuatro etapas de la metodología de solución donde a cada centro de distribución le son asignados los ciclos elementales y de esta manera se construyen los territorios que a su vez incluyen las rutas para la distribución de la sal en las carreteras.

En un problema para diseñar rutas de transportes escolares, Haugland, Ho y Laporte [28] consideran construir territorios para problemas de ruteos de vehículos, es decir, cada territorio es servido por un solo vehículo. Este enfoque es utilizado debido a que las demandas son consideradas estocásticas. La demanda es conocida después de que los territorios son formados y se construye un plan de viaje dentro de cada uno. Los territorios se mantienen fijos debido a que es considerado que las fronteras de un territorio tienden a permanecer sin cambio a lo largo del tiempo a diferencia de la demanda que cambia a diario. El problema es presentado de manera que en el primer estado las decisiones de ruteo (demandas) son modeladas de forma estocástica, de aquí el objetivo es construir territorios de manera que los costos de ruteo son minimizados. Además, puede ser modelado como un problema estocástico de dos etapas con recurso introduciendo una cota superior del costo total de ruteo en cada territorio. Para resolver el problema utilizan un algoritmo de búsqueda tabú y una heurística de multiestrella. Los movimientos de la búsqueda local (intercambio simple e intercambio de parejas) son de nodos frontera para mantener la conexidad y respetando la cota superior de la demanda.

El problema de transporte fue utilizado por Marlin [37] para resolver un problema de diseño territorial en el cual se pretende que un número fijo de consultores de prevención de pérdidas (LPC's) presten servicio a empresas en una gran área

comprendida en Estados Unidos y Canadá. Cada territorio es denominado “recorrido” y representa un año de trabajo de un hombre, mientras que a cada empresa se le llama facilidad de riesgo. Para el desarrollo de los recorridos tomaron en cuenta dos pasos: asignar facilidades de riesgos a un recorrido, y secuenciarlas para propósitos de calendarización y asignación. Como el mismo autor menciona este problema es análogo a un problema de ruteo, pero al momento de secuenciar las facilidades, la asignación a cada recorrido ya fue hecha. El problema de transporte es reformulado y resuelto como un problema de flujo de costo mínimo mediante el algoritmo “fuera de la olla” [1]. Al final es necesario resolver un problema de separación para que cada facilidad de riesgo quede asignada solamente en un recorrido.

Hardy [27] realizó una comparación entre la construcción de territorios para asignar rutas y el método de ruteo de Clarke-Wright que no es más que el diseño de rutas que deben ser seguidas para satisfacer la demanda de un área específica. Comparó ambos enfoques y encontró que los resultados eran muy similares. El número de rutas necesario para ambos fue el mismo, aunque la desviación del tiempo de recorrido fue mejor para el diseño territorial.



## CAPÍTULO 3

# PLANTEAMIENTO DEL PROBLEMA

---

En este capítulo se presenta con mayor detalle al problema abordado en esta tesis. Primeramente se describe el problema y los supuestos tomados en consideración. Posteriormente, se presenta la formulación combinatoria del problema, seguida por una discusión de sus características distintivas y su complejidad computacional.

### 3.1 SUPUESTOS Y DESCRIPCIÓN DEL PROBLEMA

La compañía desea agrupar a sus clientes para distribuir sus productos de manera eficiente. Para conseguir esto, los clientes fueron agregados en una unidad geográfica: una manzana de la ciudad. De esta manera, una manzana resulta ser más efectiva en el sentido del número total de unidades básicas a agrupar. Debido a esta primera agregación de clientes, cada unidad básica tiene asociada dos actividades: el número total de clientes y la demanda total en dicha manzana. Cada grupo formado de unidades básicas es denominado *territorio*. El objetivo de la compañía es que estos territorios sean *compactos*, esto es, que las unidades básicas que pertenecen a un territorio se encuentren lo más cerca posible entre ellas, tal y como se ilustra en la Figura 3.1, para que de esta manera se reduzca el recorrido de los camiones distribuidores. Otra característica de la compactación es que se reduce el número de unidades básicas que son adyacentes con unidades de otros territorios, lo cual es de utilidad para la empresa ya que le permite establecer diferentes estrategias de ventas

en los diversos territorios sin que crezca la inconformidad de los clientes.

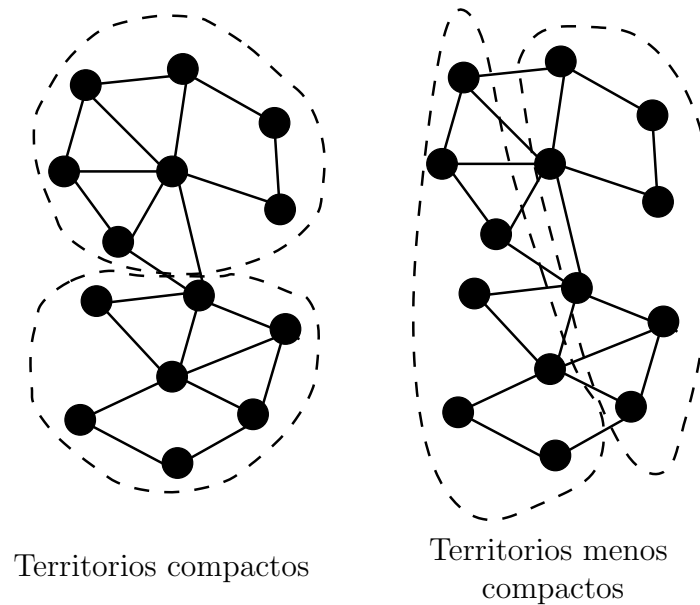


Figura 3.1: Ejemplo de compactación territorial.

Además de la compactación, es requerido que los territorios sean balanceados con respecto a las dos actividades antes mencionadas. Es difícil obtener territorios perfectamente balanceados y debido a esto, introducimos un parámetro de tolerancia que mide la desviación máxima permitida con respecto al valor meta de ambas actividades. Otro requerimiento importante es el de conectividad territorial, esto es, se desea formar territorios de tal manera que a partir de una unidad básica se puede llegar a cualquier otra que se encuentre contenida en el mismo territorio sin necesidad de salir del mismo, como se ilustra en la Figura 3.2. La conectividad es esencial debido a la repartición de los productos.

Generalmente, en los problemas de diseño territorial se utiliza una métrica de distancia euclídeana de la cual ya se hablo con anterioridad. Este problema aborda el uso de la distancia de red, dada por la ruta más corta entre unidades básicas de la red de distribución, como métrica de distancia. En problemas típicos de localización el uso de métricas de distancia, como los son la distancia Euclídea y la distancia de red, suele ser indistinto debido a que no manejan restricciones de conectividad territorial.

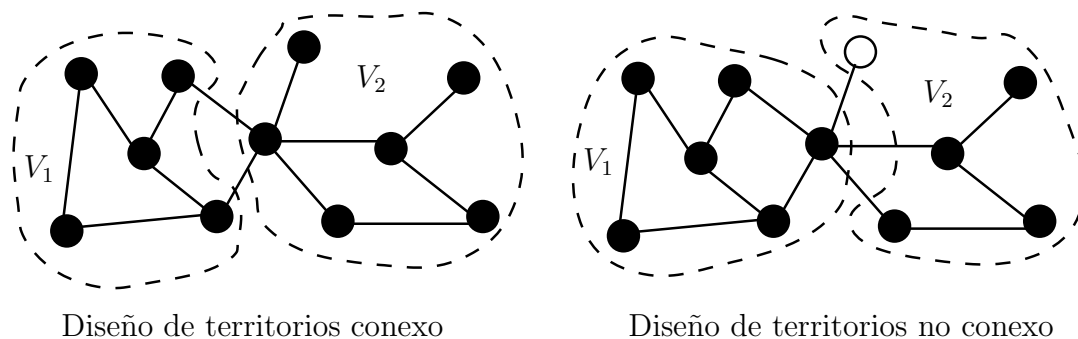
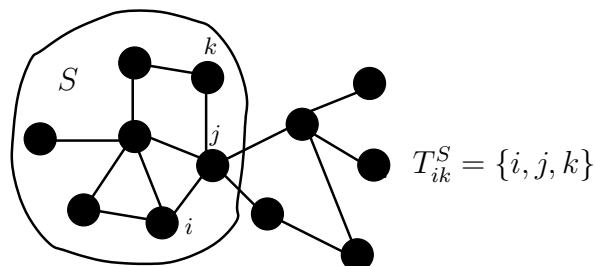


Figura 3.2: Ejemplo de conectividad territorial.

En el modelo propuesto en esta tesis, el incorporar la métrica de distancia de red, con la restricción de conectividad que restringe a que esta ruta más corta esté contenida en el territorio, provoca una complejidad algorítmica considerable debido a que esta distancia de red pasa a ser un parámetro dependiente de la solución. Esto lo podemos apreciar en el ejemplo de la Figura 3.3. Finalmente, se supone que se trabaja con un modelo determinista, es decir que todos sus parámetros se conocen con certeza.

Solución 1



Solución 2

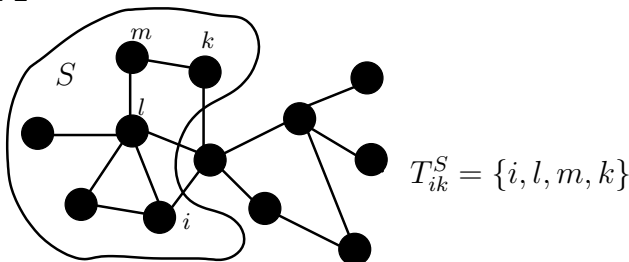


Figura 3.3: La distancia de red depende de la solución.

### 3.2 LIMITACIONES DEL MODELO EUCLIDEANO

Antes de abordar el problema con la métrica de distancias de red, se analizó lo que sucedería si se implementara un método de solución basado en distancia euclideana y simplemente cambiar la matriz de distancias por la matriz de rutas más cortas. Para esto, se tomó el método desarrollado en el trabajo de Caballero-Hernández et al. [6] donde se utiliza el  $p$ -centro como medida de dispersión.

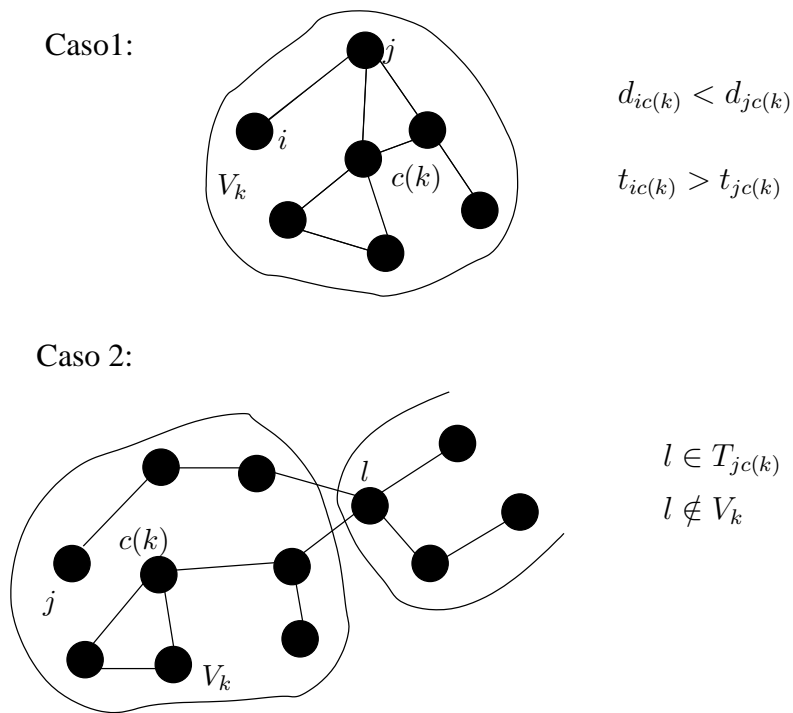


Figura 3.4: Casos que se presentan cuando se utiliza un método basado en distancia euclideana para resolver un enfoque basado en distancia de red.

Al finalizar la evaluación se encontró que ocurren dos casos no deseables que se muestran en la Figura 3.4 y además, son descritos a continuación:

1. Sea  $c(k)$  el centro del territorio  $k$ , sean  $i$  y  $j$  dos unidades básicas asignadas al territorio  $k$ . Sea  $t_{ij}$  la distancia de la ruta más corta entre  $i$  y  $j$ ;  $i, j \in V$  (el conjunto de todas las unidades). Puede presentarse que  $d_{ic(k)} < d_{jc(k)}$  y  $t_{ic(k)} > t_{jc(k)}$ .

No. de Unidades	Tolerancia	No. de Instancias	Violaciones( %)
500	0.05	30	10.3
500	0.10	30	9.8
1000	0.05	30	10.95
1000	0.10	30	10.24

Tabla 3.1: Resultados al resolver un enfoque basado en distancia euclideana usando las distancias de las rutas más cortas  $t_{ij}$  en lugar de las distancias euclideanas  $d_{ij}$ . Instancias de 500 y 1000 unidades fueron usadas.

2. Sea  $j \in V_k$  y  $l \in T_{jc(k)}$  (ruta más corta entre  $j$  y  $c(k)$ ) tal que  $l \notin V_k$ .

A estos dos casos les denominaremos violaciones. Estas violaciones fueron contabilizadas al término del algoritmo. Para este análisis se ejecutaron 30 réplicas de cuatro diferentes clases de instancias, dos de 1000 nodos y dos de 500, con tolerancias de 0.10 y 0.05, respectivamente.

En la Tabla 3.1 se observa que alrededor del 10% de los casos que se presentan dentro de un territorio son violaciones. Este porcentaje se obtuvo a través de la relación entre el número de rutas de cada nodo con su centro y el número de violaciones presentadas en la solución. De aquí concluimos que el método de solución basado en distancias euclideanas no es efectivo resolviendo el enfoque con una métrica de distancia de red. Es por esto que el desarrollo de un nuevo modelo es necesario para implementar la métrica de distancias de red.

### 3.3 FORMULACIÓN DEL PROBLEMA

En general, podemos representar nuestra red de distribución mediante un grafo plano  $G = (V, E)$  no dirigido, donde  $V$  es el conjunto de nodos y  $E$  el conjunto de aristas. De ahora en adelante una unidad básica representará un nodo de la red de distribución. Una arista,  $(i, j) \in E$ , entre dos manzanas representa la existencia

de una conexión directa entre ellas (Figura 3.5). El peso de cada arista  $(i, j) \in E$  está definido por la distancia euclidea  $d_{ij}$ . Como se dijo antes, cada nodo tiene asignado dos actividades,  $a = 1, 2$ , el número de clientes y la demanda total. La medida de la actividad  $a$  en el nodo  $i$  se representa mediante  $w_i^a$ . El número de territorios a construir está dado por el parámetro  $p$ . Sea  $V_k \subset V$  un territorio,  $k =$

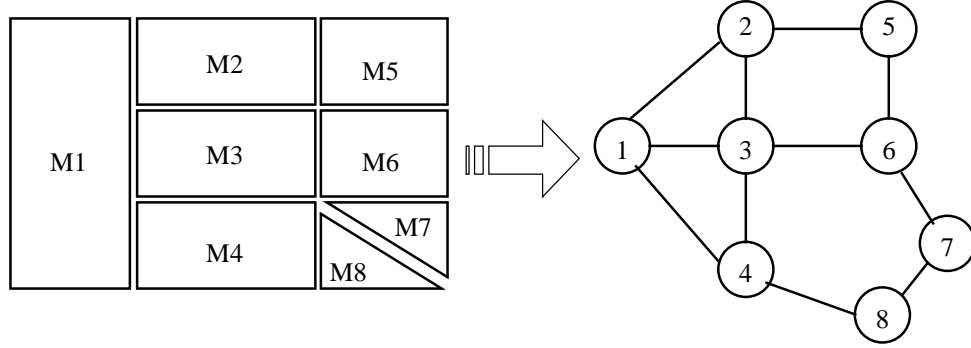


Figura 3.5: Representación gráfica de las manzanas de la ciudad.

