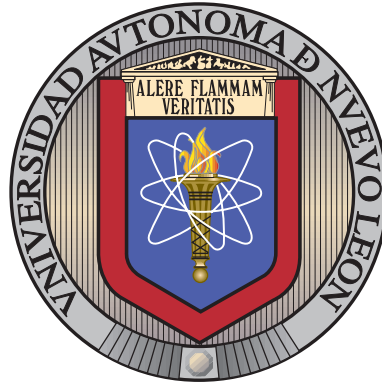


UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN

FACULTAD DE INGENIERÍA MECÁNICA Y ELÉCTRICA



OPTIMIZACIÓN DE LA LOGÍSTICA DE
TRANSPORTE Y DISTRIBUCIÓN DE
PRODUCTOS

POR

ROBERTO CANTU FUNES

COMO REQUISITO PARA OBTENER EL GRADO DE

MAESTRÍA EN CIENCIAS

EN INGENIERÍA DE SISTEMAS

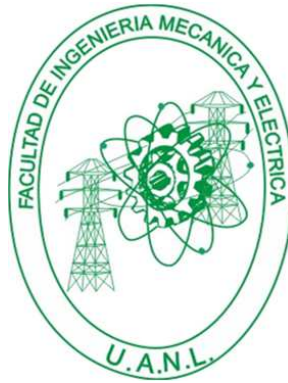
SAN NICOLÁS DE LOS GARZA, NUEVO LEÓN

MAYO 2016

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN

FACULTAD DE INGENIERÍA MECÁNICA Y ELÉCTRICA

SUBDIRECCIÓN DE ESTUDIOS DE POSGRADO



OPTIMIZACIÓN DE LA LOGÍSTICA DE
TRANSPORTE Y DISTRIBUCIÓN DE
PRODUCTOS

POR

ROBERTO CANTU FUNES

COMO REQUISITO PARA OBTENER EL GRADO DE

MAESTRÍA EN CIENCIAS

EN INGENIERÍA DE SISTEMAS

SAN NICOLÁS DE LOS GARZA, NUEVO LEÓN

MAYO 2016

Universidad Autónoma de Nuevo León
Facultad de Ingeniería Mecánica y Eléctrica
Subdirección de Estudios de Posgrado

Los miembros del Comité de Tesis recomendamos que la Tesis «OPTIMIZACIÓN DE LA LOGÍSTICA DE TRANSPORTE Y DISTRIBUCIÓN DE PRODUCTOS», realizada por el alumno Roberto Cantu Funes, con número de matrícula 1445951, sea aceptada para su defensa como opción al grado de Maestría en Ciencias en Ingeniería de Sistemas.

El Comité de Tesis

Dra. María Angélica Salazar Aguilar

Asesor

Dr. Vincent André Lionel Boyer

Revisor

Dr. Leonardo Gabriel Hernández Landa

Revisor

Vo. Bo.

Dr. Simón Martínez Martínez

Subdirector de Estudios de Posgrado

San Nicolás de los Garza, Nuevo León, mayo 2016

ÍNDICE GENERAL

Agradecimientos	x
Resumen	xi
1. Introducción	1
1.1. Metodología de investigación	5
1.2. Estructura de la tesis	6
2. Problema de distribución mono-producto	7
2.1. Descripción del problema	7
2.2. Trabajo relacionado	9
2.3. Formulación Matemática	11
2.3.1. Descripción formal del problema	11
2.3.2. Modelo matemático del problema	12
2.4. Instancias del problema	15
2.4.1. Resultados computacionales	16
3. Distribución mono-producto: Algoritmo Propuesto	18

3.1. Greedy Randomized Adaptive Search Procedure	18
3.1.1. Etapa constructiva	19
3.1.2. Etapa de mejora	22
3.2. Reactive Greedy Randomized Adaptive Search Procedure	24
3.3. Análisis experimental	27
3.4. Modelo matemático con solución inicial	29
4. Problema de distribución multi-producto	32
4.1. Trabajo relacionado	33
4.2. Formulación Matemática	34
4.2.1. Modelo matemático del problema	34
4.3. Instancias del problema	37
4.3.1. Resultados computacionales	38
5. Distribución multi-producto: Algoritmo Propuesto	40
5.1. Heurística constructiva con aleatoriedad	41
5.2. Modelo de cobertura de conjuntos	44
5.2.1. Metodología de solución	46
5.3. Análisis experimental	47
5.3.1. Heurística Constructiva y modelo de cobertura de conjuntos .	47
5.3.2. Modelo del problema multi-producto y modelo de cobertura de conjuntos	51
6. Epílogo	53

6.1. Aportación Científica	53
6.2. Conclusiones	54
6.3. Áreas de oportunidad	55
A. Resultados	56
A.1. Experimentación problema mono-producto	57
A.2. Experimentación problema multi-producto	62
B. Notas complementarias	67
B.1. Problema de cobertura de conjuntos	67
B.2. Heurísticas y metaheurísticas	69

ÍNDICE DE FIGURAS

1.1. Instancia de un VRP (izquierda) y su solución(derecha)	2
3.1. Resultados Clase 3P25C: SPD vs RGRASP	27
3.2. Tiempo promedio: SPD vs SPD_0	30
5.1. Resultados Clase 10C: MPD-SC vs MPD-H	49
5.2. Resultados Clase 50C: MPD-SC vs MPD-H	49
5.3. Resultados Clase 100C: MPD-SC vs MPD-H	50
5.4. Resultados Clase 10C: MPD vs MPD-SC	51

ÍNDICE DE TABLAS

2.1. Trabajo relacionado al VRP mono-producto con restricciones adicionales	10
2.2. Resultados promedio reportados por CPLEX, modelo <i>SPD</i>	17
3.1. Resultados promedio <i>SPD</i> vs RGRASP	28
3.2. Resultados promedio <i>SPD</i> vs <i>SPD</i> ₀	30
4.1. Trabajo relacionado al VRP multi-producto con restricciones adicionales	33
4.2. Resultados promedio reportados por CPLEX, modelo <i>MPD</i>	39
5.1. Resultados promedio: <i>MPD</i> -H vs <i>MPD</i> -SC	48
5.2. Resultados promedio <i>MPD</i> vs <i>MPD</i> -SC	51
A.1. Experimentación: Problema mono-producto – 3P25C	58
A.2. Experimentación: Problema mono-producto – 3P50C	59
A.3. Experimentación: Problema mono-producto – 4P100C	60
A.4. Experimentación: Problema mono-producto – 5P150C	61
A.5. Experimentación: Problema multi-producto – 10C	63
A.6. Experimentación: Problema multi-producto – 50C	64

A.7. Experimentación: Problema multi-producto – 100C	65
A.8. Experimentación: Problema multi-producto – 200C	66

AGRADECIMIENTOS

Durante los dos años en que realicé mis estudios de maestría y generando como fruto, esta investigación conté con el apoyo de personas e instituciones a las cuales agradezco profundamente.

Agradezco al Consejo Nacional de Ciencia y Tecnología que me permitió realizar mis estudios de posgrado tanto nacionales como internacionales mediante la gratificación de una beca.

Doy gracias a los revisores de esta tesis, el Dr. Vincent André Lionel Boyer, el Dr. Leonardo Gabriel Hernández Landa y principalmente a la Dra. María Angélica Salazar Aguilar, quien fungió como supervisora y asesora de mi investigación, profesora por excelencia cuyos consejos y comentarios ayudaron a mejorar la calidad de mi trabajo.

Además quiero dirigir mi agradecimiento a la Universidad Autónoma de Nuevo León, a la Facultad de Ingeniería Mecánica y Eléctrica y al Posgrado en Ingeniería de Sistemas por haberme tomado como estudiante de maestría durante todos mis estudios. Así mismo agradezco a la Universidad LAVAL, especialmente al Dr. Leandro Callegari Coelho, por haberme recibido durante mi estancia de investigación en Quebec, Canadá.

Un agradecimiento para los maestros del Posgrado en Ingeniería de Sistemas, quienes me forjaron como estudiante e investigador con su conocimiento y habilidad de docencia.

RESUMEN

Roberto Cantu Funes.

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

Universidad Autónoma de Nuevo León.
Facultad de Ingeniería Mecánica y Eléctrica.

Título del estudio:

OPTIMIZACIÓN DE LA LOGÍSTICA DE TRANSPORTE Y DISTRIBUCIÓN DE PRODUCTOS

Número de páginas: 76.

OBJETIVOS Y MÉTODO DE ESTUDIO: Se tratará una variante del problema de ruteo de vehículos, conocido como VRP (Vehicle Routing Problem), enfocado en la minimización de costos de transporte entre plantas y centros de distribución. Las plantas cuentan con diferentes niveles de inventario y los centros de distribución demandan cantidades diferentes del producto. Para realizar este envío se cuenta con una flota de vehículos propia y una flota de vehículos externos, disponibles para rentar, cuyo costo depende de la relación origen-destino. Las flotas son heterogéneas ya que cuentan con dos tipos de vehículos con capacidades distintas; también se conoce el tiempo de traslado entre plantas y centros de distribución. Cada una de las plantas

puede satisfacer a un subconjunto de los centros de distribución, los centros de distribución cuentan con una fecha límite (due date) para recibir el producto, así como con una ventana de tiempo de servicio por día. Se desea asignar los vehículos a las plantas, de tal manera que las plantas cumplan con la demanda de los centros de distribución dentro de la fecha límite de los mismos, así como determinar los viajes que realizará cada camión y la cantidad de producto a transportar en cada viaje.

Se estudian dos variantes del problema: distribución mono-producto y distribución multi-producto. Para el caso mono-producto se propone un modelo de optimización entera mixta y para el caso multi-producto se proponen dos modelos, uno de programación entera mixta y otro basado en cobertura de conjuntos.

Para la solución de las variantes del problema de estudio, se proponen dos metodologías de solución. La primera es una heurística basada en RGRASP (Reactive Greedy Randomized Adaptive Search Procedure), la segunda metodología es una combinación entre un método constructivo y un modelo de cobertura de conjuntos. Las metodologías propuestas son evaluadas a través de experimentos computacionales sobre un gran número de instancias, así mismo los resultados obtenidos son comparados con aquellas instancias en las cuales el optimizador CPLEX 12.6 es capaz de encontrar soluciones óptimas.

CONTRIBUCIONES Y CONCLUSIONES: Se introdujo una variante del problema de ruteo, que no se había estudiado previamente en la literatura, el cual puede ser visto como una combinación de un problema de asignación y uno de ruteo de vehículos con múltiples restricciones. Como producto de este trabajo se presentan tres formulaciones matemáticas y dos metodologías de solución para los casos de distribución mono-producto y multi-producto. Ambas metodologías proveen soluciones de calidad en un tiempo computacional razonable.

Firma del asesor: _____

Asesor: Dra. María Angélica Salazar Aguilar

CAPÍTULO 1

INTRODUCCIÓN

La motivación de este trabajo proviene de la situación actual que enfrenta una compañía cervecera de la localidad. Esta compañía cuenta con un conjunto de plantas que se encargan de elaborar cerveza y almacenar temporalmente el producto terminado antes de enviarlo a los centros de distribución, los cuales son encargados de almacenar el producto para su distribución al cliente final.

Con los productos derivados de este trabajo, se podrá llevar a cabo la optimización de costos de transporte incurridos por la empresa. Además, el nivel de servicio a los clientes se verá favorecido dado que durante el proceso de solución se consideran las fechas límite de entrega, acordadas con los centros de distribución.

El problema de estudio corresponde a una variante del problema de ruteo de vehículos, comúnmente conocido en la literatura como *Vehicle Routing Problem* (VRP), con características adicionales del problema de asignación generalizada, conocido como *Generalized Assignment Problem* (GAP).

El problema de ruteo de vehículos es un problema ampliamente estudiado en la literatura, razón por lo cual existe una gran cantidad de variantes o extensiones al problema general, esto tratando de asemejar el entorno real en el que se aplica. En el VRP general, propuesto por Dantzig and Ramser (1959), se conocen a priori el conjunto de clientes a los que se debe satisfacer, así como la demanda asociada a cada cliente; para satisfacer la demanda se cuenta con un depósito que posee un inventario suficiente para cubrir toda la demanda acumulada de los clientes y

una flota de vehículos. El objetivo es asignar rutas a los vehículos de tal forma que las demandas sean satisfechas y las rutas empiecen y terminen en el depósito, minimizando el costo total de transporte.