$1, 2, 3, \dots, p$ , entonces  $w^a(V_k) = \sum_{i \in V_k} w_i^a$ ,  $a = 1, 2$  representa la medida del territorio  $V_k$  con respecto a la actividad  $a$ . Una meta deseable sería obtener territorios que tengan la medida perfecta con ambas actividades, la cual definimos para la actividad  $a$  como:  $\mu^a = w^a(V)/p$ . Pero la naturaleza discreta del problema hace que sea prácticamente imposible obtener territorios con esta medida para cualquiera de las dos actividades (Figura 3.6). Es por esto que introducimos un parámetro  $\tau^a$  como desviación a la medida perfecta de balanceo  $\mu^a$ . De esta manera podemos considerar que un territorio  $k$  está balanceado respecto a la actividad  $a$  si cumple con la siguiente condición:

$$w(V_k) = \sum_{i \in V_k} w_i^a \in [(1 - \tau^a)\mu^a, (1 + \tau^a)\mu^a], a = 1, 2.$$

Definamos  $T_{ij}^{V_k}$  como la ruta más corta del nodo  $i$  al nodo  $j$  en  $G = (V_k, A(V_k))$ ,  $V_k \subset V$ ,  $i, j \in V$  y denotemos a  $t_{ij}^{V_k}$  como su longitud. existen diferentes medidas de dispersión territorial. En este caso, se usa una medida de diámetro dada por:

$$f(V_1, V_2, \dots, V_p) = \max_{k=1, \dots, p} \left\{ \max_{i, j \in V_k} \left\{ t_{ij}^{V_k} \right\} \right\}.$$

Sea  $\Pi$  la colección de todas las posibles  $p$ -particiones de  $V$ . El modelo se muestra a continuación:

$$\min_{X \in \Pi} f(X) = \max_{k=1, \dots, p} \left\{ \max_{i, j \in X_k} \left\{ t_{ij}^{X_k} \right\} \right\} \quad (3.1)$$

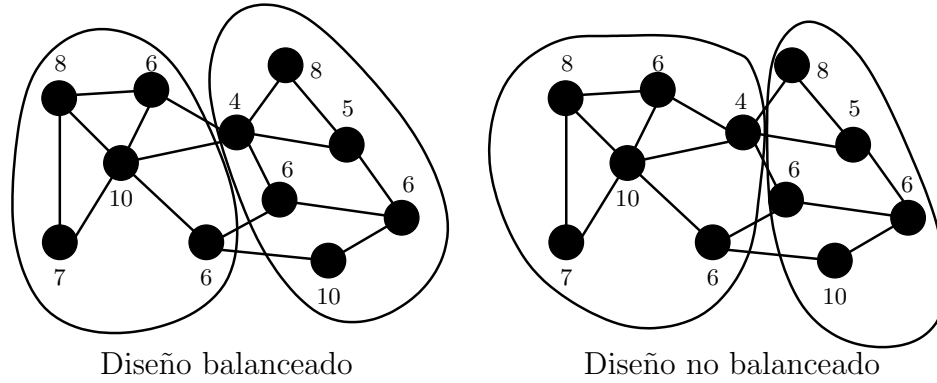
sujeto a:

$$X = (X_1, X_2, \dots, X_p), X \text{ es una } p\text{-partición} \quad (3.2)$$

$$w^a(X_k) \in [(1 - \tau)^a \mu^a, (1 + \tau)^a \mu^a], a = 1, 2; k = 1, \dots, p \quad (3.3)$$

$$G = (X_k, E(X_k)) \text{ es conexo } \forall k = 1, \dots, p \quad (3.4)$$

donde (3.1) es la función objetivo en la cual se requiere minimizar la dispersión basada en la medida del diámetro, (3.5) asegura una  $p$ -partición, (3.3) describe el balance de los territorios y (3.4) establece que cada uno de ellos debe ser conexo. Nótese que este modelo ignora el costo de ruteo.



$$w(V) = 76, p = 2, \mu = 76/2 = 38, \tau = 0.05$$

$$\text{Rango factible} = [36.1, 39.9]$$

Figura 3.6: Ilustración del concepto de balanceo territorial.

### 3.4 COSTOS DE RUTEO

En un problema de ruteo es de interés encontrar un conjunto de rutas para un determinado número de flotillas de vehículos partiendo cada uno desde un nodo

central de abasto, visitando a los clientes y regresando al nodo central. Cada una de estas rutas en cierta medida define también un territorio el cual consta de los nodos visitados. Sin embargo, este tipo de territorios, por lo general, no cumplen con el criterio de compacidad deseado por la empresa (ver Figura 3.7). Es debido a esto que necesitamos construir las rutas de manera diferente. Como hemos visto en nuestra revisión de la literatura, el ruteo puede formar parte de una segunda etapa de implementación justo después del diseño territorial para diseñar rutas que satisfagan la demanda. Aunque en la práctica la empresa le da una importancia mayor al criterio de compacidad que al costo de ruteo, y así ha sido reflejado en los trabajos desarrollados previamente, la idea central de esta tesis es la de proponer un modelo que incorpore una medida de costos de ruteo con una cierta penalización definida por el usuario la cual dependerá del grado de importancia deseado con respecto a la medida de dispersión. Este modelo puede además ayudar a discernir, entre una serie de diseños territoriales con medida de dispersión muy similar, aquéllos que sean más atractivos desde el punto de vista del ruteo del producto.

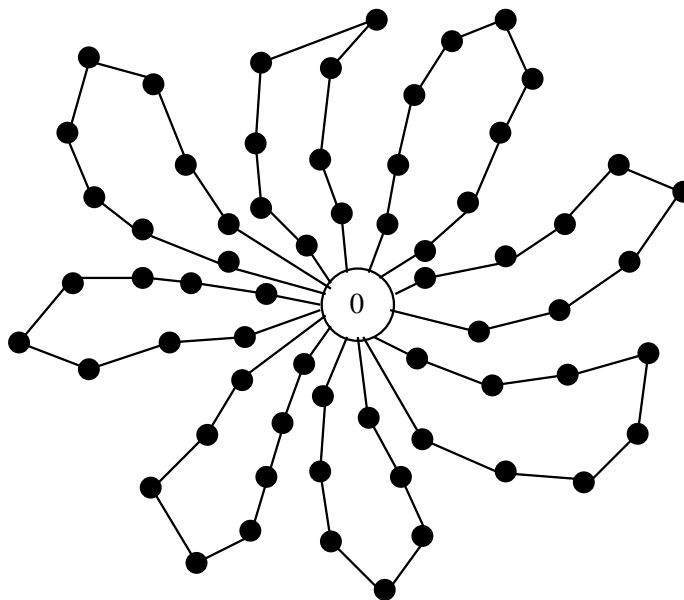


Figura 3.7: Ruteo de vehículos.

En particular cada territorio puede ser tomado como una red de distribución conexa, de esta manera podemos calcular el costo de visitar a cada unidad básica que



pertenecen al mismo territorio. Ahora, podemos definir otro objetivo, minimizar los costos de visitar a cada unidad básica en los territorios construidos. En este problema consideraremos solo los costos de visita internos en cada territorio, donde además, se tomará en cuenta los costos de llegada y de salida del territorio con respecto al centro de distribución (ver Figura 3.8). Comúnmente estaríamos hablando de resolver

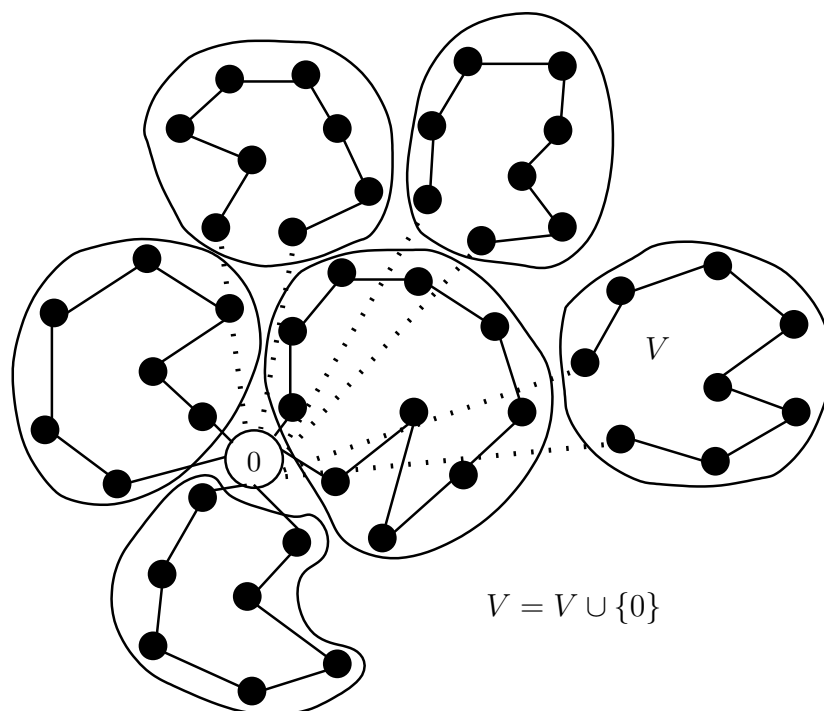


Figura 3.8: Costos de ruteo para cada territorio.

un problema del agente viajero (TSP por sus siglas en inglés) para cada territorio, pero dada la condición de que nuestra red de distribución es considerada como un grafo planar (que no es completo) no necesariamente podemos obtener un recorrido en cada territorio. Haciendo una pequeña transformación en las condiciones de un TSP, sustituir la matriz de distancias por la matriz de rutas más cortas, obtenemos lo que se conoce en la literatura como el Problema del Agente Viajero con Múltiples Visitas (TSPM por sus siglas en inglés). No es de nuestro interés obtener recorridos en cada territorio sino como ya vimos, evaluar el costo de recorrer todo el territorio aunque se visiten algunos nodos en más de una ocasión. De aquí, podemos establecer

nuestra nueva función objetivo:

$$\min_{X \in \Pi} f(X) = \max_{k=1, \dots, p} \left\{ \max_{ij \in X_k} \left\{ t_{ij}^{X_k} \right\} \right\} + \sum_{k=1, \dots, p} TSPM(X_k), \quad (3.5)$$

donde  $TSPM(X_k)$  representa la longitud óptima del TSPM en el subgrafo  $X_k \cup \{0\}$ ,  $E(X_k \cup \{0\})$  con 0 representando al centro de distribución. Este costo está dado por  $TSPM(X_k) = \min_{\pi \in \Theta(X_k)} \sum_{\pi(i) \in X_k \cup \{0\}} t_{\pi(i)\pi(i+1)}^{X_k}$ , donde  $\pi(i)$  representa al nodo de la posición  $i$  en la permutación  $\pi$  y  $\Theta(X_k)$  es el conjunto de todas las posibles permutaciones de  $X_k \cup \{0\}$ .

### 3.5 COMPLEJIDAD COMPUTACIONAL

La *complejidad computacional* se refiere al estudio de los recursos que requiere el cómputo de un algoritmo para resolver un problema. Generalmente, se estudian dos recursos: el *tiempo* y el *espacio*. El tiempo se relaciona con el número y tipo de pasos de ejecución del algoritmo, mientras que el espacio se refiere a la cantidad de memoria utilizada. Algunos de los conceptos que utilizaremos en esta sección se encuentran descritos con mayor detalle en el texto de Papadimitriou y Steiglitz [40].

Un *problema de decisión* es una pregunta que se puede responder con *si* o *no*. Problemas de complejidad similar suelen pertenecer a la misma clase. Entre éstas las más comunes son: la clase  $P$  y la clase  $NP$ . La clase  $P$  consta de todos los problemas de decisión que pueden ser resueltos mediante un algoritmo determinista de tiempo polinomial, en cambio, la clase  $NP$  consta de los problemas de decisión que solo pueden ser resueltos mediante un algoritmo no determinista en tiempo polinomial, donde un algoritmo *determinista* es aquel en el que existe solo una alternativa para encontrar la solución, mientras que en un algoritmo *no determinista* existen varias alternativas.

A grandes rasgos, un problema es  $NP$ -completo si pertenece a la clase  $NP$  y además, todos los demás problemas de la clase  $NP$  pueden ser transformados polinomialmente al problema en cuestión. Cuando todos los problemas de la clase

$NP$  pueden ser transformados a un problema en cuestión y no se puede argumentar que este mismo problema pertenezca a la clase  $NP$ , se dice entonces que el problema es  $NP$ -duro, ya que es al menos tan difícil como todos los problemas de la clase  $NP$ .

Para demostrar que un problema es  $NP$ -duro no es necesario transformar todos los problemas de la clase  $NP$  al problema en cuestión, sino que se puede utilizar el método que se conoce como *reducción*. Este método consiste en tomar una instancia del problema  $A$  (que se conoce es  $NP$ -duro) y transformarla en una instancia del problema  $B$  que pueda ser resuelta por un algoritmo que resuelva las instancias del problema  $B$ .

El problema abordado en esta tesis es un problema de Diseño de Territorios Comerciales con restricciones de conectividad, distancias de red y costos de ruteo. Para simplificar la notación lo denominaremos TDP-CR.

Una manera de demostrar que nuestro problema es  $NP$ -duro es reducir polinomialmente el Problema del Agente Viajero (TSP) al TDP-CR. Para esto, tomaremos una instancia particular del TDP-CR y mostraremos que ésta corresponde a un problema  $NP$ -duro. En nuestro caso, consideremos el caso especial de TDP-CR donde  $p = 1$ ,  $\tau^a = M$ , donde  $M$  es un número muy grande. Con esto desaparecen las restricciones de balanceo (3.3), de conectividad (3.4) y de número de territorios y de asignación (3.5). El problema resultante es simplemente encontrar un tour de mínimo recorrido que visite a todos los nodos en  $V \cup \{0\}$ . Este caso especial del TDP-CR corresponde precisamente al conocido Problema del Agente Viajero (en  $n + 1$  vértices), el cual es conocido que es  $NP$ -duro [26]. Con esto concluimos que TDP-CR es al menos tan difícil como el TSP y por ende TDP-CR es  $NP$ -duro.

## CAPÍTULO 4

# MÉTODO DE SOLUCIÓN

---

Como observamos, nuestro problema pertenece a la clase de los NP-duro. Dado el tamaño de las instancias que se pretenden resolver y el hecho de que se tiene conocimiento que aún las versiones más simplificadas del problemas (por ejemplo, dispersión basada en distancia euclideana sin costos de ruteo) tienen muchas dificultades en resolverse en forma exacta como programas enteros mixtos lineales, más allá de los 100 nodos se propone una metaheurística como método de solución. Las metaheurísticas son métodos de solución que controlan una interacción entre procedimientos de mejora local y estrategias de mayor nivel para crear un proceso capaz de escapar de un óptimo local y realizar una búsqueda robusta del espacio de soluciones [23]. Un procedimiento metaheurístico muy conocido es el Procedimiento de Búsqueda Adaptativo, Voraz y Aleatorizado (GRASP, por sus siglas en inglés), desarrollado por Feo y Resende [17], el cual es un procedimiento multiarranque en el que cada arranque corresponde a una iteración. En cada una de estas iteraciones encontramos dos fases: la fase de construcción, que se encarga de obtener una solución factible de alta calidad; y la fase de mejora o postprocesamiento, que busca mejorar la solución obtenida en la primera. Se ha encontrado que el método GRASP es muy efectivo al momento de ser implementado en problemas combinatorios. En nuestro caso, el problema de considerar la conectividad en la solución puede resolverse construyendo soluciones que ya lo sean, y de esta manera, mantenerla durante todo el proceso. La estructura del GRASP nos permite establecer este enfoque.