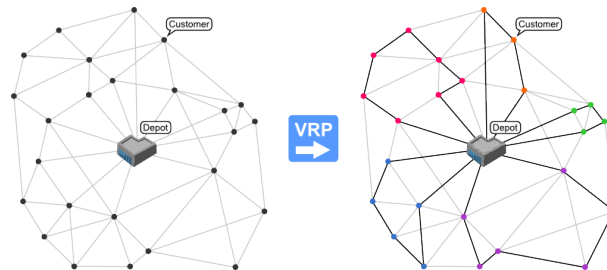


Figura 1.1: Instancia de un VRP (izquierda) y su solución(derecha)

Fuente: <http://neo.lcc.uma.es/vrp/vehicle-routing-problem/>

Dependiendo de las restricciones adicionales al VRP general, en la literatura se encuentran varios tipos de VRP, a continuación se presentan las variantes que son relevantes para el problema de estudio:

VRP CON FLOTA FIJA (FF): Esta variante del VRP cuenta con la característica de que los vehículos disponibles para el envío es limitada. Todas las demás restricciones del VRP permanecen iguales.

VRP CON FLOTA HETEROGÉNEA (HF): En esta variante del problema se agrega la característica de que los vehículos cuentan con capacidad heterogénea, esto quiere decir que no necesariamente todos los vehículos son iguales, así mismo la demanda acumulada de los clientes en una ruta debe ser menor que la capacidad del vehículo que realice dicha ruta.

VRP CON MÚLTIPLES DEPÓSITOS (MD): Como el nombre lo indica, esta variante cuenta con más de un depósito para realizar los envíos. Las rutas deben partir y finalizar en el mismo depósito.

VRP CON MÚLTIPLES VIAJES (MT): Este problema cuenta con una variante que permite a los vehículos realizar más de una ruta, las rutas deben respetar las

condiciones mencionadas en el VRP general y los clientes solo pueden ser visitados una sola vez.

VRP CON ENTREGA DIVIDIDA (SD): En esta variante, se permiten múltiples visitas a los clientes. Es principalmente útil cuando la demanda de los clientes es superior a la capacidad de los vehículos.

VRP CON MÚLTIPLES PERIODOS (P): En el VRP clásico el horizonte de planeación suele ser de 1 día (periodo). Esta variante alarga el horizonte de planeación a múltiples períodos, así mismo las rutas tienen la posibilidad de iniciar en un período diferente.

VRP CON MÚLTIPLES PERIODOS Y FECHA LÍMITE (D): Esta variante del problema es un complemento al VRP con múltiples periodos que se explicó previamente. En esta variante se establece una fecha límite para satisfacer al cliente, esta fecha es interpretada en su mayoría como un rango de periodos $[p_j^1, p_j^2]$ o un solo período L_j , el rango de periodos establece que el cliente j debe ser visitado a partir del período p_j^1 y hasta el período p_j^2 , en cambio el período L_j establece el período (día) límite para satisfacer al cliente j .

VRP CON VENTANAS DE TIEMPO (TW): Este variante del VRP ha sido muy estudiada por su alto grado de aplicación. En esta variante se cuenta con ventanas de tiempo en los clientes, esto quiere decir que cada cliente tendrá un horario $[a_j, b_j]$ en el que puede recibir el pedido, siendo a_j el inicio y b_j el fin de la ventana de tiempo.

VRP CON DURACIÓN DE RUTA (RD): Esta variante impone un límite máximo en la duración de las rutas, el uso que se ha dado a esta variante es para representar el tiempo máximo de trabajo de un vehículo o para indicar la distancia máxima que se puede recorrer con el tanque de gasolina.

VRP CON MÚLTIPLES PRODUCTOS: En el VRP común se distribuye un solo producto, en esta variante se distribuye más de un producto, por lo que los clientes cuentan ahora con demandas específicas para cada producto, así mismo las plantas tienen inventarios diferentes para cada producto. Se debe satisfacer la demanda de cada producto requerido por los clientes, manteniendo las restricciones del VRP clásico.

VRP CON CLIENTES POR RUTA (N): Esta variante impone un límite máximo en la cantidad de clientes que se pueden visitar en cada ruta..

VRP CON MÚLTIPLES VEHÍCULOS (MV): La variante extiende la cantidad de vehículos, permitiendo contar con más de un vehículo para realizar los envíos.

VRP CON ASIGNACIÓN DE VEHÍCULOS (A) : En esta variante se cuenta con una flota de vehículos, cada vehículo en la flota será asignado a un cliente y solo podrá realizar viajes para ese mismo cliente.

Esta última variante es propuesta en este trabajo, con el objetivo de satisfacer las características especiales del problema de estudio, en el cual se cuentan con 2 flotas de vehículos, la flota de vehículos privada y la flota de renta, donde la flota privada debe ser distribuida entre las plantas de producción para su uso exclusivo, de acuerdo a los requerimientos de la empresa.

El trabajo realizado parte de una investigación que toma en cuenta métodos de solución exactos y heurísticos, si bien es cierto que en la literatura se han realizado investigaciones previas que involucran distintas variantes del ruteo de vehículos, en este trabajo se incluyen todas aquellas características que se han presentado en otras investigaciones, en conjunto con características que otras investigaciones no presentan. Es importante destacar que no se encontró trabajo alguno que considere de manera simultánea el conjunto de restricciones del problema de estudio (como se muestra en el trabajo relacionado, descrito en las Secciones 2.2 y 4.1). Es por lo

anterior que con este trabajo se contribuye al avance del estado del arte en las áreas de transporte y logística.

1.1 METODOLOGÍA DE INVESTIGACIÓN

Para realizar este trabajo de tesis se siguió una metodología que consta de los siguientes pasos:

- Revisión del estado del arte de problemas de ruteo de vehículos y sus variantes, así como la incorporación del problema de asignación.
- Elaboración de formulaciones matemáticas para ambos casos del problema de estudio, distribución mono-producto y multi-producto.
- Diseño de un generador de instancias para el problema de estudio.
- Generación de instancias para el problema mono-producto y multi-producto.
- Validación y análisis de las formulaciones propuestas, mediante la solución de instancias de prueba con el optimizador CPLEX 12.6.
- Diseño e implementación de metaheurística para el problema mono-producto.
- Diseño e implementación de algoritmo constructivo para el problema con múltiples productos.
- Elaboración de una formulación matemática simplificada para el problema con múltiples productos.
- Implementación de un método de solución para el problema con múltiple producto utilizando las soluciones del algoritmo constructivo como entrada para la formulación matemática simplificada.
- Experimentación computacional utilizando los métodos de solución sobre las instancias generadas.

- Análisis de los resultados para determinar la eficiencia de los métodos de solución propuestos.
- Escritura de tesis.

1.2 ESTRUCTURA DE LA TESIS

Esta tesis se encuentra estructurada en capítulos, los mismos que detallan una o varias etapas de la metodología de investigación. El Capítulo 2 presenta la descripción formal del problema de distribución mono-producto, así como el trabajo relacionado, la formulación matemática y el análisis experimental de los resultados obtenidos con el optimizador CPLEX 12.6.

Posteriormente, en el Capítulo 3, se describe la metaheurística propuesta para la solución del problema de distribución mono-producto, presentando la metodología heurística GRASP y su variante reactiva. Los resultados obtenidos por dicha metaheurística son mostrados en el mismo capítulo, así como el análisis de su desempeño, utilizando como base los resultados obtenidos con CPLEX 12.6.

En el Capítulo 4 se aborda el problema de distribución multi-producto, presentando el trabajo relacionado, formulación matemática y análisis experimental. En el Capítulo 5 detalla la metodología de solución utilizada para el problema multi-producto, presentando así una nueva formulación matemática simplificada de dicho problema aunado a un algoritmo constructivo. La experimentación correspondiente, así como en análisis de la calidad de las soluciones se encuentran incluidas en el mismo. Finalmente el Capítulo 6 presenta las conclusiones del trabajo realizado.

CAPÍTULO 2

PROBLEMA DE DISTRIBUCIÓN MONO-PRODUCTO

El *problema de ruteo de vehículos* es un problema ampliamente estudiado en la literatura, desde que fue propuesto por Dantzig and Ramser (1959), se han introducido diferentes métodos para su solución que abarcan desde métodos heurísticos hasta metodologías exactas, así mismo se ha demostrado una amplia gama de aplicaciones reales. Sin embargo, no se conoce trabajo alguno cuya investigación se haya enfocado en la versión que se presenta en esta tesis, pero si se lograron identificar trabajos que comparten características del problema que estamos tratando.

Empezaremos estableciendo la descripción formal del problema para poder identificar las características que se comparten con otros trabajos en la literatura. El problema de ruteo de vehículos que se presenta también incluye características del problema de asignación, a dicho problema se hará referencia como VRP con asignación de vehículos (A).

2.1 DESCRIPCIÓN DEL PROBLEMA

El problema de estudio proviene de la situación que presenta una empresa cervecera de la localidad, la cual cuenta con diferentes plantas de producción y una vez que se tiene producto terminado, la compañía lo transporta a los diferentes centros de distribución. En esta tesis nos enfocaremos a resolver una problemática

que, desde el punto de vista de investigación de operaciones, puede ser vista como una combinación de un problema de asignación y uno de ruteo de vehículos con múltiples restricciones adicionales.

La empresa dispone de una flota de vehículos privada, la cual debe ser distribuida entre las diferentes plantas. Una vez que un vehículo es asignado a una planta, este debe realizar envíos exclusivamente para esa planta, durante el período de planeación. Las plantas cuentan con un inventario de producto terminado y los centros de distribución cuentan con una demanda de producto a satisfacer. Es común recurrir a la contratación de transporte externo, debido a la demanda de los centros de distribución y a las condiciones que estos imponen. Cada centro de distribución requiere ser abastecido en un tiempo no mayor a una fecha límite (due date) dentro del período de planeación, además, cuenta con una ventana de tiempo de recepción de producto por día.

La cerveza de la compañía es distribuida en envase retornable, los centros de distribución son los encargados de almacenar estos envases vacíos además del producto a distribuir entre los clientes. La compañía establece que los viajes de los vehículos deben partir de una planta, visitar solamente un centro de distribución y volver a la planta, esto debido a que los viajes deben ser capaces de traer el envase vacío de regreso a la planta de ser necesario, así mismo los vehículos tanto propios como rentados tienen un costo por viaje dependiente de la planta de la cual se parte y el centro de distribución a visitar.

Cada una de las plantas puede satisfacer a un subconjunto de los centros de distribución. Las flotas de vehículos son heterogéneas ya que cuentan con dos o más tipos de camiones con capacidades distintas. Dado un período de planeación, el problema consiste en asignar el transporte propio a las plantas, y determinar qué planta, cuándo y qué cantidad de producto va a enviar a cada centro de distribución, y además, determinar que vehículos deben contratarse para satisfacer las restricciones de capacidad, demanda, ventanas de tiempo y fechas límite, minimizando el costo total.

2.2 TRABAJO RELACIONADO

En el caso de problemas de ruteo de vehículos mono-producto se han realizado diversas investigaciones utilizando métodos de solución tanto exactos como heurísticos. Existen investigaciones que hacen referencia a una o más extensiones utilizando metodologías de solución exactas como branch-and-price (Azi et al. (2010)).

Debido a la complejidad del problema se han propuesto una gran cantidad de métodos heurísticos y metaheurísticos para resolver el VRP con una o más extensiones, tales como la búsqueda tabú (Jiang et al. (2014)), búsqueda por entornos variables (Levy et al. (2014)), algoritmos genéticos híbridos (Vidal et al. (2012)), algoritmos de cúmulo de partículas (Belmecheri et al. (2013)), búsqueda dispersa (Qu and Bard (2013)), RGRASP (Boudia et al. (2007)), procedimiento de búsqueda miope aleatorio con ALNS (Qu and Bard (2013)) y heurísticos constructivos (Afshar-Nadjafi and Afshar-Nadjafi (2014)).

El trabajo de Bettinelli et al. (2011) es una de las investigaciones en la literatura que involucra más extensiones, abordando un problema de ruteo de vehículos con flota fija (FF), flota heterogénea (HF), múltiples depósitos (MD), ventanas de tiempo (TW) y duración de ruta (RD), utilizando como método de solución un branch-and-cut-and-price. Posteriormente en Bettinelli et al. (2014), los autores realizan un análisis del impacto producido por el tipo de ventanas de tiempo (suaves o duras), utilizando un algoritmo de branch-and-price.

A continuación se presenta una tabla comparativa de los trabajos que se han realizado, es importante mencionar que el trabajo que realizamos en esta tesis cuenta con todas las características que se incluyen en la Tabla 2.1.

Referencia	FF	HF	MD	MT	SD	P	D	TW	A	N	MV	RD
Bettinelli et al. (2011)	✓	✓	✓					✓			✓	✓
Ceschia et al. (2011)		✓				✓		✓			✓	✓
Bettinelli et al. (2014)	✓	✓	✓					✓			✓	
Afshar-Nadjafi and Afshar-Nadjafi (2014)	✓	✓	✓					✓			✓	
Lai et al. (2013)		✓			✓					✓	✓	
Seixas and Mendes (2013)		✓		✓				✓			✓	✓
Xu et al. (2012)		✓	✓					✓			✓	✓
Dondo and Cerdá (2007)		✓	✓					✓			✓	✓
Azi et al. (2010)	✓				✓			✓				✓
Jiang et al. (2014)	✓	✓						✓			✓	
Belmecheri et al. (2013)	✓	✓						✓			✓	
Tavakkoli-Moghaddam et al. (2007)		✓			✓						✓	
Dayarian et al. (2015)	✓	✓	✓					✓			✓	
Xu and Jiang (2014)	✓	✓	✓					✓			✓	
Vidal et al. (2014)		✓	✓			✓					✓	
Salhi et al. (2014)		✓	✓								✓	✓
Levy et al. (2014)		✓	✓								✓	
Brandao and Mercer (1997)		✓									✓	✓
Jair et al. (2013)		✓						✓			✓	
Yoshizaki et al. (2009)		✓			✓			✓			✓	
Yao et al. (2014)	✓	✓									✓	
Wang et al. (2014)	✓	✓									✓	
Qu and Bard (2013)	✓	✓									✓	
Liu (2013)	✓	✓									✓	
Kwon et al. (2013)	✓	✓									✓	
Bolduc et al. (2008)	✓	✓									✓	
Juan et al. (2014)		✓									✓	✓
Kritikos and Ioannou (2013)		✓						✓			✓	
Vidal et al. (2012)			✓									✓
Boudia et al. (2007)						✓						✓
Gonçalves et al. (2005)	✓					✓						✓

Tabla 2.1: Trabajo relacionado al VRP mono-producto con restricciones adicionales

Podemos observar que las investigaciones previas que se encontraron en la literatura no logran abarcar todas las características que presenta el problema que abordamos en esta tesis, por lo que concluimos que este trabajo está enriqueciendo el estado del arte de los problemas de ruteo de vehículos.

2.3 FORMULACIÓN MATEMÁTICA

A continuación se propone un modelo de programación lineal entera mixta, que aborda el problema de estudio en el caso mono-producto.

Primero se introducirá en la Sección 2.3.1 la notación matemática general que será utilizada a lo largo de la investigación, seguido por la formulación matemática explicada a detalle en la Sección 2.3.2.

2.3.1 DESCRIPCIÓN FORMAL DEL PROBLEMA

Dado un conjunto de plantas I y un conjunto de centros de distribución J , el problema que se plantea consiste en asignar y diseñar rutas para la flota de vehículos K tales que cada ruta comience y termine en la misma planta. Teniendo como objetivo minimizar el costo de transporte, satisfaciendo la demanda para todo cliente $j \in J$ y no excediendo el inventario en cada planta $i \in I$.

El horizonte de planeación es extendido a P días (periodos), en cada período $p \in P$ los vehículos están disponibles para realizar una ruta, cada ruta esta conformada por una cantidad máxima de viajes $|V|$, siendo V el conjunto de viajes, que se pueden realizar por ruta, el tiempo total de la ruta no debe exceder T , que representa el tiempo máximo de ruta. Un viaje inicia y termina en la misma planta y visita solamente a un centro de distribución, esto debido a políticas de la compañía, explicadas anteriormente.

Se cuenta con una flota privada de vehículos K' , cada uno de los vehículos en esta flota deben ser asignados a una planta y una vez asignados solo pueden realizar

viajes en la planta que les fue asignada durante todo el período de planeación; la flota privada de vehículos K' pertenece al conjunto de vehículos K , de tal forma que $K' \subset K$. Así mismo cada vehículo k cuenta con una capacidad de producto q_k que puede almacenar para su traslado.

Cada planta sólo puede servir a un subconjunto de centros de distribución, para su representación se cuenta con el conjunto A de tal forma que si (i, j) pertenece al conjunto A , se establece que la planta i puede servir al centro de distribución j .

Para la recepción de producto en los centros de distribución, se cuenta con dos restricciones principales. La primera es la existencia de una fecha límite de entrega (due-date) L_j , la misma, hace referencia al último periodo dentro del cual el cliente j puede recibir producto, de tal manera que $L_j \leq P$. La segunda restricción se debe a que los centros de distribución cuentan con ventanas de tiempo $[a_j, b_j]$, en el cual el cliente j puede recibir el producto cada período.

2.3.2 MODELO MATEMÁTICO DEL PROBLEMA

El modelo de optimización que representa el problema de distribución mono-producto se describe a continuación.

Conjuntos

I	Plantas de Producción.
J	Centros de Distribución.
N	$I \cup J$.
V	Viajes.
K	Vehículos.
K'	Flota de vehículos privada, $K' \subset K$.
P	Periodos.
A	Arcos.

Parámetros

T	Tiempo máximo de la ruta por período.
Q_i	Inventario de producto en la planta $i \in I$.
D_j	Demanda de producto en el centro de distribución $j \in J$.
L_j	Fecha límite (due-date) del centro de distribución $j \in J$.
a_j	Inicio de la ventana de tiempo del centro de distribución $j \in J$.
b_j	Fin de la ventana de tiempo del centro de distribución $j \in J$.
q_k	Capacidad del vehículo $k \in K$.
$t_{i,j}$	Tiempo de traslado entre los nodos (i,j) , $i, j \in N \mid (i, j) \in A$.
$c_{i,j,k}$	Costo de viaje del vehículo k desde la planta i al centro de distribución j , $k \in K$, $(i, j) \in A$.

Variables de decisión

$$x_{ijvk}^p = \begin{cases} 1 & \text{Si la planta } i \text{ envía producto al centro de distribución } j \\ & \text{en el viaje } v \text{ del vehículo } k, \text{ en el período } p, \\ 0 & \text{en otro caso.} \end{cases}$$

$$u_{ik} = \begin{cases} 1 & \text{Si el vehículo } k \in K' \\ & \text{es asignado a la planta } i \in I, \\ 0 & \text{en otro caso.} \end{cases}$$

$$s_{vk}^p \quad \text{Tiempo en que el vehículo } k \text{ parte de la planta para realizar el viaje } v \\ \text{del período } p.$$

$$(SPD) \text{Min} \sum_{(i,j) \in A} \sum_{v \in V} \sum_{k \in K} \sum_{p \in P} (c_{ijk} x_{ijvk}^p) \quad (2.1)$$

sujeito a:

$$\sum_{(i,j) \in A} \sum_{v \in V} \sum_{k \in K} \sum_{\substack{p \in P, \\ p \leq L_j}} (q_k x_{ijvk}^p) \geq D_j \quad \forall j \in J \quad (2.2)$$

$$\sum_{(i,j) \in A} \sum_{v \in V} \sum_{k \in K} \sum_{p \in P} (q_k x_{ijvk}^p) \leq Q_i \quad \forall i \in I \quad (2.3)$$

$$s_{vk}^p + \sum_{(i,j) \in A} x_{ijvk}^p (t_{ij} + t_{ji}) \leq s_{v+1,k}^p \quad \forall v \in V, k \in K, p \in P \mid v+1 \in V \quad (2.4)$$

$$s_{vk}^p + \sum_{(i,j) \in A} (t_{ij} x_{ijvk}^p) \geq \sum_{(i,j) \in A} (a_j x_{ijvk}^p) \quad \forall v \in V, k \in K, p \in P \quad (2.5)$$

$$s_{vk}^p + \sum_{(i,j) \in A} (t_{ij} x_{ijvk}^p) \leq \sum_{(i,j) \in A} (b_j x_{ijvk}^p) \quad \forall v \in V, k \in K, p \in P \quad (2.6)$$

$$x_{ijvk}^p \leq u_{ik} \quad \forall i \in I, j \in J, v \in V, k \in K', p \in P \quad (2.7)$$

$$\sum_{i \in I} u_{ik} = 1 \quad \forall k \in K' \quad (2.8)$$

$$\sum_{(i,j) \in A} x_{ijvk}^p \leq 1 \quad \forall v \in V, k \in K, p \in P \quad (2.9)$$

$$s_{vk}^p + \sum_{(i,j) \in A} x_{ijvk}^p (t_{ij} + t_{ji}) \leq T \quad \forall v \in V, k \in K, p \in P \quad (2.10)$$

La función objetivo (2.1) busca minimizar el costo de los viajes. La expresión (2.2) garantiza que la demanda de los clientes se satisfaga. Las restricciones (2.3) establecen que el inventario de cada planta no sea excedido. El conjunto de inecuaciones (2.4) asegura que los vehículos no pueden empezar un nuevo viaje sin haber concluido el anterior. El conjunto de restricciones (2.5) – (2.6) se encargan de garantizar que las ventanas de tiempo sean respetadas para todos los clientes. Las inecuaciones (2.7) – (2.8) son responsables de la asignación de la flota privada, asegurando que solo se realicen viajes desde la planta asignada y que cada vehículo sea asignado exclusivamente a una planta. La restricción (2.9) limita a los vehículos a visitar solamente un centro de distribución en cada viaje, por último la restricción (2.10) garantiza que todos los viajes finalizan antes de que se acabe la duración del período.

2.4 INSTANCIAS DEL PROBLEMA

Para poder determinar el alcance y desempeño del modelo matemático descrito anteriormente fue necesario realizar experimentación computacional sobre un conjunto de instancias. Para el problema mono-producto se generaron un total de 120 instancias divididas en cuatro clases.

Las clases se encuentran definidas por los siguientes parámetros:

- P – plantas de producción
- C – centros de distribución

Explicado lo anterior se puede deducir que la clase $3P25C$ hace referencia al conjunto de instancias que cuentan con 3 plantas de producción y 25 centros de distribución. Así mismo las subclases realizan un detalle más específico de la instancia. Las subclases se encuentran definidas únicamente por la longitud de las ventanas de tiempo de los clientes(TW) y para la creación de instancias se tomaron solo las siguientes longitudes posibles:

- 3TW
- 5TW
- 8TW

Un ejemplo de una ventana de tiempo 3TW puede ser $[7,10]$ siendo 7 el inicio de la ventana de tiempo y 10 final de la misma, por lo que se puede observar que se cuentan con tres unidades de diferencia entre el tiempo en que se inicia la ventana de tiempo hasta que se concluye la misma.

Para este problema se decidió generar 10 instancias por subclase, así mismo cada clase tiene 3 subclases posibles como observamos anteriormente por lo que se

contaron con cuatro clases y doce subclases. En las instancias de este problema mono-producto la flota privada K' se fijó a un tercio de la flota total K .

Para concluir se presenta el ejemplo de la instancia $3P25C-3TW-45K2$, en la cual podríamos deducir, retomando lo explicado previamente, que pertenece a la clase $3P25C$ compuesta por tres plantas y veinticinco centros de distribución, y a su vez, pertenece a la subclase $3P25C-3TW$ especificando que las ventanas de tiempo asignadas a los clientes son de 3 unidades de longitud. La última parte del nombre de la instancia $45K2$, hace referencia a la cantidad de vehículos $|K|$ y al número de instancia de la subclase, por lo que se cuentan con 45 vehículos en la flota total, de los cuales 15 pertenecen a la flota propia, y la instancia es la segunda generada de la subclase $3P25C-3TW$.