Otro punto importante es que la fase de construcción está basada en una fun-

ción voraz que solamente considera la dispersión, además de un término relacionado con las infactibilidades de las restricciones de balanceo (3.3). Esto se debe a que la empresa se encuentra muy interesada en obtener territorios compactos. Es a partir de aquí, que se pretende establecer la evaluación de los costos de ruteo que irán mejorando durante la fase de postprocesamiento para que de esta manera no se incumpla el criterio de la compacidad. Al finalizar la fase de construcción es posible que el número de territorios contruidos sea mayor o menor a  $p$ . Es por esto que introducimos una fase de ajuste antes de comenzar con la fase dos del algoritmo. Por otra parte, la fase de mejora consiste en una búsqueda local en donde la función de mérito incluye los mismos dos términos de la función voraz y adicionalmente un tercer término que evalúa los costos de ruteo. En las secciones de este capítulo, describiremos con detalle cada uno de los componentes del algoritmo GRASP propuesto.

## 4.1 FASE DE CONSTRUCCIÓN

La fase de construcción de nuestro algoritmo GRASP (mostrado en la Figura 4.1) está basada en el trabajo de Ríos-Mercado y Fernández [44] en el cual, la función voraz se encuentra formada de dos términos: el primero que corresponde a la medida de compacidad y el segundo que indica la suma de infactibilidades con respecto a las restricciones de balance de las actividades. El método considera, a partir de una iteración dada, un territorio parcial e intenta ya sea asignarle un nodo que no ha sido considerado ó cerrar el territorio actual y comenzar uno nuevo. En este procedimiento de construcción sustuiremos la matriz de distancias euclideanas,  $d_{ij}$ , por la matriz de rutas más cortas,  $t_{ij}$ , del grafo original. El motivo principal es obtener los territorios contruidos en el menor tiempo posible para compensar los recálculos de rutas más cortas en la fase de postprocesamiento. Lo mismo es considerado para los costos de ruteo, los cuales son evaluados al final de la fase (Paso 7).

---

*GRASP* ( $NumIt, \alpha$ )
 

---

**Input:**  $NumIt$ : Número de iteraciones $\alpha$ : Parámetro de calidad de *RCL***Output:**  $S^*$ : Mejor solución

```

1:  $S^* \leftarrow \emptyset$ ;
2: for ( $i = 1, \dots, NumIt$ ) do
3:    $(S_1, S_2, \dots, S_q) \leftarrow Construcción(\alpha)$ ;
4:   if ( $q \neq p$ ) then
5:      $(S_1, S_2, \dots, S_p) \leftarrow Ajuste(S, q)$ ;
6:   end if
7:   EvaluaciónRuteoExacto ( $S$ );
8:    $S \leftarrow Postprocesamiento(S)$ ;
9:   if ( $S$  mejor que  $S^*$ ) then
10:     $S^* \leftarrow S$ ;
11:  end if
12: end for
13: return  $S^*$ ;

```

---

Figura 4.1: Algoritmo GRASP.

En el Paso 3 se lleva a cabo la fase constructiva la cual se describe a continuación. Formalmente, sea  $V_k$  el territorio que actualmente se está construyendo, su medida de compacidad se encuentra definida por  $f(V_k) = \max_{i,j \in V_k} \{t_{ij}\}$ , y  $w(V_k) = \sum_{i \in V_k} w_i^a$  representa la medida del territorio,  $a = 1, 2$ . Decimos que si un nodo aún no ha sido asignado a algún territorio, es un nodo candidato. Entonces, sea  $v \in V$  un nodo candidato, definimos su función voraz como:

$$\phi(v) = \lambda F_k(v) + (1 - \lambda)G_k(v), \quad (4.1)$$

donde

$$F_k(v) = f(V_k \cup \{v\}) = \max \left\{ f(V_k), \max_{j \in V_k} \{t_{vj}\} \right\}$$

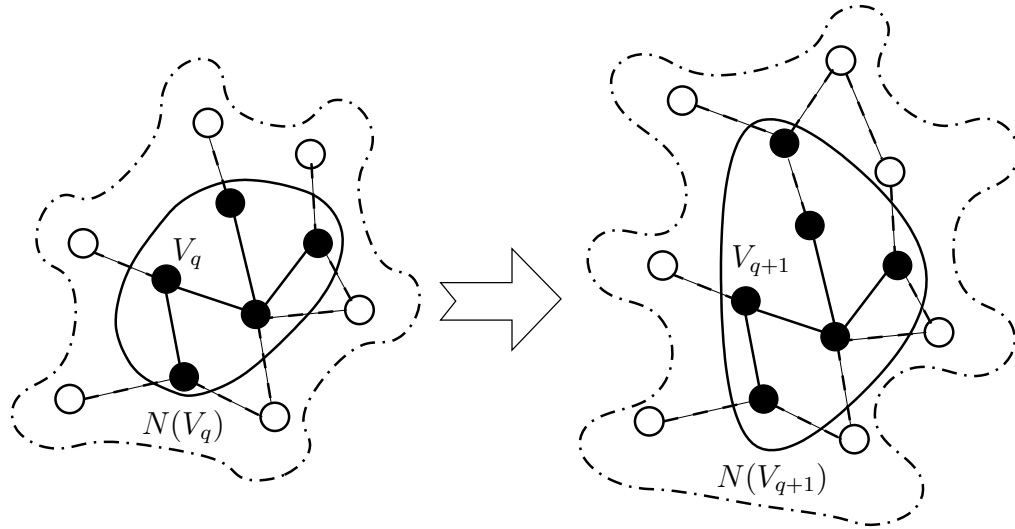


Figura 4.2: Inserción de un nodo al territorio  $V_q$  actualmente abierto.

es el término de la dispersión, y

$$G_k(v) = \sum_{a=1,2} g_k^a(v),$$

con  $g_k^a(v) = (1/\mu^a) \max \{w^a(V_k \cup \{v\}) - (1 + \tau^a)\mu^a, 0\}$ , es la suma de las infactibilidades con respecto a las restricciones de balance. Cabe señalar que  $g_k^a(v)$  representa las infactibilidades con respecto a la cota superior de las restricciones de balance de la actividad  $a$ . Finalmente,  $0 \leq \lambda \leq 1$ .

Un componente importante de un GRASP es la lista restringida de candidatos (RCL por sus siglas en inglés), de esta manera solo los candidatos más prometedores a mejorar la solución son escogidos. Hay dos maneras de construirla: por cardinalidad y por umbral de calidad. La lista por umbral de calidad es utilizada más comúnmente en este tipo de procedimientos y nuestro caso no es la excepción. Para aceptar a un nodo como potencial candidato debe estar dentro de este umbral que se define de la siguiente manera: sea  $V_q$  nuestro territorio actual, y  $N(V_q)$  representa el conjunto de nodos vecinos candidatos del territorio  $V_q$ . Definimos a  $\Phi_{\min} := \min_{v \in N(V_q)} \{\phi(v)\}$  y a  $\Phi_{\max} := \max_{v \in N(V_q)} \{\phi(v)\}$  de manera que  $RCL := \{j \in N(V_q) : \phi(j) \in [\Phi_{\min}, \Phi_{\min} + \alpha(\Phi_{\max} - \Phi_{\min})]\}$ . El parámetro  $\alpha$  es un parámetro de calidad que nos permite

establecer un umbral donde podemos encontrar soluciones de mayor calidad. Este parámetro es definido por el usuario, comúnmente por experimentación.

---

*Construcción* ( $\alpha$ )

---

**Input:**  $\alpha$ : Parámetro de calidad

**Output:**  $S = \{V_1, V_2, \dots, V_q\}$ : Diseño territorial

```

1:  $q \leftarrow 1, \bar{V} \leftarrow V;$ 
2:  $v \leftarrow \arg \min\{|N^i| : i \in \bar{V}\};$ 
3:  $V_q \leftarrow \{v\};$ 
4:  $\bar{V} \leftarrow \bar{V} \setminus \{v\};$ 
5: while ( $\bar{V} \neq \emptyset$ ) do
6:    $N(V_q) \leftarrow$  conjunto de vecinos de  $V_q;$ 
7:    $\Phi_{\min} \leftarrow \min\{\phi(v) : v \in N(V_q)\}, \Phi_{\max} \leftarrow \max\{\phi(v) : v \in N(V_q)\};$ 
8:    $\text{RCL} \leftarrow \{j \in N(V_q) : \phi(j) \in [\Phi_{\min}, \Phi_{\min} + \alpha(\Phi_{\max} - \Phi_{\min})]\};$ 
9:   Escoger  $v \in \text{RCL}$  aleatoriamente;
10:   $V_q \leftarrow V_q \cup \{v\}, \bar{V} \leftarrow \bar{V} \setminus \{v\};$ 
11:  if ( $\exists \in \{1, 2\}$  tal que  $w^a(V_q) > (1 + \tau^a)\mu^a$ ) then
12:     $q \leftarrow q + 1;$ 
13:     $v \leftarrow \arg \min\{|N^i| : i \in \bar{V}\};$ 
14:     $V_q \leftarrow \{v\};$ 
15:     $\bar{V} \leftarrow \bar{V} \setminus \{v\}$ 
16:  end if
17: end while
18: return  $S = \{V_1, V_2, \dots, V_q\}$ 

```

---

Figura 4.3: Fase de Construcción.

Luego, escogemos un nodo  $v \in \text{RCL}$  aleatoriamente para insertarlo en nuestro territorio actualmente abierto,  $V_q = V_q \cup \{v\}$  como se muestra en la Figura 4.2. Es en este punto cuando se comprueba si un territorio ha violado las restricciones de balanceo con respecto a la cota superior, si  $w^a(V_q) > \rho(1 + \tau^a)\mu^a$ , para al menos un  $a = 1, 2$ , y donde  $\rho > 0$  es un parámetro de ajuste que da mayor flexibilidad



a etapas posteriores, entonces se decide cerrar el territorio y abrir un nuevo. Este procedimiento continúa realizándose hasta que  $q$  territorios son construidos. El nodo inicial se toma aleatoriamente entre aquellos de menor grado (ver Paso 2 en el algoritmo de la Figura 4.3).

## 4.2 FASE DE AJUSTE

El procedimiento de construcción no necesariamente termina con una solución factible. Esto se debe a que el número final de territorios  $q$  puede no ser  $p$ , además de no tener territorios totalmente balanceados con respecto a ambas actividades. Esta fase intenta recuperar la factibilidad de la solución obtenida, con respecto al tamaño de la partición cuando  $q \neq p$ . Se tienen dos situaciones diferentes: si  $q > p$ , se combinan territorios y si  $q < p$ , se dividen territorios. Para realizar la operación de combinar territorios, tomamos el territorio de menor tamaño y lo unimos al territorio vecino de menor tamaño, donde definimos  $M(V_i)$  como el conjunto de índices de los territorios vecinos a  $V_i$ , esto se realiza iterativamente hasta que  $q = p$ . En contraste con la operación de combinar territorios, dividir un territorio en dos territorios resulta por sí mismo un TDP con  $p = 2$ . Esta operación toma el territorio de mayor tamaño y lo divide en dos territorios conexos aplicando el mismo algoritmo con  $p = 2$ , y de igual manera, se realiza hasta que  $q = p$ . En la Figura 4.4 se observa el pseudo código de nuestra fase de ajuste.

Ríos-Mercado y Fernández [44] obtienen resultados empíricos para los cuales se observa que la operación de dividir territorios se presenta sólo en el 0.4% de los casos. En la Figura 4.4 podemos observar con mayor detalle el funcionamiento de esta fase.

---

*Ajuste* ( $S$ )
 

---

**Input:**  $S = \{V_1, V_2, \dots, V_q\}$ : Diseño territorial ( $q$ -partición)

**Output:**  $S = \{V_1, V_2, \dots, V_p\}$ : Diseño territorial ( $p$ -partición)

```

1: while ( $q > p$ ) do
2:    $k_1 \leftarrow \arg \min \sum_{a=1}^2 \frac{w^a(V_l)}{w^a(V)}$ ;  $l = 1, \dots, p$ ;
3:    $k_2 \leftarrow \arg \min \sum_{a=1}^2 \frac{w^a(V_l)}{w^a(V)}$ ;  $l \in M(V_{k_1})$ ;
4:    $V_{k_1} \leftarrow V_{k_1} \cup V_{k_2}$ ;
5:   Eliminar  $V_{k_2}$  de  $S$ ;
6:    $q \leftarrow q - 1$ ;
7: end while
8: while ( $q < p$ ) do
9:    $k = \arg \max \sum_{a=1}^2 \frac{w^a(V_l)}{w^a(V)}$ ;  $l = 1, \dots, p$ ;
10:   $(V_1^{\text{new}}, V_2^{\text{new}}) \leftarrow \text{ResolverTDP}(V_k, E(V_k), 2)$ ;
11:  Reemplazar  $V_k$  por  $(V_1^{\text{new}}, V_2^{\text{new}})$  en  $S$ ;
12:   $q \leftarrow q + 1$ ;
13: end while
14: return  $S = \{V_1, V_2, \dots, V_p\}$ ;

```

---

Figura 4.4: Fase de Ajuste.

### 4.3 EVALUACIÓN DE COSTOS DE RUTEO

Hay diferentes etapas donde se evalúan los costos de ruteo. por ejemplo, al final de la fase constructiva o cuando se tiene un diseño particular ya dado. En este caso, la evaluación del costo de ruteo con territorios ya fijos, se lleva a cabo mediante la solución de  $p$  problemas de agente viajero, uno por cada territorio. En este caso se emplea un método exacto basado en un esquema de ramificación y corte. Una descripción más detallada del método se encuentra en el Apéndice A.

En nuestro caso, se empleó el paquete CONCORDE, desarrollado por Applegate et al. [2]. En la Figura 4.5 se ilustra el pseudo-código del procedimiento de

evaluación, donde  $SolTSPExacto(H)$  resuelve un TSP optimamente para el conjunto de nodos  $H$  y además, 0 representa el centro de distribución.

---

*EvaluaciónRuteoExacto* ( $S$ )

---

**Input:**  $S = \{V_1, V_2, \dots, V_p\}$ : Diseño territorial

**Output:**  $c$ : Costo de ruteo

```

1:  $c \leftarrow 0$ ;
2: for ( $k = 1, \dots, p$ ) do
3:    $c \leftarrow c + SolTSPExacto(V_k \cup \{0\})$ ;
4: end for
5: return  $c$ ;
```

---

Figura 4.5: Fase de Evaluación exacta de costos de ruteo.

## 4.4 BÚSQUEDA LOCAL

El vecindario de nuestra búsqueda local  $N(S)$  se encuentra dado para todas las posibles soluciones que se puedan obtener mediante el movimiento de un nodo (o unidad básica) a otro territorio. Debido a que debemos mantener la conexidad durante todo momento, no todos los nodos pueden ser movidos y es por esto que solamente consideramos los nodos que son adyacentes a algún otro nodo en cualquier otro territorio. La Figura 4.6 explica el movimiento. Sea  $i$  un nodo cualquiera,  $m(i)$  representa el territorio actual del nodo  $i$ , si existe  $(i, j) \in E$  tal que  $m(i) \neq m(j)$ , entonces podemos intercambiar el nodo  $i$  al territorio  $m(j)$  si además, la conexidad de ambos territorios se mantiene después del movimiento.