2.4.1 RESULTADOS COMPUTACIONALES

Para su experimentación se codificó el modelo matemático, explicado en la Sección 2.3.2, utilizando CPLEX 12.6 y C++. La experimentación del problema mono-producto se realizó sobre una computadora con un CPU *Intel Xeon E5-2687W @ 3.10GHz* y 64GB de memoria RAM. Para la experimentación del modelo matemático se fijaron los siguientes parámetros en CPLEX:

- Tiempo Límite: 1 hora
- Hilos utilizados: 1 hilo
- GAP: 1%

Se cuentan con cuatro clases en las que se probó el modelo matemático las cuales son $3P25C$, $3P50C$, $4P100C$ y $5P150C$. Los resultados de la experimentación se muestran a detalle en el apéndice A.

Clase	Soluciones	SPD	
		GAP (%)	Tiempo (s)
3P25C	29/30	0.45	511
3P50C	0/30	–	>3600
4P100C	0/30	–	>3600
5P150C	0/30	–	>3600

Tabla 2.2: Resultados promedio reportados por CPLEX, modelo *SPD*

La columna uno indica la clase a la que pertenecen las instancias, la columna 2 muestra la cantidad de instancias que fueron resueltas del total de instancias de dicha clase (resueltas/totales), las ultimas columnas hacen referencia al GAP y tiempo promedio que reporto el optimizador CPLEX.

Podemos ver que CPLEX logró resolver casi todas las instancias de la clase *3P25C*, dejando sin resolver solamente una de las treinta con un GAP promedio de 0.459% en un tiempo promedio de 511 segundos.

El análisis de las clases restantes muestra que para CPLEX 12.6 fue imposible resolver las instancias en el tiempo límite de 1 hora y aunado a eso no se encontró cota o solución factible para todas las instancias de prueba. Los resultados muestran que la formulación matemática por si misma en casos con más de cincuenta centros de distribución es difícil de resolver. Es por esta razón que se optó por diseñar una heurística que permita resolver instancias más grandes.

CAPÍTULO 3

DISTRIBUCIÓN MONO-PRODUCTO: ALGORITMO PROPUESTO

Para la solución del problema de distribución de un solo producto se decidió utilizar una metaheurística. Una heurística es un método que consiste en un enfoque para resolver una problemática que no garantiza una respuesta óptima o perfecta pero si suficiente para objetivos inmediatos. En cambio, la metaheurística utiliza procedimientos heurísticos fijando parámetros que el usuario especifique de manera que sea eficiente para el problema de estudio. Se puede encontrar una descripción más detallada en el Apéndice B.

La metaheurística diseñada para resolver el problema mono-producto es un RGRASP (reactive greedy randomized adaptive search procedure). El GRASP fue introducido por Feo and Bard (1989) y consiste en un número fijo de iteraciones, donde en cada iteración se genera una solución al problema de estudio. La mejor solución de todas las iteraciones del GRASP es guardada como solución incumbente.

3.1 GREEDY RANDOMIZED ADAPTIVE SEARCH PROCEDURE

La estructura principal de las iteraciones dentro del GRASP fue explicada años más tarde en la publicación realizada por Feo and Resende (1995), misma que

explicaba que las iteraciones dentro del GRASP consisten en dos etapas.

La primera etapa, también conocida como la etapa constructiva, realiza la construcción de una solución a través de una función adaptativa con aleatoriedad. La segunda etapa o la etapa de mejora se encarga de aplicar algún procedimiento de búsqueda local a la solución entregada por la etapa constructiva buscando así una posible mejora.

Se seleccionó la metodología de solución GRASP, debido a la complejidad del problema para generar soluciones factibles; tomando ventaja de la simplicidad de la estructura del GRASP que nos permite el diseño de una etapa constructiva robusta haciendo posible la generación de soluciones factibles en cada iteración.

3.1.1 ETAPA CONSTRUCTIVA

La etapa constructiva consiste de un método aleatorio con enfoque “greedy” cuyo trabajo consiste en crear una solución completa agregando pequeños elementos a la misma. Explicaremos esta etapa constructiva en tres pasos. En el paso uno creamos una lista de candidatos (CL), los candidatos son evaluados con base en una función con enfoque “greedy” (e), estos candidatos son elementos que pueden ser agregados a la solución.

El paso dos consiste en filtrar, en base a un parámetro α , la lista CL hacia una lista restringida de candidatos (RCL), de tal manera que RCL se define de la siguiente forma $RCL = \{j \in CL \mid e(j) \leq e_{min} + \alpha(e_{max} - e_{min})\}$, tomando en cuenta que $e(j)$ es la evaluación dada por la función con enfoque “greedy” (e) al elemento j en la lista de candidatos CL y a su vez e_{min} y e_{max} son los valores mínimo y máximo, diferentes de 0, evaluados por la función. El parámetro α nos permite restringir la lista RCL ; con un $\alpha = 1$, la lista restringida de candidatos (RCL) tendrá todos los elementos de la lista CL y con un $\alpha = 0$ la lista restringida de candidatos solamente va a contener los elementos de $j \in CL$ tal que $e(j) = e_{min}$.

El paso tres consiste simplemente en seleccionar un elemento de la lista res-

tringida de candidatos (RCL) e insertarlo a la solución. La selección debe agregar aleatoriedad al método, de manera que si se ejecuta el método constructivo múltiples veces, podría generar soluciones diversas, esto se logra a través de una función de probabilidad ρ .

En el Algoritmo 1 se utilizara la notación explicada en la Sección 2.3.2, así como las siguientes definiciones.

Sea V_j el conjunto de viajes factibles al centro de distribución j y $V = \{V_1, V_2, \dots, V_{|J|}\}$ el conjunto de todos los viajes.

Para cada viaje $v \in V$, se definen los siguientes parámetros:

- $c_v :=$ costo de envío al realizar el viaje v .
- $\theta_v :=$ vehículo que asociado al viaje v .
- $w_v := \frac{c_v}{q\theta_v}$.
- $\gamma_v :=$ Centro de distribución al que se dirige el viaje v .
- $\lambda_v :=$ Planta de producción de la que parte el viaje v .

En nuestro *GRASP* los elementos de la lista de candidatos (CL) son viajes y la función de probabilidad se muestran a continuación.

Para cada viaje $v \in V$, se utilizan las siguientes definiciones:

- $e(v) :=$ Cantidad de arcos incidentes al centro de distribución γ_j .
- $\rho(v) := \frac{\frac{|P|}{L_{\gamma_v}}}{\sum_{v \in RCL} \frac{|P|}{L_{\gamma_v}}}$.

El procedimiento detallado de la etapa constructiva se encuentran en el Algoritmo 1.

Algoritmo 1 Método Constructivo

Entrada: parámetro de calidad α

Salida: sol : solución a la instancia del problema.

- 1: Sea V_j el conjunto de viajes factibles al centro de distribución j
 - 2: $V \leftarrow \{V_1, V_2, \dots, V_{|J|}\}$
 - 3: $sol \leftarrow \emptyset$
 - 4: **mientras** exista $D_j > 0$, $j \in J$ **hacer**
 - 5: Sea $v'_j \leftarrow \operatorname{argmin}_{v \in V_j} \{w_v : e(v) > 0\}$
 - 6: $CL \leftarrow \bigcup_{j \in J} v'_j$
 - 7: $RCL \leftarrow \{v \in CL \mid e(v) \leq e_{min} + \alpha(e_{max} - e_{min})\}$
 - 8: **si** $RCL = \emptyset$, **entonces** **Retorna** Solución No Factible
 - 9: **fin si**
 - 10: Seleccionar un viaje aleatorio $v^* \in RCL$ considerando la función de probabilidad ρ
 - 11: **si** $\theta_{v^*} \in K'$ y el vehículo θ_{v^*} no se ha asignado, **entonces**
 - 12: Vehículo θ_{v^*} es asignado a la planta λ_{v^*}
 - 13: Eliminar cada viaje $v \in V$ tal que $\theta_v = \theta_{v^*}$ y $\lambda_v \neq \lambda_{v^*}$
 - 14: **fin si**
 - 15: $Q_{\lambda_{v^*}} \leftarrow Q_{\lambda_{v^*}} - \min\{q_{\theta_{v^*}}, D_{\gamma_{v^*}}\}$
 - 16: $D_{\gamma_{v^*}} \leftarrow D_{\gamma_{v^*}} - q_{\theta_{v^*}}$
 - 17: **si** $Q_{\lambda_{v^*}} \leq 0$, **entonces**
 - 18: Eliminar cada viaje $v \in V$ tal que $\lambda_v = \lambda_{v^*}$
 - 19: **fin si**
 - 20: **si** $D_{\gamma_{v^*}} \leq 0$, **entonces**
 - 21: Eliminar cada viaje $v \in V$ tal que $\gamma_v = \gamma_{v^*}$
 - 22: **fin si**
 - 23: Actualizar aristas activas
 - 24: Eliminar $v \in V$ tal que v no es factible
 - 25: **fin mientras**
 - 26: **Retorna** sol
-

3.1.2 ETAPA DE MEJORA

Como se explicaba anteriormente la etapa de mejora de GRASP consta normalmente de un algoritmo de búsqueda local. En nuestro caso se utilizó un algoritmo de búsqueda local que consiste en dos pasos.

En el paso uno se intenta mejorar la solución realizando iteraciones de la siguiente forma: seleccionando una planta de producción (i^*) y un vehículo (k^*) aleatorio. Posteriormente, se eliminan todos los viajes asociados al vehículo k^* desde la planta i^* y se reconstruye la solución evitando la asignación del vehículo k^* a la planta i^* . Este proceso termina una vez que se alcanzan cinco iteraciones consecutivas sin mejora.

El paso dos es un complemento del método constructivo, tomando en cuenta que la demanda de algún cliente puede terminar con un valor negativo. Tener una demanda negativa indica la cantidad de unidades que se cargarán de menos en el último vehículo que se dirige a ese cliente, por ejemplo si el cliente 1 tiene una demanda de -2 quiere decir que el vehículo que realice el último viaje al cliente 1 viajará con 2 unidades menos de capacidad, para así cumplir con la demanda exacta del cliente.

Tomando ventaja de la situación anterior, el paso dos identifica dichos casos con demanda negativa y en cada caso elimina el viaje de algún vehículo con el costo más alto y se intenta insertar, en su lugar, un viaje de un vehículo que se acomode mejor con la demanda del cliente, con el supuesto de que un viaje que realiza un vehículo con una capacidad mayor tendrá un costo más alto que un viaje realizado por un vehículo con capacidad menor. El método creado para la etapa de mejora de nuestro GRASP se encuentra en el Algoritmo 2.

Algoritmo 2 Método de mejora

Entrada: SOL : Solución inicial.

1: parámetro de calidad α

Salida: $BSOL$ Solución modificada {Paso 1}

2: $it \leftarrow 0$

3: $BSOL \leftarrow SOL$

4: **mientras** $it < 5$ **hacer**

5: $TSOL \leftarrow SOL$

6: $i^* \leftarrow$ planta aleatoria $i^* \in I$

7: $k^* \leftarrow$ vehículo aleatorio $k^* \in K$

8: Elimina cada viaje $v \in TSOL$ tal que $i^* = \lambda_v$ & $k^* = \theta_v$

9: **si** $k^* \in K'$, **entonces**

10: Se restringe la asignación del vehículo k^* a la planta i^*

11: **fin si**

12: Actualizar en $TSOL$ demandas, inventarios y arcos disponibles

13: $TSOL \leftarrow$ Método constructivo($TSOL, \alpha$)

14: $it \leftarrow it + 1$

15: **si** $TSOL$ es factible y $(BSOL = SOL \parallel f(TSOL) < f(BSOL))$, **entonces**

16: $BSOL \leftarrow TSOL$

17: $it \leftarrow 0$

18: **fin si**

19: **fin mientras**

20: {Paso 2}

21: **para** cada $j \in J$ tal que $D_j < 0$ **hacer**

22: Elimina de $BSOL$ el viaje v con el mayor c_v tal que $\gamma_v = j$

23: Inserta de $BSOL$ el viaje v con el menor c_v tal que $q_{\theta_v} \geq D_j$

24: **fin para**

25: **Retorna** $BSOL$

3.2 REACTIVE GREEDY RANDOMIZED ADAPTIVE SEARCH PROCEDURE

En el procedimiento de búsqueda adaptativa con enfoque “greedy” (GRASP) el parámetro de calidad α juega un papel decisivo ya que es el encargado de filtrar los elementos a la lista restringida (*RCL*). Por lo que es común encontrar que un valor del parámetro de calidad α , digamos α_1 , ofrezca mejor resultado para el conjunto de instancias σ_1 pero genere peores valores objetivos para instancias del conjunto σ_2 . Debido a esto, el seleccionar un parámetro de calidad que se ajuste correctamente a todas las instancias había sido tarea difícil del tomador de decisiones. Fue hasta años después de la creación de la metodología GRASP que se pudo dar solución a esta problemática.

El RGRASP fue propuesto por Prais and Ribeiro (2000) y su principal ventaja es que cuenta con la capacidad de ajustar el valor del parámetro de calidad α de entre un pequeño conjunto de valores permitidos, tomando así el que genere mejor solución para la instancia dada.

Sea Λ el número de valores α y $A = \{a_1, a_2, \dots, a_\Lambda\}$ el conjunto de valores permitidos. El RGRASP encuentra el mejor valor de α seleccionando un valor de α aleatorio en cada iteración del GRASP y teniendo en cuenta que ρ_i es la probabilidad de elegir el valor $\alpha = a_i$. Las probabilidades ρ son inicialmente fijas en $\frac{1}{\Lambda}$ y cambian a medida que se obtienen nuevas soluciones, favoreciendo a los valores de α que generen mejor valor en la función objetivo. De esta manera se consigue ajustar el valor del parámetro de calidad.

Para la re-actividad del nuestro RGRASP se toma en cuenta lo siguiente: Sea n_i la cantidad de soluciones construidas con cada valor a_i y sum_i la suma de los valores objetivos de las soluciones. Así mismo $f(S^*)$ sea el valor objetivo de la mejor solución encontrada y $f(S^-)$ sea el valor objetivo de la peor solución generada hasta el momento. Las probabilidades de seleccionar un valor de α se obtienen calculando

el costo promedio de las soluciones encontradas para cada valor α_i : $\bar{f}_i = \frac{sum_i}{n_i}$, si $n_i = 0$ entonces $\bar{f}_i = f(S^-)$ de tal manera que la probabilidad está dada por:

$$\rho_i := \frac{q_i}{\sum_{j=0}^{|A|} q_j}, \quad q_i := \frac{f(S^*)}{\bar{f}_\alpha}$$

Así mismo las probabilidades de seleccionar un valor de α son cambiadas cada 200 iteraciones, de esta forma nuestro RGRASP se compone de los pasos mostrados en el Algoritmo 3.

Algoritmo 3 RGRASP**Entrada:** it : Número de iteraciones del GRASP.**Salida:** S^*

- 1: $S^* \leftarrow \emptyset$
- 2: $S^- \leftarrow \emptyset$
- 3: $A \leftarrow \{0, 0.2, 0.4, 0.5, 0.6, 0.8, 1\}$
- 4: $q_\alpha \leftarrow 1 \forall \alpha \in A$
- 5: $sum_\alpha \leftarrow 0 \forall \alpha \in A$
- 6: $n_\alpha \leftarrow 0 \forall \alpha \in A$
- 7: $\bar{f}_\alpha \leftarrow 0 \forall \alpha \in A$
- 8: $i \leftarrow 0$.
- 9: **mientras** $i \leq it$ **hacer**
- 10: $\hat{\alpha} \leftarrow$ valor aleatorio $a_i \in A$ siguiendo la función de probabilidad ρ_i
- 11: $SOL \leftarrow$ Método constructivo($\hat{\alpha}$)
- 12: **si** SOL es factible , **entonces**
- 13: $SOL \leftarrow$ Método de mejora($SOL, \hat{\alpha}$)
- 14: **si** $S^* = \emptyset \parallel f(SOL) < f(S^*)$, **entonces**
- 15: $S^* \leftarrow SOL$
- 16: **fin si**
- 17: **si** $S^- = \emptyset \parallel f(SOL) > f(S^-)$, **entonces**
- 18: $S^- \leftarrow SOL$
- 19: **fin si**
- 20: $sum_{\hat{\alpha}} \leftarrow sum_{\hat{\alpha}} + f(SOL)$
- 21: $n_{\hat{\alpha}} \leftarrow n_{\hat{\alpha}} + 1$
- 22: $i \leftarrow i + 1$
- 23: **fin si**
- 24: **si** i módulo 200 = 0 , **entonces**
- 25: **para** cada $\alpha \in A$ **hacer**
- 26: **si** $n_\alpha = 0$, **entonces**
- 27: $\bar{f}_\alpha \leftarrow f(S^-)$
- 28: **si no**
- 29: $\bar{f}_\alpha \leftarrow \frac{sum_\alpha}{n_\alpha}$
- 30: **fin si**
- 31: $q_\alpha \leftarrow \frac{f(S^*)}{\bar{f}_\alpha}$
- 32: $sum_\alpha \leftarrow 0$
- 33: $n_\alpha \leftarrow 0$
- 34: **fin para**
- 35: **fin si**
- 36: **fin mientras**
- 37: **Retorna** S^*

3.3 ANÁLISIS EXPERIMENTAL

La metodología de solución previamente explicada fue evaluada con el mismo conjunto de instancias que la formulación matemática vista en 2.3.2.

El RGRASP fue codificado en C++ y se realizó la experimentación sobre una computadora con un CPU *Intel Xeon E5-2687W @ 3.10GHz* y 64GB de memoria RAM.

Se cuentan con cuatro clases de instancias *3P25C*, *3P50C*, *4P100C* y *5P150C*. Para la experimentación se fijó el número de iteraciones (*it*) en 2000. Los resultados experimentales de las clases se encuentran detallados en el apéndice A. En la Figura 3.1 se muestra una gráfica comparativa entre las soluciones que obtuvo CPLEX 12.6 (Sección 2.4.1) y se compara con la solución que brinda la metaheurística propuesta, RGRASP. Además, en la Tabla 3.1 se muestra una síntesis de los resultados experimentales.

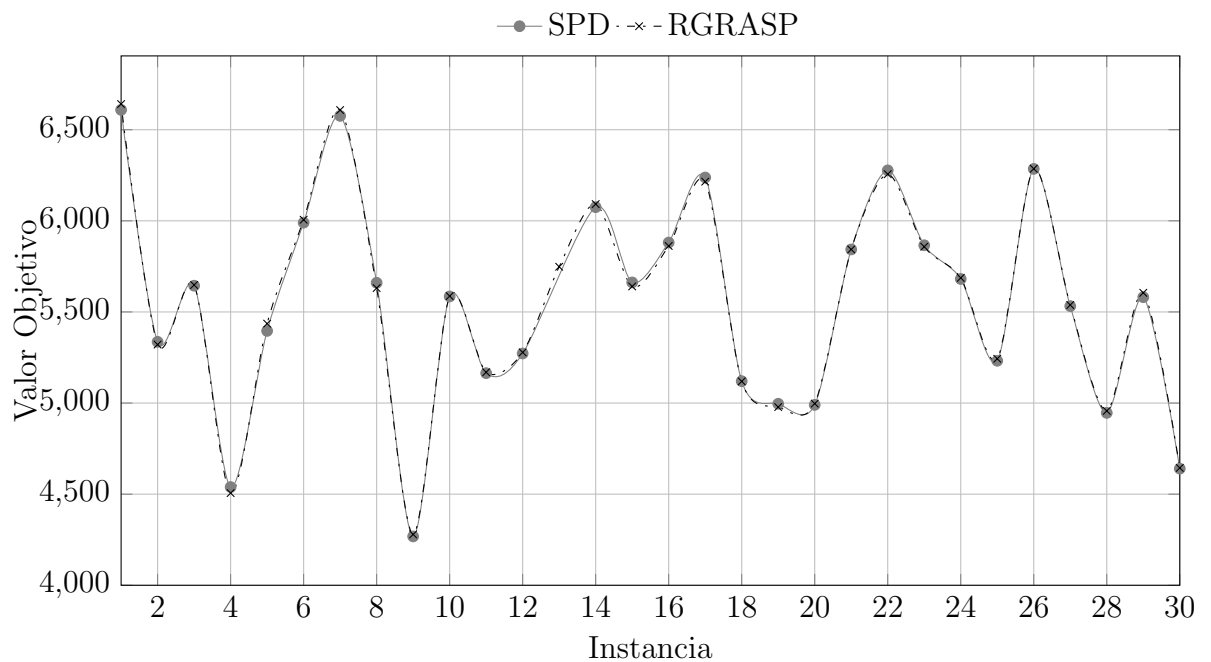


Figura 3.1: Resultados Clase 3P25C: SPD vs RGRASP

Clase	Soluciones	SPD		Soluciones	RGRASP	
		GAP (%)	Tiempo (s)		Mejora (%)	Tiempo (s)
3P25C	29/30	0.46	511.00	30/30	0.02	2.57
3P50C	0/30	–	>3600	30/30	–	9.10
4P100C	0/30	–	>3600	30/30	–	50.50
5P150C	0/30	–	>3600	30/30	–	449.77

Tabla 3.1: Resultados promedio *SPD* vs RGRASP

La primera columna indica la clase a la que pertenecen las instancias, la columna 2 y 5 muestran la cantidad de instancias resueltas del total de cada clase (resueltas/totales), las columnas 3 y 4 hacen referencia al GAP y tiempo promedio que reportó el optimizador CPLEX, la sexta columna muestra el la mejora que presenta la heurística en comparación con el modelo *SPD*, la ultima columna representa el tiempo promedio reportado por el RGRASP.

Como se observó en la Sección 2.4.1, CPLEX fue capaz de resolver casi en su totalidad las instancias de la clase *3P25C* con un criterio de paro de una hora o alcanzar un GAP de 1 %.

Nuestro RGRASP mostró resultados competitivos, en la clase *3P25C*, dando solución a la instancia numero 13, que no pudo resolverse por el modelo *SPD* y además generando, en ciertos casos, resultados mejores que CPLEX, como se muestra en la Figura 3.1, manteniendo un GAP promedio en comparación con CPLEX de 0.024 %, en un tiempo de cómputo de 2.56 segundos en promedio.

Se puede observar que nuestra metaheurística, el RGRASP, fue capaz de obtener solución para todas las instancias de las clases *3P50C*, *4P100C* y *5P150C*. Para cada instancia se generaron doscientas soluciones factibles, de las cuales se reporta la mejor encontrada, en un tiempo en segundos promedio para las clases previamente mencionadas de 9.1, 50.5 y 449.76 respectivamente.

El análisis de los resultados experimentales comprueba que formulación matemática por sí misma es difícil de resolver, utilizando CPLEX 12.6. Además se observó una mejora en tiempo computacional y valor de la función objetivo gracias a nuestra metaheurística, generando soluciones con un GAP no mayor a .74 % (en

instancias con solución factible u óptima conocida) y reportando soluciones para las instancias que no pudieron ser resueltas utilizando la formulación matemática descrita en el capítulo anterior.

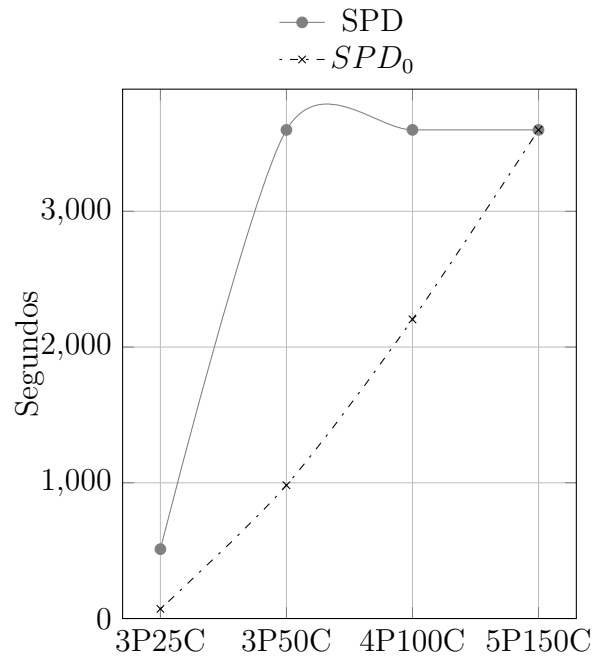
3.4 MODELO MATEMÁTICO CON SOLUCIÓN INICIAL

En el análisis de los resultados de la formulación matemática se concluye que la formulación matemática por si sola es difícil resolver , debido a que se resolvió una escasa cantidad de instancias. Por esta razón se decidió utilizar la solución que se obtiene por la metaheurística, RGRASP, como solución inicial en la formulación matemática, llamando a esta metodología modelo matemático con solución inicial.