La búsqueda local se mantiene realizando este tipo de movimientos y en cada uno se realiza un recálculo de la ruta más corta en los territorios afectados, además de recalcular los costos de ruteo en los mismos. Cabe señalar, que el recálculo de los territorios se realiza con dos heurísticas diferentes. En el territorio  $m(i)$  (de donde se extrajo el nodo  $i$ ) simplemente se reconecta el pseudo-recorrido. En el territorio  $m(j)$ , que es donde se introdujo  $i$ , se inserta el nodo  $i$  de manera que se minimice

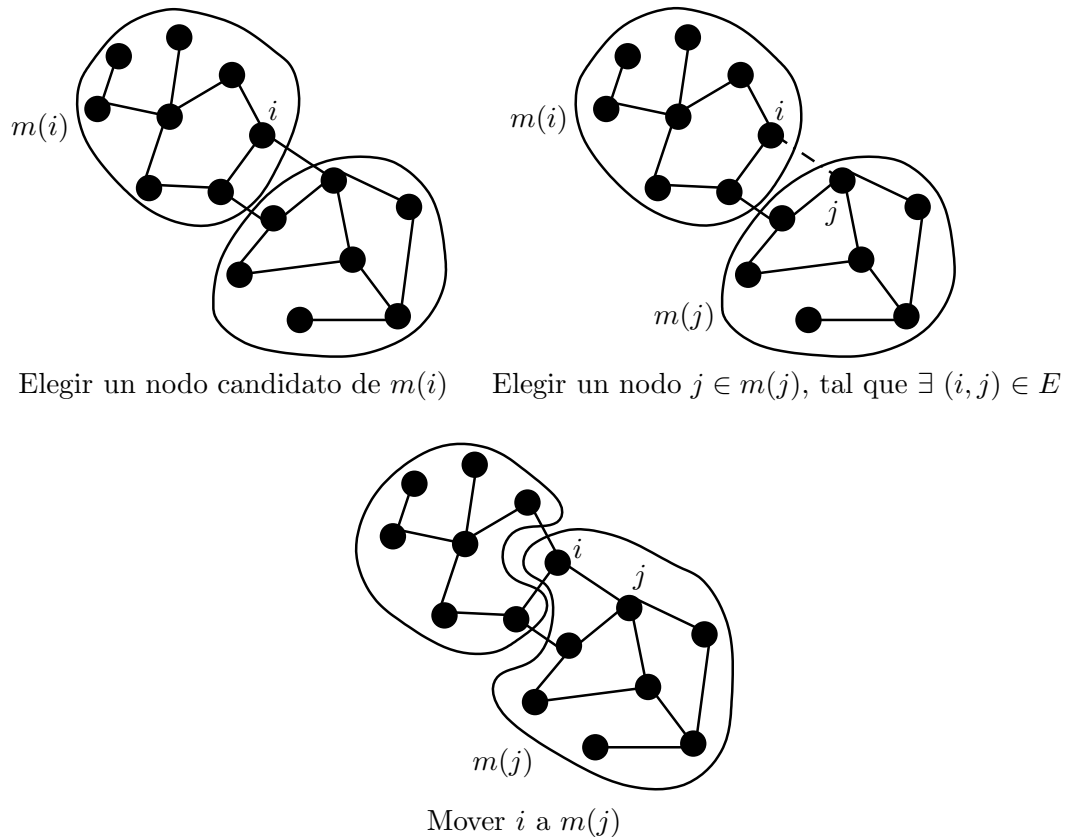


Figura 4.6: Movimiento de la búsqueda local.

el costo del pseudo-recorrido. Ambas pueden observarse en la Figura 4.7. En esta parte se introdujo un umbral en el que cada  $l$  iteraciones se emplea el método de ramificación y corte para la solución exacta de los costos de ruteo, y de esta manera permanecer cerca de la solución óptima con respecto a los costos de ruteo. En la Figura 4.8 podemos ver el pseudo código de nuestra búsqueda local.

La primera fase no garantiza factibilidad en las soluciones construidas. Nuestra búsqueda local pretende mejorar la solución de la primera fase, además de recuperar la factibilidad de la misma en caso que sea necesario. Es por esto que definimos la siguiente función de mérito: Sea  $S = \{V_1, V_2, \dots, V_p\}$  una partición de  $V$ , entonces

$$\psi(S) = F(S) + \sigma G(S) + \gamma H(S) \quad (4.2)$$

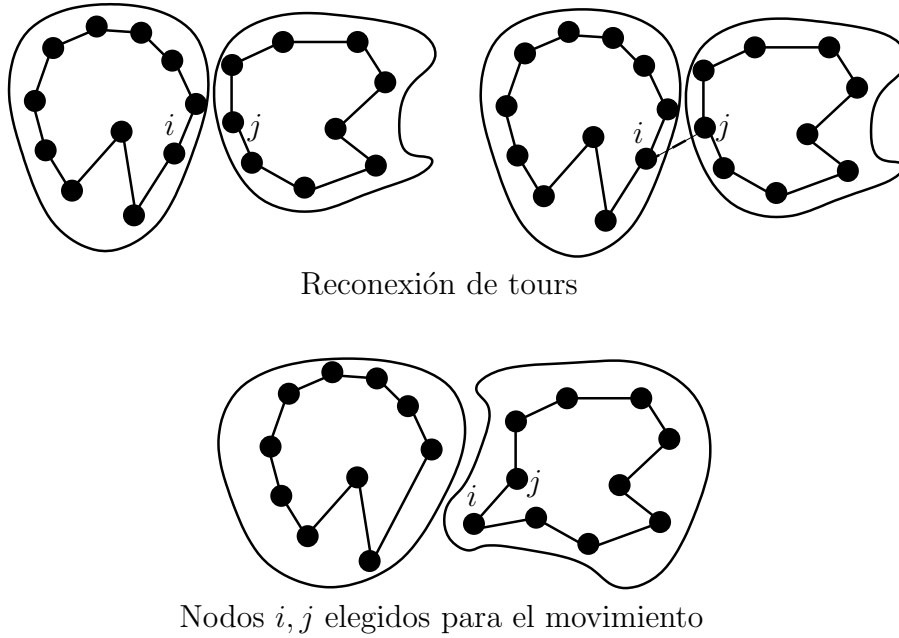


Figura 4.7: Movimientos heurísticos para la reconstrucción de recorridos.

donde,

$$F(S) = \max_{k=1, \dots, p} \left\{ \max_{i, j \in V_k} \left\{ t_{ij}^{V_k} \right\} \right\},$$

$$G(S) = \sum_{k=1, \dots, p} TSP(V_k),$$

y

$$H(S) = \sum_{k=1, \dots, p} \sum_{a=1, 2} g^a(V_k),$$

con  $TSP(V_k) = \sum_{n(i)=1}^{|V_k|+1} t_{n(i)n(i+1)}^{V_k}$ , donde  $n(i)$  representa al nodo de la posición  $i$  en la permutación  $n \in \Theta(V_k)$  que es el conjunto de todas las posibles permutaciones de  $V_k$ , y  $g^a(V_k) = (1/\mu^a) \max \{w^a(V_k) - (1 + \tau^a)\mu^a, (1 - \tau^a)\mu^a - w^a(V_k), 0\}$ , como la suma de las infactibilidades con respecto a las restricciones de balance. Los parámetros  $\sigma$  y  $\gamma$  son autoajustables. El parámetro  $\sigma$  aumenta el doble si la solución actual empeora y disminuye a la mitad si la solución mejora. Para el parámetro  $\gamma$ , se considera que si un número  $l$  de iteraciones consecutivas la solución es infactible,  $\gamma$  se incrementa al doble, por el contrario, si número  $l$  de iteraciones consecutivas la solución es factible  $\gamma$  disminuye a la mitad. Podemos destacar que  $t_{ij}^{V_k}$  representa el

---

*Postprocesamiento* ( $S$ )

---

**Input:**  $S = \{V_1, V_2, \dots, V_p\}$ : Diseño territorial**Output:**  $S^{\text{best}}$ 

```

1:  $S^{\text{best}} \leftarrow S$ ;
2:  $move \leftarrow \text{YES}$ ;
3: while ( $move = \text{YES}$ ) do
4:    $move \leftarrow \text{NO}$ ;
5:    $\bar{N} \leftarrow N(S)$ ;
6:   while ( $\bar{N} \neq \emptyset$  and  $move = \text{NO}$ ) do
7:      $\bar{S} \leftarrow \text{Random}(\bar{N})$ ;
8:      $\bar{N} \leftarrow \bar{N} \setminus \bar{S}$ ;
9:     if ( $\phi(S) < \phi(S^{\text{best}})$ ) then
10:       $S^{\text{best}} \leftarrow \bar{S}$ ;
11:       $move \leftarrow \text{YES}$ ;
12:     end if
13:   end while
14: end while
15: return  $S^{\text{best}}$ ;
```

---

Figura 4.8: Fase de Postprocesamiento: Búsqueda Local.

costo de la ruta más corta entre  $i$  y  $j$  dentro del territorio  $V_k$  y no necesariamente  $t_{ij}^{V_k} = t_{ij}$ ,  $i, j \in V$ . Así mismo,  $TSP(V_k)$  representa los costos de ruteo del territorio  $k$  que simbólicamente hacen referencia a un “recorrido”, que como mencionamos anteriormente, utiliza las rutas más cortas entre cada par de nodos, en lugar del peso de las aristas entre los mismos, para resolver un TSP.

# EVALUACIÓN COMPUTACIONAL

---

La evaluación computacional es una de las etapas más importantes en la construcción de un método de solución. En esta etapa es donde se calibran los parámetros del algoritmo construido y donde se evalúa la eficiencia del mismo. Es esencial establecer un ambiente de experimentación adecuado, además de definir cuales son los objetivos para poder analizar los resultados adecuadamente y establecer una conclusión tajante conforme al funcionamiento y calidad del algoritmo de solución.

## 5.1 OBJETIVOS

Los objetivos de nuestra evaluación computacional se establecen para determinar el funcionamiento idóneo del algoritmo, así como medir la calidad y eficiencia del mismo. Dichos objetivos se enlistan a continuación.

- Ajustar los parámetros de nuestro algoritmo para establecer su mejor funcionamiento.
- Implementar nuestro algoritmo en diversas instancias y variar los valores de los parámetros que son considerados fijos para evaluar el desempeño y la robustez del mismo.
- Mostrar la valía del método como herramienta de apoyo a la toma de decisiones.

## 5.2 ESCENARIOS Y AMBIENTE DE EXPERIMENTACIÓN

Las instancias de prueba de este trabajo fueron generados en el trabajo de Caballero-Hernández et al. [6] y son llamados escenarios DU (debido a que se encuentran basados en la distribución uniforme) que como ellos describen, sirven para evaluar el problema de una forma más general. Las instancias fueron generadas a partir de datos reales de la empresa distribuidora de bebidas. Cada instancia se compone de los nodos, las aristas de la red de distribución, el número de territorios requeridos y la desviación para cada medida de balanceo. Para todos nuestros experimentos empleamos instancias de 1000 nodos y una desviación de balance  $\tau^a$  de 0.05,  $a = 1, 2$ . El número de instancias que se usa en cada experimento es de 15.

El ambiente de experimentación simplemente se refiere al ambiente computacional en el que un algoritmo es implementado. Nuestro algoritmo fue codificado en el lenguaje de programación C++ y ejecutado con el compilador de GNU g++ versión 4.3.2 bajo la distribución Ubuntu 9.04 de Linux en un equipo con Procesador Intel(R) Core(TM)2 Quad CPU Q6600 de 2.40 GHz.

Algunos factores del algoritmo no serán abordados en este trabajo, pero debido a que serán utilizados en todos nuestros experimentados, se convierten en parte del ambiente de experimentación tal es el caso del número de iteraciones del algoritmo GRASP y del número de iteraciones elegido para el umbral de la implementación del algoritmo de ramificación y corte en la búsqueda local.

- El número de iteraciones de GRASP se ha fijado en 500, dado que en trabajos anteriores se ha observado que las soluciones de mayor calidad suelen encontrarse dentro de este intervalo. Además, nos enfrentemos con la problemática del costo de tiempo de cómputo.
- En la búsqueda local es necesario definir un número de iteraciones para establecer un umbral en el que se decide si se implemente el método de ramificación y corte o se emplea una heurística para evaluar los costos de ruteo durante



el intercambio nodal. Se ha observado un promedio de 200 iteraciones para al búsqueda local y es por esto que este número de iteraciones  $l$ , se ha fijado en 50, debido al gran costo computacional que se requiere.

## 5.3 AJUSTE PARAMÉTRICO

Como hemos señalado anteriormente, nuestro algoritmo consta de varios parámetros que necesitan ajustarse. Tal es el caso del parámetro  $\alpha$  de calidad de la RCL del GRASP y  $\lambda$  que es el parámetro de la función voraz en la fase de construcción. El ajuste paramétrico se realizó para instancias con un valor objetivo de  $p = 40$  territorios.

### 5.3.1 EXPERIMENTO A: PARÁMETRO $\alpha$

Este experimento tiene el objetivo de determinar el valor de  $\alpha$ , estudiando su efecto sobre el algoritmo. Para esto, se fijó el valor del parámetro  $\lambda = 0.6$  debido a la importancia que tiene el objetivo de compacidad. La búsqueda local se ejecuta hasta que se encuentra un óptimo local o, en el peor caso, hasta un número de iteraciones muy grande que en esta caso se ha fijado en 5000.

Para definir el valor del parámetro  $\alpha$  decidimos tomar el valor de la mejor solución encontrada para todos los valores de  $\alpha$  y después obtener la desviación relativa con respecto a cada valor de  $\alpha$ ,  $100 \times \frac{(f_\alpha - f_{mejor})}{f_{mejor}}$ . Debemos resaltar que la función objetivo consta de dos componentes: un costo por dispersión y otro por ruteo.

Recordemos también que el método arroja tres soluciones:  $S^{obj}$  mejor solución con respecto a la función objetivo original;  $S^{disp}$  mejor solución con respecto a la función de dispersión; y  $S^{ruteo}$  mejor solución con respecto a la función de costos de ruteo.

En la Tabla 5.1 se muestran los resultados que se obtuvieron después de emplear el algoritmo sobre las 15 instancias. Se muestra el promedio y la desviación estándar de la desviación relativa para cada  $\alpha$ .

		$\alpha = 0.0$	$\alpha = 0.1$	$\alpha = 0.2$	$\alpha = 0.3$	$\alpha = 0.4$
$s_{obj}$	Promedio	3.808	3.443	2.202	3.806	8.479
	Desviación	3.501	3.095	1.999	3.427	3.751
$s_{disp}$	Promedio	17.139	13.945	5.218	10.562	20.242
	Desviación	12.752	12.063	9.010	9.096	13.988
$s_{ruteo}$	Promedio	2.254	2.565	3.027	3.762	7.368
	Desviación	2.485	2.814	2.592	3.166	2.751
Soluciones Factibles		10	15	15	15	15

Tabla 5.1: Desviación relativa promedio con respecto a la mejor solución para cada valor de  $\alpha$ .

Para establecer el mejor valor de  $\alpha$  observamos la desviación relativa promedio con respecto al valor objetivo del problema. El menor promedio se consiguió con  $\alpha = 0.2$ , al igual que la menor desviación estándar. Aún más, el menor promedio de desviación relativa con respecto al término de dispersión también fue para  $\alpha = 0.2$ .

Observamos también que para todos los valores de  $\alpha$  se encontraron soluciones factibles, excepto para  $\alpha = 0$  el cual corresponde al costo totalmente voraz. A pesar que el menor promedio de desviación relativa con respecto al costo de ruteo no fue para  $\alpha = 0.2$ , observamos una gran cercanía de este valor con el mejor. Por lo tanto, se decidió establecer el valor de  $\alpha = 0.2$  como un compromiso razonable entre dispersión y ruteo.

### 5.3.2 EXPERIMENTO B: PARÁMETRO $\lambda$

El parámetro  $\lambda$  es el parámetro de la función voraz en la cual sólo encontramos dos términos: la dispersión y un término que representa la suma de las infactibilidades

con respecto a la cota superior de las restricciones de balanceo (ver sección 4.1). En la etapa de construcción es muy difícil encontrar soluciones factibles, es por esto que ambos términos son de gran importancia. Para determinar el mejor valor para el parámetro  $\lambda$  se efectuarán ejecuciones del método aplicando únicamente la fase constructiva para valores de  $\lambda \in \{0.0, 0.2, 0.4, 0.6, 0.8, 1.0\}$  a 15 instancias del problema. Se mide para cada una su valor de dispersión alcanzado y su infactibilidad relativa. En la Figura 5.1 se muestra el promedio de infactibilidades relativas para los diferentes valores de  $\lambda$ . En la Figura 5.2 se muestra el promedio de las desviaciones relativas con respecto al mejor valor de dispersión para diferentes valores de  $\lambda$ .