Se utilizara SPD_0 , para representar al modelo matemático SPD con solución inicial, la cual fue alimentada por la metaheurística RGRASP.

Con esta parte de la experimentación se espera ayudar a CPLEX a encontrar soluciones factibles en aquéllas instancias en las que no le fue posible anteriormente y reducir el tiempo computacional requerido.

La prueba se realizó usando CPLEX 12.6, con las mismas especificaciones que se describieron en la Sección 2.4.1, la única variante es que el GAP para el criterio de paro se estableció como .5%. Los resultados obtenidos se muestran en la Tabla 3.2.

Figura 3.2: Tiempo promedio: SPD vs SPD_0

Clase	Soluciones	SPD		Soluciones	SPD ₀	
		GAP (%)	Tiempo (s)		GAP (%)	Tiempo (s)
3P25C	29/30	0.46	511.00	30/30	0.37	71.43
3P50C	0/30	–	>3600	30/30	0.46	982.00
4P100C	0/30	–	>3600	30/30	0.51	2204.13
5P150C	0/30	–	>3600	28/30	100.00	>3600

Tabla 3.2: Resultados promedio SPD vs SPD_0

La primera columna indica la clase a la que pertenecen las instancias, la columna 2 y 5 muestran la cantidad de instancias resueltas del total de cada clase (resueltas/totales), las columnas 3 y 4 hacen referencia al GAP y tiempo promedio que reportó el optimizador CPLEX para el modelo *SPD*; de igual manera las columnas 6 y 7 representan el GAP y tiempo promedio que reportó el optimizador CPLEX para el modelo *SPD* con solución inicial, provista por el metaheurístico RGRASP.

Cuando utilizamos la heurística propuesta para proveer una solución inicial al problema en CPLEX se logró resolver todas las instancias de las clases *3P25C*, *3P50C* y *4P100C*, logrando un decremento del tiempo de ejecución promedio de 511 segundos a solo 71.4 segundos para la clase *3P25C*, por otro lado se lograron obtener soluciones con mejor evaluación en la función objetivo, por lo que se mejora la calidad de las soluciones y se decremento el tiempo de solución.

Se observa que al igual que en a clase *3P25C*, se logró un gran decremento en el tiempo de cómputo de las clases *3P50C* y *4P100C*.

Lo que es importante mencionar es que utilizando el RGRASP en combinación con la formulación matemática se logró obtener soluciones para todas las instancias de las clases *3P50C* y *4P100C*, logrando mejorar los resultados reportados por la heurística y los que se habían obtenido con CPLEX sin solución inicial. Reportando un GAP promedio de 0.45 % y 0.51 % respectivamente. Por otra parte se logró generar soluciones para 28 de las 30 instancias de la clase *5P150C*.

CAPÍTULO 4

PROBLEMA DE DISTRIBUCIÓN MULTI-PRODUCTO

En este capítulo se aborda una extensión del problema de estudio, que se presentó en la Sección 2.3.1, en el que se toman en cuenta múltiples productos. La motivación de esta extensión al problema fue representar de mejor manera el caso real de la empresa cervecera, donde normalmente no se cuenta con un solo producto a distribuir sino con un conjunto de productos.

Como se mencionó en el Capítulo 2, el problema de ruteo de vehículos (vehicle routing problem) fue propuesto por primera vez en Dantzig and Ramser (1959) y el nombre con que se refirieron a él fue problema de despacho de camiones (The truck dispatching problem). El problema buscaba el ruteo óptimo de camiones que distribuían gasolina entre una terminal de carga y un conjunto de estaciones de servicio. Sin embargo, el problema de ruteo de vehículos que se presenta en este capítulo también incluye características del problema de asignación, a dicho problema se hará referencia como VRP con asignación de vehículos (A).

Como se mencionó en el Capítulo 1, existen diferentes extensiones del problema de ruteo de vehículos. En la literatura se encuentran investigaciones con una o más extensiones y hacemos referencia a estas a continuación.

4.1 TRABAJO RELACIONADO

Al igual que el VRP mono-producto, el VRP con múltiples productos se ha investigado extensamente, existiendo así una gran variedad de métodos de solución, ya sea exactos o heurísticos.

Entre los métodos heurísticos y metaheurísticos para resolver el VRP con múltiples productos se encuentran algunas metodologías como colonia de hormigas ((Jair et al., 2013)), búsqueda por entornos variables (Popović et al. (2012)), búsqueda por grandes vecindarios (Kilby and Urli (2016)), heurísticas híbridas (Hernández-Pérez et al. (2015)) y búsqueda local iterada (Cattaruzza et al. (2014)).

Por otro lado se encuentran investigaciones que utilizaron metodologías exactas, entre los cuales se encuentran modelos de programación lineal entera mixta (Ramkumar et al. (2012)), branch-and-cut (Lahyani et al. (2015)) y branch – price and cut (Archetti et al. (2015)).

Para detallar el estado del arte referente al segundo problema abordado, en esta tesis, se utilizará una tabla similar a la del Capítulo 2, mostrando, representando ahora los problemas de ruteo de vehículos con múltiples productos más relevantes en la literatura.

Referencia	FF	HF	MD	MT	SD	P	D	TW	A	N	MV	RD
Ramkumar et al. (2012)	✓		✓		✓	✓					✓	✓
Kilby and Urli (2016)	✓	✓			✓	✓					✓	
Lahyani et al. (2015)	✓	✓			✓	✓					✓	
Popović et al. (2012)	✓					✓				✓	✓	
Cattaruzza et al. (2014)				✓				✓			✓	✓
Jair et al. (2013)		✓						✓			✓	
Archetti et al. (2015)	✓				✓						✓	

Tabla 4.1: Trabajo relacionado al VRP multi-producto con restricciones adicionales

En cuanto a la investigación del estado del arte para el caso de múltiples productos, no encontramos un trabajo que considere de manera simultánea el conjunto de restricciones que se tienen en nuestro problema de estudio.

4.2 FORMULACIÓN MATEMÁTICA

En esta sección se propone un modelo de programación lineal entera mixta, para representar el problema de distribución de múltiples productos con asignación de vehículos a plantas, diseño de rutas, ventanas de tiempo, fechas límite de entregas y restricciones de capacidad.

Este modelo matemático es una extensión del que se describió anteriormente en la Sección 2.3.2, ya que toma en cuenta múltiples productos. Considerando la notación de la Sección 2.3.1, en esta sección se introduce el conjunto H , este conjunto contiene los productos que se pueden distribuir en una instancia dada.

4.2.1 MODELO MATEMÁTICO DEL PROBLEMA

El modelo matemático del problema multi-producto es descrito por las expresiones de la (4.1) a (4.11).

Conjuntos

I	Plantas de Producción.
J	Centros de Distribución.
N	$I \cup J$.
V	Viajes.
K	Vehículos.
K'	Flota de vehículos privada, $K' \subset K$.
P	Periodos.
A	Arcos.
H	Productos.

Parámetros

T	Tiempo máximo de la ruta por período.
$Q_{i,h}$	Inventario disponible de producto h en la planta $i \in I$.
$D_{j,h}$	Demanda de producto h en el centro de distribución $j \in J$.
L_j	Fecha límite (due-date) del centro de distribución $j \in J$.
a_j	Inicio de la ventana de tiempo del centro de distribución $j \in J$.
b_j	Fin de la ventana de tiempo del centro de distribución $j \in J$.
q_k	Capacidad del vehículo $k \in K$.
$t_{i,j}$	Tiempo de traslado entre los nodos (i,j) , $i, j \in N \mid (i, j) \in A$.
$c_{i,j,k}$	Costo de viaje del vehículo k desde la planta i al centro de distribución j , $k \in K$, $(i, j) \in A$.

VARIABLES DE DECISIÓN

$$x_{ijvk}^p = \begin{cases} 1 & \text{Si la planta } i \text{ envía producto al centro de distribución } j \\ & \text{en el viaje } v \text{ del vehículo } k, \text{ en el período } p, \\ 0 & \text{en otro caso.} \end{cases}$$

$$u_{ik} = \begin{cases} 1 & \text{Si el vehículo } k \in K' \\ & \text{es asignado a la planta } i \in I, \\ 0 & \text{en otro caso.} \end{cases}$$

s_{vk}^p Tiempo en que el vehículo k parte de la planta para realizar el viaje v del período p .

y_{hij}^{vkp} Cantidad de producto h , enviado de la planta i al centro de distribución j en el viaje v del vehículo k , en el período p .

$$(MPD) \text{Min} \sum_{(i,j) \in A} \sum_{v \in V} \sum_{k \in K} \sum_{p \in P} (c_{ijk} x_{ijvk}^p) \quad (4.1)$$

sujeto a:

$$\sum_{(i,j) \in A} \sum_{v \in V} \sum_{\substack{k \in K \\ p \in P, \\ p \leq L_j}} (y_{hij}^{vkp}) = D_{jh} \quad \forall j \in J, h \in H \quad (4.2)$$

$$\sum_{(i,j) \in A} \sum_{v \in V} \sum_{k \in K} \sum_{p \in P} (y_{hij}^{vkp}) \leq Q_{ih} \quad \forall i \in I, h \in H \quad (4.3)$$

$$s_{vk}^p + \sum_{(i,j) \in A} x_{ijvk}^p (t_{ij} + t_{ji}) \leq s_{v+1,k}^p \quad \forall v \in V, k \in K, p \in P \mid v+1 \in V \quad (4.4)$$

$$s_{vk}^p + \sum_{(i,j) \in A} (t_{ij} x_{ijvk}^p) \geq \sum_{(i,j) \in A} (a_j x_{ijvk}^p) \quad \forall v \in V, k \in K, p \in P \quad (4.5)$$

$$s_{vk}^p + \sum_{(i,j) \in A} (t_{ij} x_{ijvk}^p) \leq \sum_{(i,j) \in A} (b_j x_{ijvk}^p) \quad \forall v \in V, k \in K, p \in P \quad (4.6)$$

$$\sum_{h \in H} y_{hij}^{vkp} \leq q_k x_{ijvk}^p \quad \forall i \in I, j \in J, v \in V, k \in K, p \in P \quad (4.7)$$

$$x_{ijvk}^p \leq u_{ik} \quad \forall i \in I, j \in J, v \in V, k \in K', p \in P \quad (4.8)$$

$$\sum_{i \in I} u_{ik} = 1 \quad \forall k \in K' \quad (4.9)$$

$$\sum_{(i,j) \in A} x_{ijvk}^p \leq 1 \quad \forall v \in V, k \in K, p \in P \quad (4.10)$$

$$s_{vk}^p + \sum_{(i,j) \in A} x_{ijvk}^p (t_{ij} + t_{ji}) \leq T \quad \forall v \in V, k \in K, p \in P \quad (4.11)$$

La función objetivo (4.1) busca minimizar el costo de los viajes realizados. La expresión (4.2) garantiza que la demanda de los clientes se satisfaga. Las restricciones (4.3) establecen que el inventario de cada planta no sea excedido. El conjunto de inecuaciones (4.4) asegura que los vehículos no pueden empezar un nuevo viaje sin haber concluido el anterior. El conjunto de restricciones (4.5) – (4.6) se encargan de asegurar que las ventanas de tiempo sean respetadas para todos los clientes. La restricción (4.7) asegura que la capacidad de los vehículos no sea excedida. Las inecuaciones (4.8) – (4.9) son responsables de la asignación de la flota privada, asegurando que solo se realicen viajes para la planta asignada y que los vehículos solo sean asignados una sola vez. La restricción (4.10) limita a los vehículos a visitar solamente un centro de distribución en cada viaje, por último la restricción (4.11) garantiza que todos los viajes finalizan antes de que se acabe la duración del período.

4.3 INSTANCIAS DEL PROBLEMA

Para poder determinar el alcance y desempeño del modelo matemático mostrado anteriormente fue necesario realizar experimentación computacional sobre un conjunto de instancias. Para el problema multi-producto se creó un conjunto de 95 instancias divididas en cuatro clases.

Las instancias para el problema multi-producto fueron creadas de manera similar a las instancias que se explicaron en la Sección 2.4. Los parámetros utilizados para la creación del nombre de la instancia fueron los siguientes:

- P – plantas de producción
- C – centros de distribución
- H – productos
- TW – longitud de las ventanas de tiempo
- K – Cantidad de vehículos

Para la creación de instancias se utilizaron parámetros de hasta siete plantas de producción, 200 centros de distribución y se combinó con uno, tres y hasta cinco productos. Las cuatro clases que se crearon fueron las siguientes *10C*, *50C*, *100C* y *200C*. Para la creación de ventanas de tiempo se utilizaron tres rangos de ventanas de tiempo:

- 1TW – [3,8]
- 2TW – [8,16]
- 3TW – [16,24]

Utilizando estos tres rangos el generador de instancias seleccionaba un número aleatorio entre el límite inferior y el superior del rango. Por ejemplo, suponer que uti-

lizamos el rango 3TW, en este caso podríamos seleccionar una longitud de ventana de tiempo entre 16 y 24.

Utilizando lo explicado anteriormente podremos ejemplificar la instancia *1P50C-3H3TW75K*. Esta instancia cuenta con una planta de producción y 50 centros de distribución. Se cuentan con 3 productos a distribuir y sabemos que las ventanas de tiempo de los centros de distribución tendrán una longitud de tiempo entre [16,24] y para realizar el envío se cuentan con 75 vehículos compuestos por la flota privada y la flota externa.

4.3.1 RESULTADOS COMPUTACIONALES

La experimentación del problema multi-producto se realizó sobre una computadora con *Intel Xeon E5450 @ 3.00GHz* y *32GB* de memoria RAM. Para la experimentación del modelo matemático se fijaron parámetros en CPLEX, de tal manera que se consideraron los siguientes criterios de paro: una hora como tiempo máximo o GAP de optimalidad de 0%. El modelo matemático fue codificado en C++ con las librerías de CPLEX.

La configuración que se utilizó en CPLEX fue la siguiente:

- Tiempo Límite: 1 hora
- Hilos utilizados: 1 hilo
- GAP: 0%

El modelo matemático fue probado en las 95 instancias divididas en las clases *10C*, *50C*, *100C* y *200C*. Los resultados obtenidos de la experimentación se muestran a detalle en el Apéndice A y en la Tabla 4.2 se muestra una síntesis de los mismos.

Clase	Soluciones	MPD	
		GAP (%)	Tiempo (s)
10C	28/36	0.40	2548.05
50C	1/32	0	3600
100C	0/21	–	3600
200C	0/6	–	3600

Tabla 4.2: Resultados promedio reportados por CPLEX, modelo *MPD*

La primera columna indica la clase a la que pertenecen las instancias, la segunda columna muestra la cantidad de instancias que fueron resueltas, del total de instancias de cada clase (resueltas/totales), las ultimas columnas hacen referencia al GAP y tiempo promedio que reporto el optimizador CPLEX.

Se puede observar que CPLEX logró resolver una gran cantidad de instancias de la clase *10C*, resolviendo 28 de las 36 instancias, logrando obtener soluciones óptimas en 12 casos y en general alcanzó un GAP promedio de .4% en un tiempo promedio de 2548 segundos.

El análisis de las clases restantes muestra que en la clase *50C* se pudo resolver solamente una instancia. En las demás clases le fue imposible a CPLEX resolver las instancias en el tiempo límite de 1 hora. Aunado a lo anterior no se encontró una cota o solución factible para ninguna instancia. Los resultados muestran que la formulación matemática por si misma en casos con más de 50 centros de distribución es difícil de resolver. Es por esta razón que se presenta la oportunidad de desarrollar de un método heurístico que permita dar solución a las instancias en estas clases.

CAPÍTULO 5

DISTRIBUCIÓN MULTI-PRODUCTO: ALGORITMO PROPUESTO

El método propuesto en esta investigación para el problema multi-producto proviene de la combinación de una heurística constructiva con aleatoriedad y un modelo de cobertura de conjuntos (set covering). El método de solución resuelve una instancia, mediante la ejecución de la heurística constructiva durante un número dado de iteraciones, cada iteración de la heurística genera una solución usualmente distinta, esto se debe a que la heurística constructiva cuenta con aleatoriedad. La heurística constructiva cuenta con dos criterios de paro, estos son completar cien soluciones factibles o alcanzar un tiempo límite de 20 minutos.

Las rutas que proveen las soluciones que se obtuvieron mediante la heurística constructiva son guardadas y utilizadas como conjuntos e introducidas como parámetros en el modelo de cobertura de conjuntos, posteriormente este modelo se resuelve utilizando CPLEX 12.6. La solución que se obtiene del modelo se reporta como mejor solución encontrada.

Es importante mencionar que aunque la solución encontrada por el modelo de cobertura de conjuntos sea óptima esto solamente implica que utilizando las rutas encontradas por la heurística no existe mejor solución, mas no implica que sea la solución óptima al problema maestro detallado anteriormente en la Sección 4.2.1, ya que el modelo de cobertura de conjuntos es un modelo simplificado del

modelo maestro. La heurística tanto como el modelo de cobertura de conjuntos serán explicados a detalle a continuación.

5.1 HEURÍSTICA CONSTRUCTIVA CON ALEATORIEDAD

La idea principal de la heurística constructiva con aleatoriedad nace del procedimiento de seleccionar el viaje v'_j con el menor valor, de un parámetro de referencia, para cada cliente $j \in J$ y almacenar dichos viajes en una lista de candidatos (CL), por lo que se tiene un viaje por cliente.

Para la creación de la lista de candidatos se cuenta con el parámetro β , que indica qué parámetro, de cada viaje $v \in V$, será utilizado como referencia en la creación de la lista de candidatos, se fija $\beta = 1$ para utilizar el costo por unidad (w_v) y por otro lado utilizar $\beta = 2$ indicara el uso del parámetro costo de envío (c_v).

Debido a que se desea generar un conjunto de soluciones diversas y a su vez evitar soluciones no factibles, se filtro la lista de candidatos (CL) a una la lista restringida de candidatos (RCL).

Sea e_{min} la mínima cantidad de arcos incidentes que presenta algún cliente $j \in J$, además, sea $e(v)$ una función que indica la cantidad de arcos activos incidentes al centro de distribución γ_j y utilizando la notación explicada a continuación. Existen 2 formas en las que se realizara el filtro a la lista restringida de candidatos.

La primera es efectuada cuando existen clientes con un solo arco incidente activo, $e_{min} = 1$, en este caso se define la lista restringida de candidatos de la siguiente manera $RCL = \{v \in CL \mid e(v) = e_{min}\}$, de esta forma se logra conservar la factibilidad. El segundo caso en el que todos los clientes tienen más de un arco incidente, $e_{min} > 1$, se define la lista restringida de candidatos de la siguiente manera $RCL = CL$, logrando obtener de esta manera soluciones diversas.

Posteriormente se selecciona un viaje aleatorio v^* de la lista restringida de candidatos, en base a la probabilidad $\rho_v \forall v \in RCL$ y se inserta el viaje v^* en la

solución. Este proceso se realiza hasta que toda la demanda quede satisfecha.

En el Algoritmo 1 se utilizara la notación explicada en la Sección 2.3.2, así como las siguientes definiciones.

Sea V_j el conjunto de viajes factibles al centro de distribución j y $V = \{V_1, V_2, \dots, V_{|J|}\}$ el conjunto de todos los viajes.

Para cada viaje $v \in V$, se definen los siguientes parámetros y definiciones:

- $c_v :=$ costo de envío al realizar el viaje v .
- $\theta_v :=$ vehículo que asociado al viaje v .
- $w_v := \frac{c_v}{q_{\theta_v}}$.
- $\gamma_v :=$ Centro de distribución al que se dirige el viaje v .
- $\lambda_v :=$ Planta de producción de la que parte el viaje v .
- $e(v) :=$ Cantidad de arcos activos incidentes al centro de distribución γ_j .
- $\rho(v) := \frac{\frac{|P|}{L_{\gamma_v}}}{\sum_{v \in RCL} \frac{|P|}{L_{\gamma_v}}}$.
- $e_{min} :=$ Valor mínimo actual, diferente de 0, evaluado por la función p , de los viajes $v \in CL$.

La heurística constructiva propuesta hace uso de la notación explicada en las Secciones 2.3.1 y 4.2.1, y se detalla en el Algoritmo 4.

Algoritmo 4 MPD-H**Entrada:** β Parámetro que indica como se creará la lista de candidatos**Salida:** *sol* Solución

```

1: Sea  $V_j$  el conjunto de viajes factibles al centro de distribución  $j$ 
2:  $V \leftarrow \{V_1, V_2, \dots, V_{|J|}\}$ 
3:  $sol \leftarrow \emptyset$ 
4: mientras exista  $D_{jh} > 0, j \in J, h \in H$  hacer
5:   si  $\beta = 1$ , entonces
6:     Sea  $v'_j \leftarrow \operatorname{argmin}_{v \in V_j} \{w_v : e(v) > 0\}$ 
7:      $CL \leftarrow \bigcup_{j \in J} v'_j$ 
8:   si no
9:     Sea  $v'_j \leftarrow \operatorname{argmin}_{v \in V_j} \{c_v : e(v) > 0\}$ 
10:     $CL \leftarrow \bigcup_{j \in J} v'_j$ 
11:   fin si
12:   si  $CL = \emptyset$ , entonces Retorna Solución no factible
13:   fin si
14:   si  $e_{min} = 1$ , entonces
15:      $RCL \leftarrow \{v \in CL \mid e(v) = e_{min}\}$ 
16:   si no
17:      $RCL \leftarrow CL$ 
18:   fin si
19:   Seleccionar un viaje aleatorio  $v^*$ , donde cada viaje  $v \in RCL$  tiene probabilidad  $\rho(v)$  de ser seleccionado.
20:   para cada producto  $h \in H$  hacer
21:      $envio_h \leftarrow 0$  {Cantidad de producto a enviar de tipo  $h$ }
22:   fin para
23:    $k \leftarrow 0$ 
24:   para cada producto  $h \in H$  hacer
25:     si  $k \geq q_{\theta_{v^*}}$ , entonces
26:       salir para cada
27:     fin si
28:      $min^e \leftarrow \min\{Q_{\lambda_{v^*}, h}, D_{\gamma_{v^*}, h}, q_{\theta_{v^*}} - k\}$ 
29:     si  $min^e > 0$ , entonces
30:        $envio_h \leftarrow envio_h + min^e$ 
31:        $k \leftarrow k + min^e$ 
32:     fin si
33:   fin para
34:   si  $k = 0$ , entonces
35:     El viaje  $v^*$  es factible pero no hay producto a enviar
36:      $V \leftarrow V \setminus \{v^*\}$ 
37:   si no
38:     Insertar viaje  $v^*$  en sol
39:     si  $\theta_{v^*} \in K'$  & El vehículo  $\theta_{v^*}$  no esta asignado, entonces
40:       Asignar vehículo  $\theta_{v^*}$  a la planta  $\lambda_{v^*}$ 
41:       Eliminar cada viaje  $v \in V$  tal que  $\theta_v = \theta_{v^*}$  &  $\lambda_v \neq \lambda_{v^*}$ 
42:     fin si
43:     para cada producto  $h \in H$  hacer
44:        $Q_{\lambda_{v^*}, h} \leftarrow Q_{\lambda_{v^*}, h} - envio_h$ 
45:        $D_{\gamma_{v^*}, h} \leftarrow D_{\gamma_{v^*}, h} - envio_h$ 
46:     fin para
47:     si  $\sum_{h \in H} Q_{\lambda_{v^*}, h} = 0$ , entonces
48:       Eliminar cada viaje  $v \in V$  tal que  $\lambda_v = \lambda_{v^*}$ 
49:     fin si
50:     si  $\sum_{h \in H} D_{\gamma_{v^*}, h} = 0$ , entonces
51:       Eliminar cada viaje  $v \in V$  tal que  $\gamma_v = \gamma_{v^*}$ 
52:     fin si
53:   fin si
54:   Actualizar aristas activas
55:   Eliminar  $v \in V$  tal que  $v$  no es factible
56: fin mientras
57: Retorna sol

```

5.2 MODELO DE COBERTURA DE CONJUNTOS

El siguiente modelo matemático está basado en el problema de cobertura de conjuntos (Apéndice B), se plantea la situación del problema de estudio, tomando en cuenta rutas ya predefinidas, estas rutas representan los conjuntos del modelo.