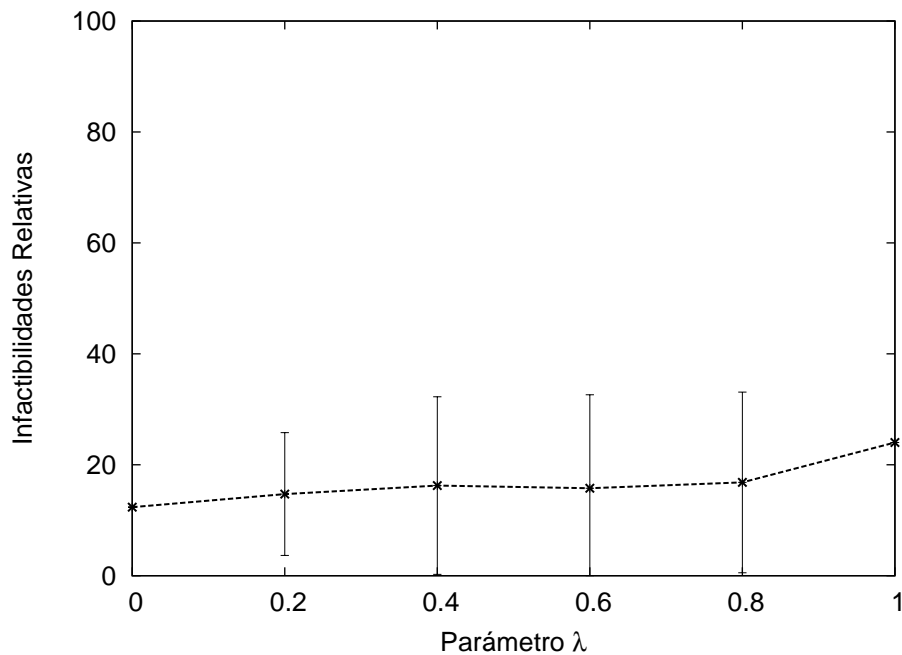


Figura 5.1: Infactibilidades con respecto al parámetro  $\lambda$ .

Como se puede apreciar, tanto el valor de  $\lambda = 0.6$  como el de  $\lambda = 0.8$  ofrecen un buen compromiso para ambos valores de la función voraz. Finalmente, decidimos establecer a  $\lambda = 0.8$  como el mejor valor debido a que un mayor peso en la función voraz ayuda a la construcción de territorios con una mejor medida de dispersión sin perder el compromiso con la suma de infactibilidades, como se observó.

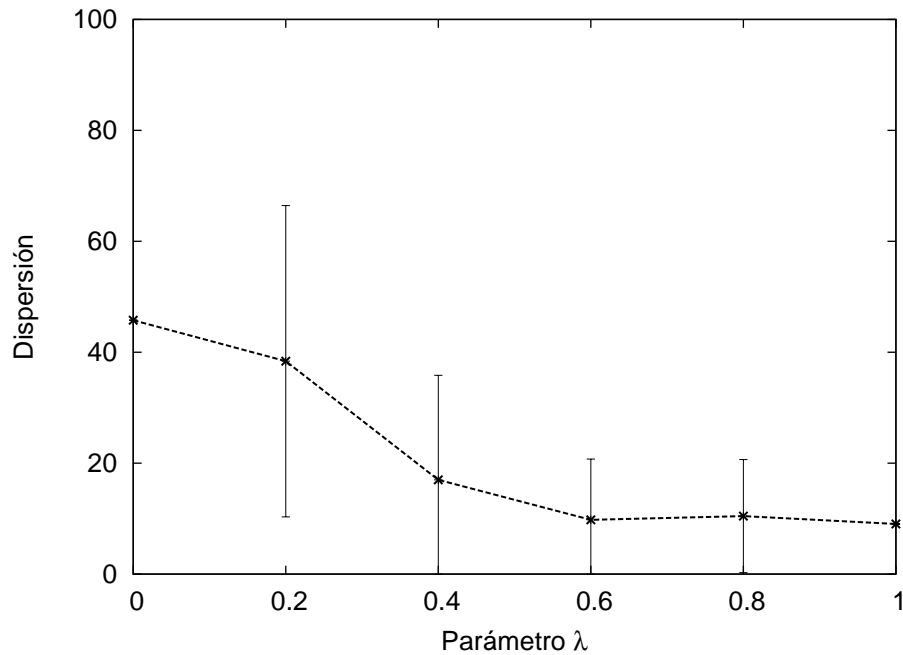


Figura 5.2: Dispersión con respecto al parámetro  $\lambda$ .

### 5.3.3 EXPERIMENTO C: PARÁMETROS $\sigma$ Y $\gamma$

Los parámetros  $\sigma$  y  $\gamma$  son parámetros auto ajustables que se encuentran ponderando al término del costo de ruteo y al término de infactibilidades en la función de mérito de la búsqueda local, respectivamente. En esta sección analizaremos el comportamiento de ambos parámetros considerando casos en los que se mantienen fijos y el caso en que son autoajustables. La idea es precisamente mostrar la valía que tiene el carácter de autoajuste de los parámetros como estrategia de búsqueda local.

Los parámetros  $\sigma$  y  $\gamma$  tienen un impacto directo con el número de movimientos de nuestra búsqueda local (el cual tiene un valor máximo de  $\text{MAX} = 5000$ ), además, su comportamiento es trascendental para encontrar soluciones factibles. Es por esto que en algunos casos fijamos el valor del parámetro en uno para identificar el impacto que tiene éste sobre el algoritmo.

La Tabla 5.2 muestra los resultados de aplicar el procedimiento de búsqueda

Parámetros		Movimientos	Soluciones
Ajustable	Fijo	Búsqueda Local	Factibles
$\sigma$	$\gamma$	258	15
$\sigma$	$\gamma$	314	14
$\gamma$	$\sigma$	MAX	0
	$\sigma$	MAX	0

Tabla 5.2: Funcionamiento de los parámetros  $\sigma$  y  $\gamma$ .

local bajo diferentes condiciones de  $\sigma$  y  $\gamma$  para 15 instancias del problema. En la primera columna se muestra cuál de los parámetros se mantuvo fijo y cuál auto ajustable

Puede apreciarse que cuando los parámetros se auto ajustan durante el transcurso del algoritmo siempre se obtiene una solución factible. También observamos que el número promedio de movimientos de la búsqueda local es de 258. Cuando fijamos  $\gamma$ , el número de iteraciones aumenta un poco y, además, se encontró un caso donde la solución es infactible. Fijar este parámetro que pondera las infactibilidades con respecto al balanceo nos cuesta al momento de obtener las soluciones.

Cuando fijamos  $\sigma$  nos encontramos que la búsqueda local siempre realizaba el número máximo de movimientos permitidos para la misma. Esto además trae a consecuencia que no se puede encontrar una solución factible. Lo mismo ocurrió cuando fijamos ambos parámetros con el valor de uno.

Los mejores resultados se obtienen cuando a ambos parámetros se les permite el auto ajuste.

## 5.4 EVALUACIÓN DEL ALGORITMO

Después del ajuste paramétrico se evalúa ahora la eficiencia del algoritmo. Es importante observar el desempeño de nuestro algoritmo con relación al tiempo de

ejecución. Es por esto que ejecutamos nuestro algoritmo sobre 15 instancias diferentes para observar el tiempo de cómputo del mismo. En la Tabla 5.3 podemos observar el promedio y la desviación estándar de los tiempos de ejecución de cada una de las fases del algoritmo: Construcción, Ajuste y Postprocesamiento (Búsqueda Local). El

Componente \ Tiempo	Promedio (s)	Desviación Estándar (s)	Distribución Promedio (%)
Construcción	50.18	1.49	2.68
Ajuste	9.21	0.52	0.90
Postprocesamiento	1799.93	239.35	96.42
Total	1866.76	240.25	100

Tabla 5.3: Análisis de los tiempos de ejecución del algoritmo GRASP.

tiempo promedio de nuestro algoritmo fue de 1866.76 segundos con una desviación estándar de 240.25 segundos, se observa que la fase que más tiempo consume es la fase de postprocesamiento con un 96.42% del tiempo total. Tanto la fase de construcción como la fase de ajuste prácticamente permanecen constantes, tienen una muy pequeña desviación, esto conlleva a que la desviación total del tiempo es debida en su mayoría a la fase de Postprocesamiento.

En el resto de la sección analizamos el desenvolvimiento de nuestro algoritmo, además de analizar y comparar las soluciones arrojadas por el mismo. Comprobar o establecer pautas de robustez del mismo es algo importante, es por esto que analizaremos la implementación del algoritmo sobre diversos valores de  $p$  (número deseado de territorios) y  $\tau$  (desviación permitida del tamaño de los territorios con respecto a la medida perfecta de balanceo de las actividades nodales). Como experimento final, presentamos una comparación entre nuestro algoritmo y otro que solo toma en cuenta la dispersión como objetivo, esto con la finalidad de observar las diferencias entre las configuraciones de red de ambos tipos de soluciones.

### 5.4.1 EXPERIMENTO D: DIFERENTES SOLUCIONES

El algoritmo presentado en este trabajo arroja tres tipos distintos de soluciones: la solución con respecto al mejor valor de la función objetivo ( $S^{\text{obj}}$ ), la solución con respecto al mejor valor de dispersión ( $S^{\text{disp}}$ ) y la solución con respecto al mejor valor del costo de ruteo ( $S^{\text{ruteo}}$ ). En este experimento observamos solo una comparación entre los tres tipos de solución, ya que proveen una variedad para el tomador de decisiones. Tener varios tipos de soluciones es una ventaja, pero sería aún mejor que en ese conjunto encontráramos soluciones de calidad.

		$S^{\text{obj}}$	$S^{\text{disp}}$	$S^{\text{ruteo}}$
Dispersión	Promedio	3.40	0	17.87
	Desviación	3.85	0	14.65
Costo de Ruteo	Promedio	1.27	3.85	0
	Desviación	1.42	2.34	0

Tabla 5.4: Comparación entre los tres tipos de soluciones.

La Tabla 5.4 muestra los valores promedios y la desviación estándar de los objetivos de dispersión y costo de ruteo para las tres soluciones. Observamos que  $S^{\text{obj}}$  establece un buen compromiso entre ambos objetivos, mientras que  $S^{\text{disp}}$  y  $S^{\text{ruteo}}$  aportan el mejor valor para cada uno de los objetivos, respectivamente, sin perder demasiado en la calidad del otro.

### 5.4.2 EXPERIMENTO E: VARIACIÓN DEL PARÁMETRO $p$

El parámetro  $p$  es sin duda trascendental para el algoritmo. El valor real de este parámetro es de alrededor de 40, es por esto que tomamos en cuenta tres diferentes valores  $p = 30$ ,  $p = 40$  y  $p = 50$ . Al comparar diferentes números de territorios deseados no podremos comparar las soluciones arrojadas por los mismos, es por esto que solamente se hace un análisis de los tiempos de cómputo y de las infactibilidades

para cada  $p$ .

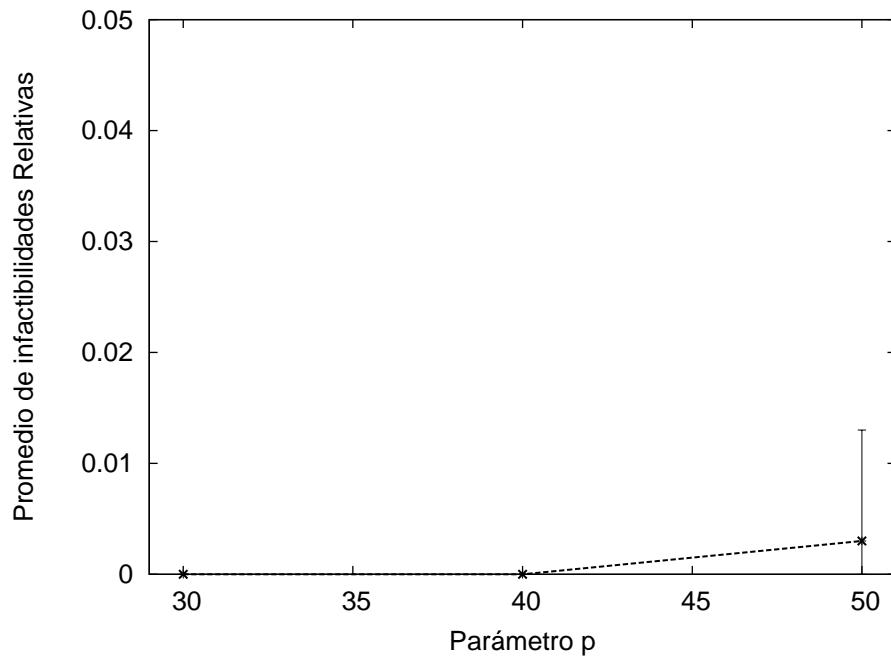


Figura 5.3: Comparación de la suma de infactibilidades con respecto a los diversos valores de  $p$ .

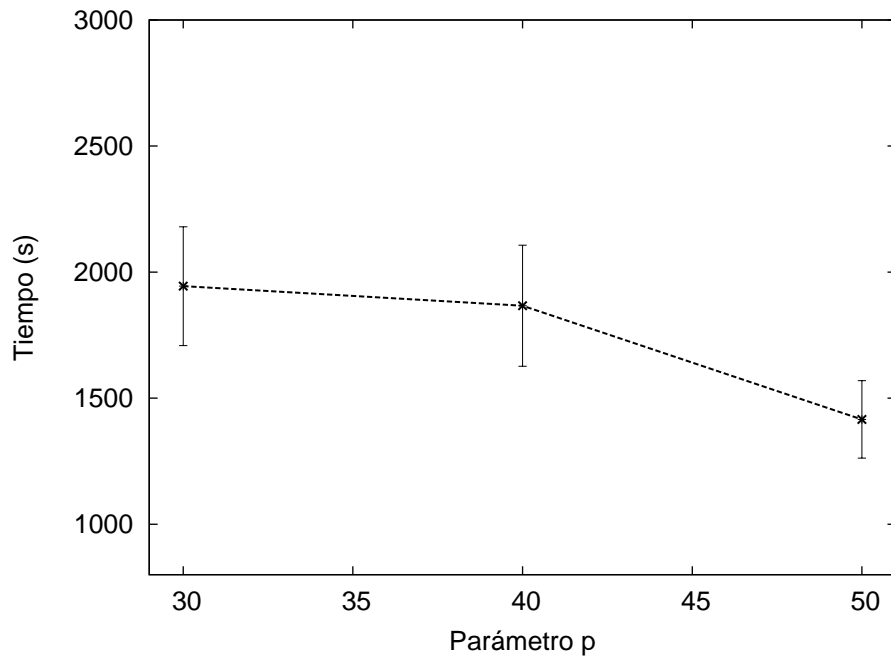


Figura 5.4: Comparación de los tiempos de cómputo con los diversos valores de  $p$ .



Se probaron 15 diferentes instancias para cada valor de  $p$ , de las cuales se obtuvo el promedio y además se observó la desviación estándar para cada caso.

En la gráfica de la Figura 5.3 podemos observar que para 30 y 40 territorios siempre se encuentran soluciones factibles. El problema se comienza a presentar cuando  $p = 50$  donde encontramos dos soluciones infactibles de las 15 instancias. El tiempo de cómputo empleado por el algoritmo se incrementa mientras el valor de  $p$  disminuye, como lo observamos en la gráfica de la Figura 5.4.

### 5.4.3 EXPERIMENTO F: VARIACIÓN DEL PARÁMETRO $\tau^a$

Para analizar el efecto que tiene el parámetro  $\tau^a$  en el desenvolvimiento del algoritmo, decidimos emplear diferentes valores del mismo:  $\tau^a \in \{0.05, 0.04, 0.03\}$ ,  $a = 1, 2$ .

		$\tau^a = 0.05$	$\tau^a = 0.04$	$\tau^a = 0.03$
Infactibilidad Relativa	Promedio	0.00000	0.01292	0.19874
	Desviación	0.00000	0.02585	0.22230
Soluciones Factibles		15	10	0

Tabla 5.5: Comparación de los diferentes valores de tolerancia  $\tau^a$ .

Lo interesante en esta sección es analizar el alcance de nuestro algoritmo con respecto a la tolerancia del balanceo territorial. Para ello comparamos la infactibilidad relativa y el número de soluciones factibles que se obtuvieron al ejecutar nuestro algoritmo sobre las 15 instancias para cada valor de  $\tau^a$ . Si observamos la Tabla 5.5 nos damos cuenta que para una tolerancia de  $\tau^a = 0.05$  el algoritmo no tiene problema en encontrar soluciones factibles. Conforme  $\tau^a$  disminuye, la dificultad se incrementa y, como lo vemos, cuando  $\tau^a = 0.04$  el número de soluciones factibles que se pueden encontrar se va limitando hasta el punto que cuando  $\tau^a = 0.03$ , una solución factible es difícil de encontrar.

Como observamos, nuestro algoritmo puede resolver instancias con un valor  $\tau^a = 0.05$ . Si disminuimos este valor, es más difícil encontrar soluciones factibles ya que el espacio de soluciones se reduce. Es por esto que cuando  $\tau^a = 0.03$  es prácticamente imposible obtener una solución.

#### 5.4.4 EXPERIMENTO G: MEJORAS EN LA SOLUCIÓN DEL ALGORITMO

Nuestro último experimento compara la solución de nuestro algoritmo (que considera dispersión y costos de ruteo en el proceso de decisión), al cual llamaremos ALG2, contra la solución obtenida con un algoritmo que solamente considera la dispersión durante la formación de territorios, al cual llamaremos ALG1, y al final se evalúan los costos de ruteo en la solución. De los tres tipos de soluciones que arroja el algoritmo ALG2 (la solución con respecto al mejor valor de la función objetivo, la solución con respecto al mejor valor de dispersión y la solución con respecto al mejor valor del costo de ruteo) solo tomamos en cuenta la solución con respecto al mejor valor de la función objetivo ( $S^{obj}$ ) para este experimento.

En la Tabla 5.6 se resumen los resultados al aplicar los algoritmos ALG1 y ALG2 a 15 instancias. Las últimas dos columnas reflejan la mejora relativa obtenida con ALG2 con respecto a ALG1,  $100 \times \frac{(f_{ALG2} - f_{ALG1})}{f_{ALG1}}$ , con respecto a la dispersión y al costo de ruteo.

Es evidente que un valor negativo refleja en realidad un deterioro. Por ejemplo, para la instancia 1, ALG2 obtiene una solución que mejora en un 67.45 % el costo de ruteo, aunque su dispersión se deteriora un 9.29 %

Podemos notar que en la mayoría de los casos las soluciones del algoritmo ALG1 proveen un mejor valor de la dispersión: la máxima mejora del algoritmo ALG2 es de 5.53 %, mientras que la mínima mejora es de -21 %. Con respecto al costo de ruteo la historia cambia drásticamente. El algoritmo ALG1 arroja evaluaciones muy altas para el costo de ruteo, en cambio el algoritmo ALG2 provee soluciones que

mejoran las del algoritmo ALG1 desde un 64.87 % hasta un 72.25 %. En promedio, la solución del algoritmo ALG2 con respecto a la dispersión se deteriora un 8.51 %, mientras que con respecto al costo de ruteo tiene una mejora del 67.69 %.

Estos porcentajes nos hacen ver que la naturaleza de exclusión que el algoritmo ALG1 tiene con respecto al costo de ruteo hace que las soluciones obtenidas no estén preparadas para una evaluación buena del mismo. Si bien, con el algoritmo ALG2 obtenemos soluciones que pierden, en la mayoría de los casos, un poco con respecto al criterio de dispersión, si ganan demasiado con respecto a una evaluación del costo de ruteo.

Este experimento confirmó nuestra hipótesis de que al introducir un término para la evaluación del costo de ruteo durante el proceso de decisión podríamos obtener una buena evaluación sin perder la calidad en el valor de la dispersión. Si queremos dar un mejor peso a la dispersión no olvidemos que contamos con la solución  $S^{\text{disp}}$  que nos otorgará el mejor valor con respecto a ella, esto sin perder calidad en el costo de ruteo, como lo observamos en la Sección 5.4.1.

Instancia	ALG1		ALG2		Mejora (%)	
	Dispersión	Costo de Ruteo	Dispersión	Costo de Ruteo	Dispersión	Costo de Ruteo
1	149.43	55404.96	163.32	18030.40	-9.29	67.45
2	158.41	52416.68	191.68	18156.30	-21.00	65.36
3	165.41	63616.80	190.38	17650.10	-15.09	72.25
4	161.79	50940.50	175.20	17842.60	-8.28	64.97
5	159.25	51564.56	178.84	17685.50	-12.30	65.70
6	166.20	51942.85	163.48	17662.60	1.63	65.99
7	161.60	55233.32	170.77	17641.70	-5.67	68.05
8	155.85	58822.75	180.06	18289.00	-15.53	68.90
9	167.26	52534.79	188.39	16945.70	-12.63	67.74
10	165.13	55283.07	156.00	17732.20	5.53	67.92
11	169.17	56819.06	183.36	17656.70	-8.38	68.92
12	161.78	60735.29	173.22	18106.00	-7.06	70.18
13	160.79	53560.09	175.98	17699.60	-9.44	66.95
14	163.36	54226.95	182.67	17270.10	-11.82	68.15
15	170.82	49771.46	170.33	17483.90	0.28	64.87
Promedio	162.42	54858.21	176.25	17723.49	-8.51	67.69

Tabla 5.6: Comparación de los algoritmos ALG1 y ALG2.

## CAPÍTULO 6

# CONCLUSIONES Y CONSIDERACIONES FINALES

---

### 6.1 CONCLUSIONES

Este trabajo abordó la idea de introducir una evaluación de costos de ruteo en un problema de diseño territorial basado en una métrica de distancias de red para una empresa distribuidora de bebidas de la ciudad de Monterrey, NL, México. En el estudio de la literatura observamos el vasto trabajo que se ha hecho con respecto al diseño territorial en sus diferentes ámbitos, pero vemos que el trabajo que se ha realizado con respecto a la introducción de costos de ruteo en este tipo de problemas es relativamente poco.

Se estableció un modelo combinatorio para la descripción del problema que fue formalizada mediante la teoría de grafos. Es aquí donde se introdujeron los costos de ruteo en la función objetivo. La evaluación de estos costos no era trivial, es por esto que observamos primero la estructura de nuestro problema. Detectamos que era más sencillo realizar una evaluación de costos de ruteo para cada territorio en particular, por lo cual se introdujo un enfoque del Problema del Agente Viajero (TSP). Otra observación trascendental fue que estos costos de ruteo realmente no representan un recorrido del TSP sino simplemente un recorrido en el que cada nodo es visitado al menos una vez.

Para resolver estos subproblemas de TSP se usó un método de ramificación y corte que además de proveer la solución óptima, lo hace en un lapso de tiempo demasiado pequeño. Esto facilitó la aplicación de esta metodología de evaluación de costos de ruteo.

Se demostró que el problema es *NP*-duro. Por tanto, se desarrolló un método heurístico de solución. Este método desarrollado es un algoritmo GRASP basado en un método de construcción con una métrica de distancias de red sin considerar los costos de ruteo. La búsqueda local considera la distancia de red como métrica, además de incluir la evaluación de costos de ruteo en el proceso de decisión. Se introdujeron además, dos parámetros autoajustables en la función de mérito de la búsqueda local; uno pondera el término de costos de ruteo y el otro el término de la suma de infactibilidades. Esto ayudó grandemente al comportamiento del algoritmo, de manera que un óptimo local era encontrado con rapidez.

Se introdujeron dos heurísticas para el recálculo de los costos de ruteo durante los movimientos de la búsqueda local. Esto debido a que la ejecución del algoritmo de ramificación y corte era en suma muy costoso. De esta manera se introdujo un umbral para poder invocar al método exacto cada cierto número de iteraciones y así, conservar la evaluación de los costos de ruteo cerca del valor óptimo de la solución actual.

El algoritmo diseñado en este trabajo provee tres diferentes soluciones como lo son: la solución con mejor valor de la función objetivo, la solución con el mejor valor de dispersión y la solución con el mejor valor del costo de ruteo. Esto provee de cierta diversidad al tomador de decisiones para escoger la solución que más le convenga sin perder calidad en las mismas.

Durante la experimentación se ajustaron los parámetros del algoritmo para después seguir con la evaluación del mismo. Pudimos observar que el costo de ruteo siempre descendió en cada iteración de la búsqueda local y en cambio, la suma de infactibilidades tardaba en disminuir. Los diferentes experimentos proporcionaron

una idea de la calidad del algoritmo. Los tiempos de cómputo son de alrededor de 2000 segundos en promedio. Al comparar las diferentes soluciones del algoritmo, se observó que la función objetivo realmente proporcionaba un buen compromiso entre los objetivos de dispersión y costos de ruteo, además de proveer con la solución más cercana a la factibilidad total.

Con respecto a los parámetros que son definidos por la empresa, en este caso  $p$  (número de territorios) y  $\tau^a$  (desviación de la medida perfecta de balanceo de la actividad  $a$ ), se realizaron dos experimentos para observar la robustez del algoritmo. Se concluyó que mientras el valor de  $p$  es más grande, más rápidamente se ejecuta el algoritmo y que mientras el valor de  $p$  es más pequeño, la suma de infactibilidades disminuye. Para el parámetro  $\tau^a$  se confirmó que en cuanto más pequeño su valor, es más difícil obtener soluciones factibles.

Finalmente, nuestro último experimento nos ayudó a tener una idea de la calidad de nuestro algoritmo. Al compararlo con un algoritmo que solo considera la dispersión como objetivo, pudimos observar y concluir que cuando consideramos el costo de ruteo durante el proceso de decisión, la configuración de la solución obtenida es de mucho mayor calidad. Rotundamente, nuestro algoritmo arrojó mejores soluciones con respecto a la evaluación de los costos de ruteo y valores muy cercanos al objetivo de la dispersión. De esta manera estamos obteniendo soluciones que pierden un poco en cuestión de dispersión pero ganan mucho más en cuestión de los costos de ruteo.

## 6.2 CONTRIBUCIONES

Diversas contribuciones pudieron realizarse a lo largo de este trabajo, las cuales se listan a continuación:

- Se logró un entendimiento completo del problema de manera que un enfoque de distancias de red fue establecido para el mismo.

- Se introdujo un modelo que considera como objetivo una suma ponderada de la dispersión territorial y los costos de ruteo.
- Se desarrolló una metodología de solución basada en la metaheurística GRASP, donde la fase de construcción aborda un enfoque de distancias de red, y la búsqueda local enfrentó tanto el recálculo de distancias, como la implementación de métodos exactos (ramificación y corte) y heurísticos para la evaluación de costos de ruteo.
- Se lograron resultados muy satisfactorios con evaluación de costos de ruteo sin dejar fuera los requerimientos de la empresa. Estos resultados son los primeros que se obtienen para este problema.
- Se desarrolló una herramienta donde se implementó la metodología de solución y los conocimientos adquiridos durante todo el desarrollo del presente trabajo. Dicha herramienta provee tres soluciones que dan al tomador de decisiones cierta diversidad, y de esta manera poder elegir la que más se ajuste a sus necesidades.

## 6.3 TRABAJO A FUTURO

Nuestro problema deja abiertas muchas puertas para un posible estudio o extensión del mismo. Algunas de ellas se listan en esta sección.

### 6.3.1 MODELO DE PROGRAMACIÓN ENTERA

Se presentó una formulación de optimización combinatoria del problema. Sin embargo, para probar soluciones óptimas de instancias pequeñas o inclusive desarrollar esquemas de acotamiento inferior el siguiente paso es el desarrollo del correspondiente modelo de programación entera.



### 6.3.2 MEJORAS DE LA METAHEURÍSTICA GRASP

La función voraz de la fase de construcción de nuestra metodología GRASP solo incluye dos términos y ninguno de ellos hace referencia a los costos de ruteo. Así, se puede desarrollar una función voraz en la que eficientemente sean introducidos estos costos de manera que el tiempo de ejecución del algoritmo no se vea afectado.

Otra parte importante, es considerar la distancia de red dinámica en la etapa de construcción, ya que simplemente utilizamos las distancias de red en forma estática. El problema está relacionado con el tiempo que toma el recálculo de distancias para cada unidad básica dentro del territorio en cuestión. Pero, si se eficientiza aún más lo realizado en la búsqueda local, es posible implementarlo en la construcción.

Una parte importante que puede mejorar la eficiencia del GRASP es la reactividad del parámetro  $\alpha$ . De esta manera, el parámetro  $\alpha$  tomaría el valor que más convenga o que históricamente haya dado mejores resultados en las regiones en cuestión.

### 6.3.3 BÚSQUEDA LOCAL ESPECIALIZADA

La calidad de soluciones que provee el algoritmo GRASP es relativamente buena y a partir de aquí, se puede implementar un algoritmo de búsqueda local más especializado para tratar de mejorar las soluciones obtenidas. Por ejemplo, un algoritmo de Búsqueda Local Iterada [45] (IGLS, por sus siglas en inglés), puede dar buenos resultados ya que ha sido exitoso en el área de secuenciación. Se caracteriza principalmente por una fase destructiva aleatoria y una fase de reconstrucción voraz. Además de esto, se puede aplicar una Búsqueda de Entornos Variables (VNS, por sus siglas en inglés) para obtener una mayor eficiencia del mismo.

### 6.3.4 CONSIDERACIONES DEL PROBLEMA

Los supuestos considerados para la resolución del problema dejan fuera casos que pueden presentarse en el mundo real. Tal es el caso de la demanda y de los

posibles cambios que se pueden presentar en la red de distribución. En nuestro caso, consideramos la demanda de cada cliente como determinista. Una extensión natural del trabajo es considerar la demanda como parámetro aleatorio lo cual abre la puerta a un modelo de optimización estocástica y por ende a una metodología de solución radicalmente diferente.

Los cambios en la red de distribución representan lo que se conoce como re-optimización de la red. Incluir una reoptimización durante el proceso de solución depende a su vez de la información que la empresa tenga, pero puede ocurrir en cualquier momento. La complejidad de nuestro problema aumentaría pero resolvería las problemáticas que se encuentran en el mundo real.

# BIBLIOGRAFÍA

---

- [1] R. Ahuja, T. L. Magnanti y J. Orlin. *Network Flows: Theory, Algorithms, and Applications*. Prentice Hall, Upper Saddle River, EUA, 1993.
- [2] D. Applegate, R. Bixby, V. Chvatal y W. Cook. TSP cuts which do not conform to the template paradigm. En M. Jünger y D. Naddef, editores, *Computational Combinatorial Optimization: Optimal or Provably Near-Optimal Solutions*, volume 2241 of *Lecture Notes in Computer Science*, páginas 261–304. Springer, Berlín, Alemania, 2001.
- [3] P. K. Bergey, C. T. Ragsdale y M. Hoskote. A decision support system for the electrical power districting problem. *Decision Support Systems*, 36(1):1–17, 2003.
- [4] P. Bertolazzi, L. Bianco y S. Ricciardelli. A method for determining the optimal districting in urban emergency services. *Computers & Operations Research*, 4(1):1–12, 1977.
- [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 E. Alba, S. Chicano, F. Herrera, F. Luna, G. Luke y A. J. Nebro, editores, *Actas de la I Reunión Española sobre Algoritmos Evolutivos y Metaheurísticos*, páginas 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*, páginas 422–427, IJIE, Cincinnati, EUA, 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] D. Cattrysse, D. Van-Oudheusden y T. Lotan. The problem of efficient districting: Improving distributed operations depends on efficient partitioning of a service area into districts. *OR Insight*, 10(4):9–13, 1997.
- [9] C. W. Chance. Political studies: Number 2, representation and reappointment. Reporte técnico, Department of Political Science, Ohio State University, Columbus, EUA, 1965.
- [10] L. Chapleau, J. A. Ferland y J. M. Rousseau. Clustering for routing in densely populated areas. *European Journal of Operational Research*, 20(1):48–57, 1985.
- [11] V. Chvátal. Edmonds polytopes and weakly Hamiltonian graphs. *Mathematical Programming*, 5(1):29–40, 1973.
- [12] S. J. D’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.
- [13] G. B. Dantzig, D. R. Fulkerson y S. Johnson. Solution of a large-scale traveling salesman problem. *Journal of the Operations Research Society of America*, 1(2):393–410, 1954.
- [14] R. Diestel. *Graph Theory*. Springer, Nueva York, EUA, 2006.
- [15] E. W. Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1(1):269–271, 1959.

- 
- [16] A. Drexler y K. Haase. Fast approximation methods for sales force deployment. *Management Science*, 45(10):1307–1323, 1999.
- [17] T. A. Feo y M. G. C. Resende. Greedy randomized adaptive search procedures. *Journal of Global Optimization*, 6(2):109–133, 1995.
- [18] J. A. Ferland y G. Guénette. Decision support system for the school districting problem. *Operations Research*, 38(1):15–21, 1990.
- [19] B. Fleischmann y J. N. Paraschis. Solving a large scale districting problem: A case report. *Computers & Operations Research*, 15(6):521–533, 1988.
- [20] R. W. Floyd. Algorithm 97: Shortest path. *Communications of the ACM*, 5(6):345, 1962.
- [21] E. Forrest. Apportionment by computer. *American Behavioral Scientist*, 23(7):23–35, 1964.
- [22] R. S. Garfinkel y G. L. Nemhauser. Solving optimal political districting by implicit enumeration techniques. *Management Science*, 16(8):B495–B508, 1970.
- [23] F. Glover y G. Kochenberger, editores. *Handbook of Metaheuristics*. Kluwer, Dordrecht, Holanda, 2002.
- [24] J. L. Gross y J. Yellen. *Handbook of Graph Theory*. CRC Press, Danvers, EUA, 2004.
- [25] M. Grötschel y M. W. Padberg. On the symmetric travelling salesman problem II: Lifting theorems and facets. *Mathematical Programming*, 16(1):281–302, 1979.
- [26] G. Gutin y A. P. Punnen, editores. *The Traveling Salesman Problem and its Variations*. Springer, Nueva York, EUA, 2002.
- [27] W. E. Hardy. Vehicle routing efficiency: A comparison of districting analysis and the Clarke-Wright method. *American Journal of Agricultural Economics*, 62(3):534–536, 1980.

- 
- [28] D. Haugland, S. C. Ho y G. Laporte. Designing delivery districts for the vehicle routing problem with stochastic demands. *European Journal of Operational Research*, 180(3):997–1010, 2007.
- [29] S. W. Hess y S. A. Samuels. Experiences with a sales districting model: Criteria and implementation. *Management Science*, 18(4):P41–P54, 1971.
- [30] S. W. Hess, J. B. Weaver, H. J. Siegfeldt, J. N. Whelan y P. A. Zitlau. Nonpartisan political redistricting by computer. *Operations Research*, 13(6):998–1006, 1965.
- [31] M. Hojati. Optimal political districting. *Computers & Operations Research*, 23(12):1147–1161, 1996.
- [32] J. Kalcsics, T. Melo, S. Nickel y H. Gündra. Planning sales territories: A facility location approach. En *Operations Research Proceedings 2001*, páginas 141–148. Springer, Berlín, Alemania, 2002.
- [33] J. Kalcsics, S. Nickel y M. Schröder. Towards a unified territory design approach: Applications, algorithms, and GIS integration. *Top*, 13(1):1–56, 2005.
- [34] D. E. Knuth. *The Art of Computer Programming*. Addison-Wesley, Boston, EUA, 1997.
- [35] A. Land. The solution of some 100-city travelling salesman problems. Reporte técnico, London School of Economics, Londres, Reino Unido, 1979.
- [36] G. Laporte. A short history of the traveling salesman problem. Reporte técnico, Centre for Research on Transportation (CRT) and GERAD, HEC, Montreal, Canadá, 2006.
- [37] P. G. Marlin. Application of the transportation model to a large-scale districting problem. *Computers & Operations Research*, 8(1):83–96, 1981.
- [38] P. Miliotis. Using cutting planes to solve the symmetric travelling salesman problem. *Mathematical Programming*, 15(1):177–188, 1978.

- [39] L. Muyldermans, D. Cattrysse, D. Van Oudheusden y T. Lotan. Districting for salt spreading operations. *European Journal of Operational Research*, 139(3):521–532, 2002.
- [40] C. H. Papadimitriou y K. Steiglitz. *Combinatorial Optimization: Algorithms and Complexity*. Dover, Mineola, EUA, 1998.
- [41] F. Pezzella y B. Nicoletti. A system approach to the optimal health-care districting. *European Journal of Operational Research*, 8(2):139–146, 1981.
- [42] G. Reinelt. TSPLIB - A traveling salesman problem library. *ORSA Journal on Computing*, 3(4):376–384, 1991.
- [43] F. Ricca y B. Simeone. Local search algorithms for political districting. *European Journal of Operational Research*, 189(3):1409–1426, 2008.
- [44] R. Z. Ríos-Mercado y E. A. Fernández. A reactive GRASP for a commercial territory design problem with multiple balancing requirements. *Computers & Operations Research*, 36(3):755–776, 2009.
- [45] R. Ruiz y T. Stützle. A simple and effective iterated greedy algorithm for the permutation flowshop scheduling problem. *European Journal of Operational Research*, 177(3):2033–2049, 2007.
- [46] J. A. Segura-Ramiro, R. Z. Ríos-Mercado, A. M. Álvarez Socarrás y K. de Alba-Romenus. A location-allocation heuristic for a territory design problem in a beverage distribution firm. 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*, páginas 428–434, IJIE, Cincinnati, EUA, 2007.
- [47] R. J. Shanker, R. E. Turners y A. A. Zoltners. Sales territory design: An integrated approach. *Management Science*, 22(3):309–320, 1975.
- [48] P. Toth y D. Vigo, editores. *The Vehicle Routing Problem*. Monographs on Discrete Mathematics and Applications. SIAM, Philadelphia, EUA, 2002.

- 
- [49] W. T. Tutte. *Graph Theory*. Cambridge University Press, Cambridge, Reino Unido, 2001.
- [50] D. Van-Oudheusden, D. Cattrysse y T. Lotan. On the importance of districting and its potential impact on routing. En *The 8th World Conference on Transportation Research, Proceedings 2*, páginas 521–531, Antwerp, Bélgica, 1998.
- [51] 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, *Actas del IV Congreso Español en Metaheurísticas, Algoritmos Evolutivos y Bioinspirados, Tomo II*, páginas 609–617, Granada, España, septiembre 2005.
- [52] L. A. Wolsey. *Integer Programming*. Wiley, Nueva York, EUA, 1998.
- [53] A. A. Zoltners y P. Sinha. Sales territory alignment: A review and model. *Management Science*, 29(11):1237–1256, 1983.



## APÉNDICE A

# MÉTODO DE RAMIFICACIÓN Y CORTE PARA EL PROBLEMA DEL AGENTE VIAJERO

---

El problema del agente viajero (TSP por sus siglas en inglés) es el problema de encontrar un tour o ciclo Hamiltoniano en el que todos los vértices de un grafo son visitados exactamente una vez, reduciendo los costos de este viaje. Los costos de viaje son generalmente los costos de transportación, distancias, riesgos, etc. El TSP puede resolverse tanto en grafos dirigidos (TSP asimétrico) como en no dirigidos (TSP simétrico), además de requerirse que el grafo sea completo para asegurar que realmente existe un tour. Este apéndice se enfoca solamente en el TSP simétrico, debido a la naturaleza de nuestro problema. La formulación de programación entera del TSP tiene un conjunto de restricciones, denominadas restricciones de eliminación de subtours. El problema es difícil de resolver porque es *NP*-duro [26] por lo que no existe un algoritmo que pueda resolverlo en tiempo polinomial. Los métodos de ramificación y corte son una extensión de los métodos de ramificación y acotación [52], en los cuales en cada nodo del árbol de decisión se agregan restricciones válidas del problema en cuestión a las cuales se les denomina *cortes*. De esta manera se trata de reducir el árbol de decisión, ya que cada vez que se agrega un nuevo corte, se reduce el poliedro del envolvente convexo del problema. En particular, el TSP tiene su historia con respecto a la manera de resolverse. Laporte [36] ofrece una pequeña

reseña en la cual podemos observar una breve historia del TSP. En 1966 Martin [26] implementa las restricciones de eliminación de subtours además de introducir cortes de integralidad. Miliotis [38] introduce el primer método de ramificación y corte empleando un método de ramificación y acotación utilizando planos cortantes de Gomory [52] y restricciones de eliminación de subtours. En el año 1979, Land [35] introduce las restricciones “2-matching” de Edmonds [26]. Las desigualdades de peine son introducidas por Chvátal [11]. A partir de aquí, se comenzó a desarrollar la teoría de poliedros a través de la cual, nuevos y más eficientes algoritmos han sido implementados. Algunos de los autores que se han involucrado en la teoría de poliedros son: Fonlupt, Grötschel, Holland, Hong, Naddef, Padberg, Pulleyblank, Rinaldi y Thienel, entre otros [26]. Cabe destacar que actualmente se pueden resolver problemas de hasta 2392 vértices, cosa que anteriormente era imposible. A lo largo de este capítulo describiremos tanto la formulación y las desigualdades válidas del TSP, como el método de ramificación y corte que fue utilizado para resolverlo.

## A.1 FORMULACIÓN

Existen diversas formulaciones para describir al problema del agente viajero simétrico, pero tomaremos la formulación de Dantzig, Fulkerson y Johnson [13] para ello. Sea  $V$  el conjunto de  $n$  vértices y  $E$  el conjunto de aristas, sea  $d_e$  el costo de agregar la arista  $e$  al tour, definimos

$$x_e = \begin{cases} 1, & \text{si } e \in E \text{ pertenece al tour} \\ 0, & \text{en otro caso.} \end{cases}$$

La formulación de programación entera es la siguiente:

$$\text{mín } \sum_{e \in E} d_e x_e \tag{A.1}$$

sujeto a:

$$x(\delta(\{v\})) = 2, \quad v \in V \tag{A.2}$$

$$x(\delta(S)) \geq 2, \quad S \subseteq V, \quad 2 \leq |S| \leq n - 1 \tag{A.3}$$

$$x_e \in \{0, 1\}, \quad e \in E, \quad (\text{A.4})$$

donde (A.1) es la función objetivo diseñada para reducir los costos del tour, (A.2) asegura que para cada nodo siempre debe haber dos aristas asignadas, (A.3) son las llamadas restricciones de eliminación de subtour, que como mencionamos antes, existe un número exponencial de ellas y finalmente, (A.4) nos indica que las variables  $x_e$  son binarias. En (A.2) y (A.3) existe un término  $x(\delta(B))$ ,  $B \subset V$ , que significa el número de aristas adyacentes al conjunto de nodos  $B$ , o sea, son las aristas  $(i, j) \in E$  tal que,  $i \in B$  y  $j \in V \setminus B$  y además, deben ser tomadas como parte de la solución. El número total de variables resulta ser  $n(n-1)/2$  que representa el número total de aristas del grafo en cuestión. Cabe señalar, que solamente  $n$  variables tomarán el valor de 1.

## A.2 DESIGUALDADES VÁLIDAS PARA EL TSP

Una desigualdad válida para un problema de programación entera, es una restricción que se construye para delimitar el área del conjunto de soluciones factibles y es deseable que éstas formen parte de la envolvente convexa del poliedro en cuestión. Particularmente, para el problema del TSP se han desarrollado diversas desigualdades válidas para tratar de reducir el tan complejo poliedro de solución. En nuestro caso, consideraremos dos tipos de desigualdades válidas diferentes: desigualdades de eliminación de subtours y las llamadas desigualdades de peine. Ambas se describen detalladamente en esta sección.

### A.2.1 DESIGUALDADES DE ELIMINACIÓN DE SUBTOURS

Antes de entrar completamente en las desigualdades, es necesario definir el concepto geométrico de faceta, que no es más que cualquiera de las caras de un poliedro. La desigualdad de eliminación de subtour estará definida de la siguiente

manera: sea  $S \subset V$ , elegir al menos  $|S| - 1$  nodos tal que

$$x(S) \leq |S| - 1. \quad (\text{A.5})$$

**Teorema A.1** (Grötschel y Padberg [25]) *Para todo  $S \subseteq V$ , la desigualdad de eliminación de subtour (A.5) define una faceta de  $TSP(K_n)$  para  $n \geq 4$ .*

Aquí,  $K_n$  representa el número de tours posibles de  $n$ -vértices. Este resultado es favorecedor para encontrar la envolvente convexa del TSP, pero aún así es imposible deducir cuántas desigualdades de este tipo necesitaríamos para encontrar la solución óptima del problema. La manera de construir estas desigualdades es relativamente sencilla: primero resolvemos nuestro TSP relajando las restricciones de integralidad (A.4) y las restricciones de eliminación de subtour (A.5). Luego, en la solución localizamos un subtour (para lo cual podemos utilizar un algoritmo de búsqueda por profundidad o anchura [34]) y construimos la desigualdad de eliminación de subtours que está siendo violada por la presente solución. En la Figura A.1 podemos observar un ejemplo de cómo encontrar una desigualdad de eliminación de subtour, los cuales son representados con las aristas punteadas. Al introducir esta desigualdad a nuestro problema estaríamos restringiendo esta misma solución cuando resolvamos de nuevo la relajación lineal. Agregar muchas restricciones de este tipo puede ayudarnos a encontrar la solución óptima, pero muchas veces es necesario introducir desigualdades más especializadas, como lo son las desigualdades de peine, para encontrar la solución más rápidamente.

### A.2.2 DESIGUALDADES DE PEINE

Las desigualdades de peine constan de dos tipos de conjuntos: el mango y los dientes. Sea  $H \subset V$  el mango y sean  $T_1, T_2, \dots, T_s \subset V$ ,  $s \geq 3$  e impar, los dientes, tal que cumplen las siguientes propiedades:

$$H \cap T_i \neq \emptyset, \quad i = 1, 2, \dots, s$$

$$T_i \setminus H \neq \emptyset, \quad i = 1, 2, \dots, s$$

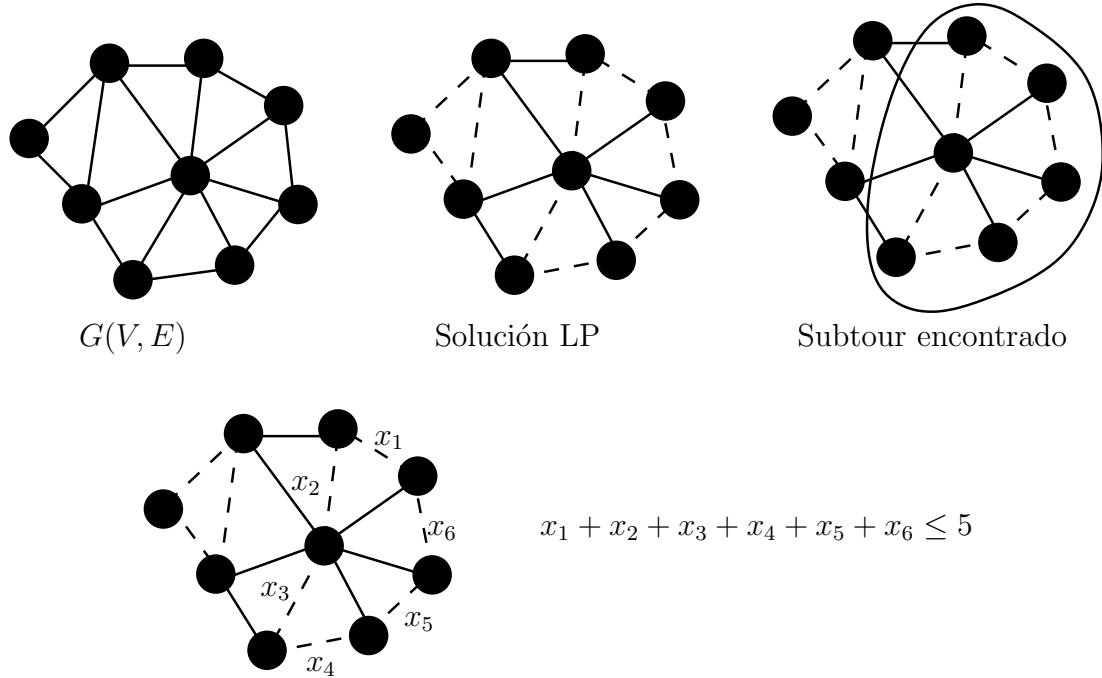


Figura A.1: Ejemplo de construcción de una desigualdad de eliminación de subtour.

$$T_i \cap T_j = \emptyset, \quad i \neq j, \quad i, j \leq s.$$

Así, definimos una desigualdad de peine de la siguiente forma:

$$x(\delta(H)) + \sum_{i=1}^s x(\delta(T_i)) \geq 3s + 1. \tag{A.6}$$

**Teorema A.2** (Grötschel y Padberg [25]) *Las desigualdades de peine (A.6) definen una faceta para el TSP ( $K_n$ ), para  $n \geq 6$*

Las restricciones de peine son generalización de las desigualdades “2-matching”, en las cuales los dientes son conjuntos de una arista. En la Figura A.2 se muestra un ejemplo de la generalización de una desigualdad de peine.

### A.2.3 ALGORITMO DE SEPARACIÓN

A diferencia de las restricciones de eliminación de subtour, no es tan trivial construir una desigualdad de peine. Es por esto que es necesario implementar un al-

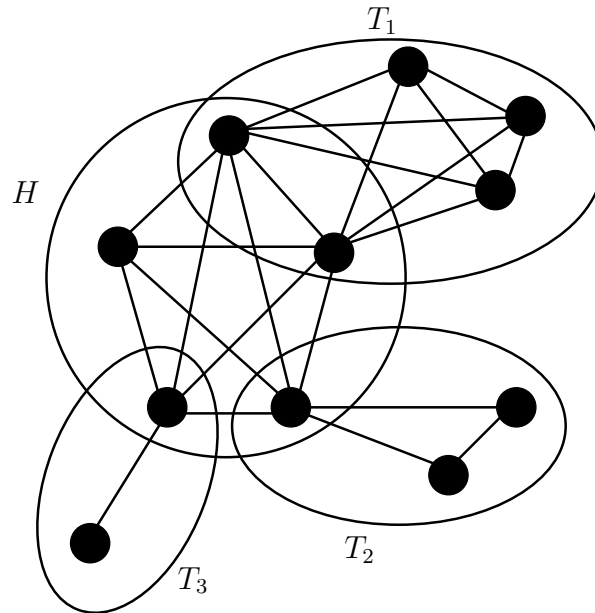
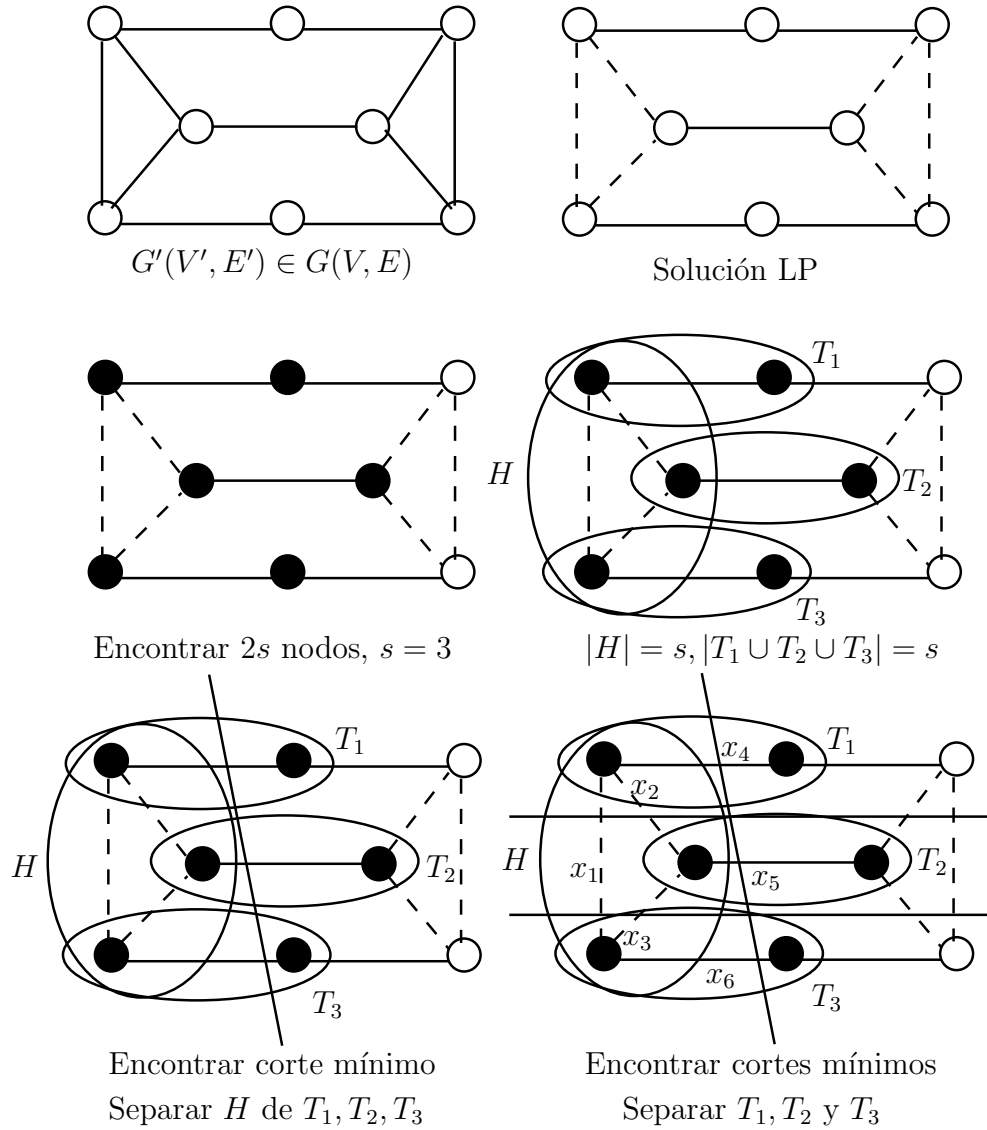


Figura A.2: Desigualdad de Peine.

goritmo de separación para ello. Al resolver la relajación lineal de un TSP (relajando la integralidad de las variables) podemos seguir el siguiente algoritmo para construir una desigualdad de peine:

1. Escoger  $2s$  vértices.
2. Escoger  $s$  vértices para que pertenezcan al mango,  $H$ .
3. Formar  $s$  pares disjuntos de vértices entre los que pertenecen al mango y los que no.
4. Encontrar el corte mínimo que separe los  $s$  vértices del mango de los otros  $s$ .
5. Para cada par de vértices asignados, encontrar un mínimo corte que los separe de los otros  $2s - 2$  vértices.

En la Figura A.3 mostramos un ejemplo donde podemos apreciar claramente los pasos a seguir para encontrar la desigualdad.



$$x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_1 + x_2 + x_4 + x_2 + x_3 + x_5 + x_1 + x_3 + x_6 \geq 10$$

Figura A.3: Ejemplo de construcción de una desigualdad de peine.

### A.3 MÉTODO DE RAMIFICACIÓN Y CORTE

Definidas las desigualdades válidas y el algoritmo de separación, podemos describir el procedimiento del algoritmo de ramificación y corte. El primer paso es eliminar las restricciones de eliminación de subtours y relajar las restricciones de

integralidad de las variables. De esta forma, la nueva formulación es la siguiente:

$$\text{mín} \sum_{e \in E} d_e x_e$$

sujeto a:

$$x(\delta(v)) = 2, \quad v \in V$$

$$0 \leq x_e \leq 1, \quad e \in E. \tag{A.7}$$

A partir del problema (A.7), que denotaremos por  $P$ , se construye el algoritmo de ramificación y corte (Figura A.4).

El pseudo código de la Figura A.5 es el método que se encarga de construir el corte e introducirlo en el problema actual. Esta rutina es llevada a cabo hasta que ningún corte es encontrado, o bien, si ya se obtuvo una solución entera. Cabe señalar que las rutinas para encontrar las desigualdades de eliminación de subtours y de peine fueron descritas en las secciones A.2.1 y A.2.3 (algoritmos de separación). En la Figura A.4 se muestra el algoritmo de ramificación y corte donde  $P$  representa la instancia del problema y  $P^i$  representa el  $i$ -ésimo problema de la lista  $L$  de problemas del árbol de decisión,  $X^i$  representa la solución del problema  $P^i$  y  $z^i$  el valor de la solución. Finalmente,  $\bar{z}$  y  $\underline{z}$  son las cotas superior e inferior durante el desarrollo del método, respectivamente.

## A.4 IMPLEMENTACIÓN DEL ALGORITMO

Para observar el desempeño del algoritmo de Ramificación y Corte, decidimos implementarlo en algunas instancias de la TSPLIB (biblioteca del TSP)[42]. Medimos algunas características del método, como los son el tiempo, el número de desigualdades válidas agregadas y el número de nodos del árbol de solución. La Tabla A.1 muestra los resultados obtenidos de la implementación del paquete CONCORDE desarrollado por Applegate et al. [2].

En la historia del TSP, se intentaron resolver diferentes instancias de diversos



---

*SolTSPExacto* ( $P$ )
 

---

**Input:**  $P$ : Instancia del problema**Output:**  $c$ : Costo de ruteo

```

1:  $X \leftarrow \emptyset$ ,  $X^{inc} \leftarrow \emptyset$ ,  $\underline{z} \leftarrow -\infty$ ,  $\bar{z} \leftarrow \infty$ ;
2:  $L \leftarrow \{P\}$ ;
3: while ( $L \neq \emptyset$ ) do
4:   Elegir  $P^i \in L$ ;
5:    $X^i \leftarrow \text{LP}(P^i)$ ,  $z^i \leftarrow f(X^i)$ ;
6:   if ( $X^i = \emptyset$ ) then
7:      $L \leftarrow L \setminus \{P^i\}$ ;
8:   end if
9:   if ( $z^i < \underline{z}$ ) then
10:     $L \leftarrow L \setminus \{P^i\}$ ;
11:    break;
12:  end if
13:  if ( $X^i \in \{0, 1\}$ ) then
14:    if ( $z^i < \bar{z}$ ) then
15:       $\bar{z} \leftarrow z^i$ ,  $X^{inc} \leftarrow X^i$ ;
16:    end if
17:     $L \leftarrow L \setminus \{P^i\}$ ;
18:  else
19:    if (EncontrarCorte ( $X^i$ ) = YES) then
20:      ir a 5;
21:    else
22:       $\{P_1, \dots, P_k\} \leftarrow \text{Ramificar}(P^i)$ ;
23:       $L \leftarrow L \cup \{P_1, \dots, P_k\}$ ,  $L \leftarrow L \setminus P^i$ ;
24:    end if
25:  end if
26: end while
27: return  $c$ ;

```

---

Figura A.4: Método de Ramificación y Corte para resolver el TSP

---

*EncontrarCorte* ( $P^i, X^i$ )

---

**Input:**  $X^i$ : Solución del problema  $P^i$ **Output:**  $P^i$ : Problema  $P^i$ 

```

1: if (Desigualdadsubtour( $X^i$ ) = TRUE) then
2:   AgregarCorte( $P^i$ )
3:   return YES
4: else
5:   if (DesigualdadPeine ( $X^i$ ) = TRUE) then
6:     AgregarCorte ( $P^i$ )
7:     return YES
8:   end if
9: end if
10: return NO

```

---

Figura A.5: Rutina que agrega un corte al problema  $P^i$ .

tamaños. En particular, el famoso problema de 42 ciudades de Dantzig se pudo resolver hasta el año de 1966. Si observamos el último renglón de la tabla solo fueron necesarios 0.11 segundos para encontrar la solución óptima del mismo.

Al analizar cada uno de los resultados, vemos que el algoritmo es demasiado eficiente, ya que el mayor tiempo de cómputo fue de 171.39 segundos y estamos hablando de una instancia de 1000 nodos. Los tiempos varían según el tamaño y el tipo de problema, pero si hablamos de problemas relativamente pequeños ( $n < 100$ ) el algoritmo tarda menos de 0.5 segundos para encontrar la solución óptima.

El mayor número de nodos del árbol de solución fue de 13 para los problemas Pa561 y Pr439. Para la mayoría de los problemas, solo se necesitó un nodo, lo cual nos habla de la importancia de la inclusión de las restricciones válidas. El mayor número de restricciones generadas fue de 3309 para un problema de 2392 nodos, pero conforme baja el tamaño del problema, el número de desigualdades necesarias es menor.

Como conclusión, se pudo constatar la eficiencia del método de ramificación y corte para el TSP. Como en nuestro caso específico se requiere resolver TSPs del orden de los 50-70 nodos para el cálculo del costo de ruteo en cada territorio, queda más que claro que este método nos brinda una vía excelente para este cálculo al reportar tiempos de solución inferior a los 0.5 segundos en problemas de tamaño similar.

Nombre	Tamaño	No. Cortes	No. Nodos	Tiempo (s)
Dsj1000	1000	2543	7	171.39
Pa561	561	2097	13	120.70
Pr439	439	1246	13	74.10
Gr299	299	441	5	4.18
Gil262	262	313	3	10.12
Eil101	101	13	3	0.34
Pr2392	2392	3309	1	51.38
Pr1002	1002	1455	1	20.60
Ali1535	535	814	1	8.70
A280	280	346	1	3.09
Pr226	226	339	1	1.20
D198	198	311	3	5.27
Brg180	180	230	1	1.94
Ch150	150	249	1	1.05
Gr120	120	200	1	0.74
Pr107	107	166	1	0.43
Gr96	96	161	1	0.48
Eil76	76	90	1	0.16
Brazil58	58	85	1	0.25
Dantzig42	42	55	1	0.11

Tabla A.1: Resultados del Método de Ramificación y Corte en instancias de la TSPLIB.

# FICHA AUTOBIOGRÁFICA

---

Juan Carlos Salazar Acosta

Candidato para el grado de Maestro en Ciencias en Ingeniería de Sistemas

Universidad Autónoma de Nuevo León

Facultad de Ingeniería Mecánica y Eléctrica

Tesis:

## DISEÑO DE TERRITORIOS COMERCIALES CON COSTOS DE RUTEO

Nací el 18 de marzo de 1985 en la ciudad de Monterrey, México. Soy el segundo hijo del Sr. Jorge Alberto Salazar Haro y de la Sra. María Elena Acosta Palos. Cursé mi licenciatura en la Facultad de Ciencias Físico Matemáticas de la Universidad Autónoma de Nuevo León obteniendo el grado de Licenciado en Matemáticas en el 2007. Posteriormente ingresé al posgrado en Ingeniería de Sistemas para cursar la maestría con el mismo nombre en la Facultad de Ingeniería Mecánica y Eléctrica donde comienza esta investigación bajo la asesoría del Dr. Roger Z. Ríos Mercado.