Conjuntos:

I	Plantas de Producción.
J	Centros de Distribución.
V	Viajes.
K	Vehículos.
K'	Flota de vehículos privada, $K' \subset K$.
P	Periodos.
H	Productos.
R	Rutas.

Parámetros

T_{rv}^{ij}	Contiene el valor 1 si el viaje v de la ruta r , viaja de la planta i al centro de distribución j .
$Q_{i,h}$	Inventario disponible de producto h en la planta $i \in I$.
$D_{j,h}$	Demanda de producto h en el centro de distribución $j \in J$.
L_j	Fecha límite (due-date) del centro de distribución $j \in J$.
q_r	Capacidad del vehículo que realiza la ruta $r \in R$.
c_r	Costo de seleccionar la ruta $r \in R$.
κ_r	Vehículo que realiza la ruta $r \in R$.
δ_r	Planta asignada al vehículo utilizado por la ruta $r \in R$.

Variables de decisión

$$x_r^p = \begin{cases} 1 & \text{si la ruta } r \text{ es seleccionada en el período (día) } p, \\ 0 & \text{en otro caso.} \end{cases}$$

$$u_{ik} = \begin{cases} 1 & \text{Si el vehículo } k \in K' \text{ es asignado a la planta } i \in I, \\ 0 & \text{en otro caso.} \end{cases}$$

y_{hrv}^p Cantidad de producto h , entregado por la ruta r en el viaje v del período p .

$$(MPD-SC) \text{Min} \sum_{r \in R} \sum_{p \in P} (c_r x_r^p) \quad (5.1)$$

sujeto a:

$$\sum_{r \in R} \sum_{v \in V} \sum_{\substack{p \in P, \\ p \leq L_j}} (\sum_{i \in I} T_{rv}^{ij} y_{hrv}^p) = D_{jh} \quad \forall j \in J, h \in H \quad (5.2)$$

$$\sum_{r \in R} \sum_{v \in V} \sum_{p \in P} (\sum_{j \in J} T_{rv}^{ij} y_{hrv}^p) \leq Q_{ih} \quad \forall i \in I, h \in H \quad (5.3)$$

$$\sum_{h \in H} y_{hrv}^p \leq q_r x_r^p \quad \forall r \in R, v \in V, p \in P \quad (5.4)$$

$$x_r^p \leq u_{\delta_r \kappa_r} \quad \forall r \in R, p \in P \mid \kappa_r \in K' \quad (5.5)$$

$$\sum_{i \in I} u_{ik} \leq 1 \quad \forall k \in K' \quad (5.6)$$

$$\sum_{\substack{r \in R: \\ \kappa_r = k}} x_r^p \leq 1 \quad \forall k \in K, p \in P \quad (5.7)$$

La función objetivo (5.1) busca minimizar el costo de las rutas seleccionadas, respetando las restricciones del problema. Las restricciones (5.2) garantizan que toda la demanda debe ser satisfecha. El conjunto de inecuaciones (5.3) establece que el inventario de cada planta de producción no debe ser excedido. El conjunto de restricciones (5.4) asegura que la capacidad de los vehículos no será excedida por las rutas. Las desigualdades (5.5) – (5.6) son las encargadas de la asignación de vehículos de la flota privada y que dichos vehículos solo sirvan a su planta asignada. Las inecuaciones (5.7) garantizan que los vehículos son utilizados como máximo por una ruta en cada período.

5.2.1 METODOLOGÍA DE SOLUCIÓN

El algoritmo que se propone para resolver el caso multi-producto es un modelo matemático de cobertura de conjuntos combinado con una heurística constructiva con aleatoriedad.

Primero se generan soluciones al problema utilizando la heurística *MPD-H* detallada en la Sección 4, utilizando como criterio de paro alcanzar 100 soluciones generadas o completar 20 minutos de ejecución. Cada vez que se genera una solución se almacenan las rutas, así como el costo total de la ruta, viajes realizados y vehículo utilizado. Una vez que el heurístico termina la ejecución se utilizan las rutas almacenadas para alimentar el modelo de cobertura de conjuntos mostrado anteriormente, el cual es resuelto a optimalidad con CPLEX 12.6. Además, la heurística *MPD-H* reporta la mejor solución encontrada y a su vez se envía como solución inicial al modelo *MPD-SC*.

La configuración que se utilizó en CPLEX para la ejecución del modelo de cobertura de conjuntos fue la siguiente:

- Tiempo Límite: 1 hora
- Hilos utilizados: 1 hilo
- GAP: 0%
- Memoria de Trabajo(WorkMem): 1250 Megabytes

Para la prueba experimental se utilizaron las instancias creadas para el problema multi-producto y comparó la solución obtenida considerando las instancias que pudieron ser resueltas por CPLEX, usando el modelo completo de la Sección 4.2.1.

5.3 ANÁLISIS EXPERIMENTAL

Se probó la metodología de solución propuesta en las 95 instancias, que a su vez agrupamos en clases (10C, 50C, 100C y 200C) para su mejor análisis. Los resultados de la experimentación para instancias de la clase 10C se muestran en la Figura 5.1.

Se utilizó la misma computadora para ejecutar toda la experimentación referente al problema multi-producto, la cual cuenta con un CPU *Intel Xeon E5450 @ 3.00GHz* y *32GB* de memoria RAM. Los resultados detallados de la experimentación se encuentran en el Apéndice A.

5.3.1 HEURÍSTICA CONSTRUCTIVA Y MODELO DE COBERTURA DE CONJUNTOS

Para la fase de prueba de nuestro método de solución se ejecutaron la heurística constructiva con aleatoriedad y el modelo de cobertura de conjuntos. Se desea observar el grado de mejora que ofrece el utilizar el modelo de cobertura de conjuntos considerando las soluciones generadas mediante nuestra heurística constructiva.

La experimentación que se llevó a cabo en esta sección se realizó de la siguiente manera: Para cada instancia primero se ejecutó la heurística constructiva, al finalizar la ejecución se reportó la mejor solución encontrada y se alimentó, con todas las rutas factibles creadas, al modelo de cobertura de conjuntos. Posteriormente se resolvió a optimalidad el modelo de cobertura de conjuntos, usando CPLEX 12.6. A continuación se presentan los resultados y su respectivo análisis.

Clase	Soluciones	MPD-H		Soluciones	MPD-SC		Tiempo (s)
		Mejora (%)	Tiempo (s)		GAP (%)	Mejora (%)	
10C	36/36	1.34	2.56	36/36	0.32	0.45	2347.22
50C	32/32	1.95	12.84	32/32	10.58	1.95	3544.88
100C	21/21	–	31.76	21/21	38.66	–	3458.90
200C	6/6	–	116.50	6/6	2.59	–	3600

Tabla 5.1: Resultados promedio: MPD-H vs MPD-SC

En la Tabla 5.1, la cabecera *MPD-H* representa a la heurística constructiva y *MPD-SC* al modelo de cobertura de conjuntos resuelto con CPLEX 12.6. La primera columna indica la clase a la que pertenecen las instancias, la columna 2 y 5 muestran la cantidad de instancias resueltas del total de cada clase (resueltas/totales), la columna 3 y 7 representan el porcentaje de mejora en comparación con las soluciones provistas por el modelo *MPD*, las columnas 4 y 8 señalan la cantidad de segundos promedio que la heurística *MPD-H* y el modelo *MPD-SC* reportaron al finalizar su ejecución, por ultimo la sexta columna representa el GAP promedio que reporto el optimizador CPLEX para el modelo *MPD-SC*.

En la Tabla 5.1 y 5.2, una mejora del 0% indicaría que se obtuvo la misma solución, así mismo un mejora negativa indica una mejor solución y una mejora positiva indicaría lo opuesto.

En la Sección 4.3.1 se observó que usando el modelo matemático para el problema multi-producto CPLEX solo fue capaz de resolver 28 de las 36 instancias de la clase *10C* y una instancia de la clase *50C*, por lo que es importante mencionar que tanto la heurística constructiva como el modelo de cobertura de conjuntos alimentado por la heurística constructiva lograron dar solución a todas las instancias.

A continuación se presentan figuras que nos permitirán analizar de manera detallada la mejora que nos provee la utilización del modelo *MPD-SC* en comparación con la heurística constructiva.

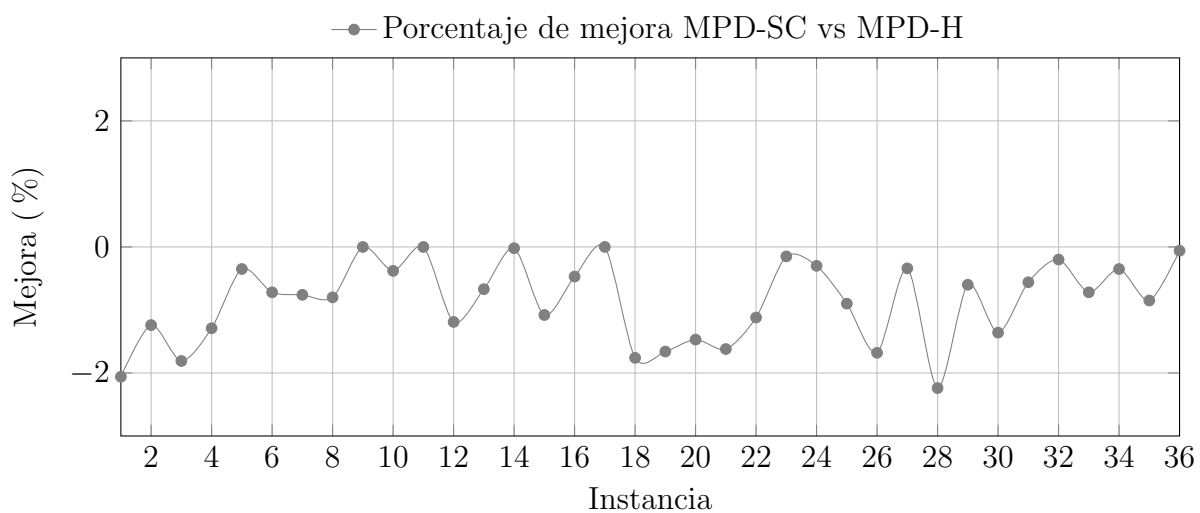


Figura 5.1: Resultados Clase 10C: MPD-SC vs MPD-H

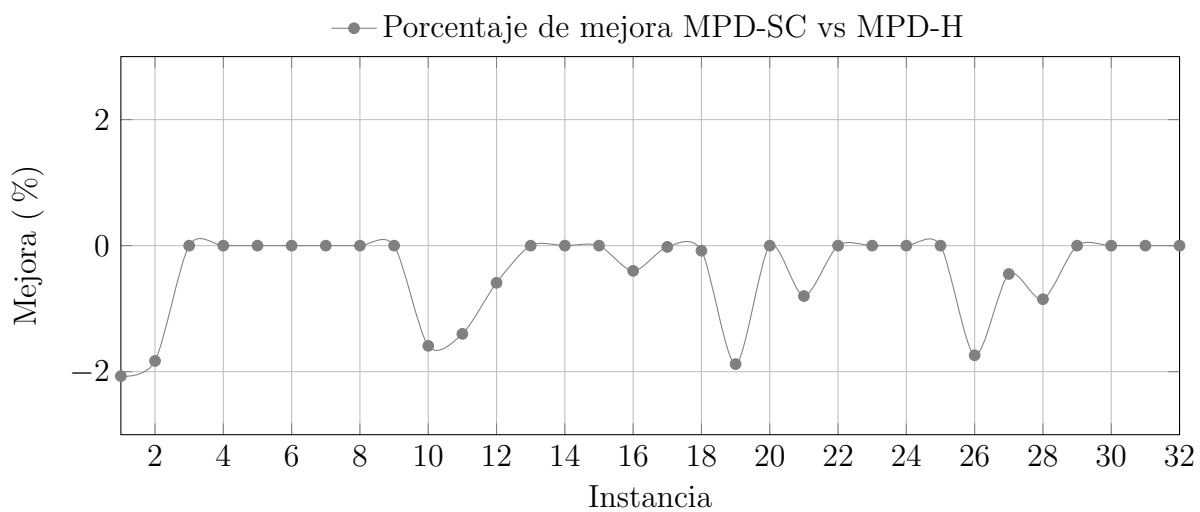


Figura 5.2: Resultados Clase 50C: MPD-SC vs MPD-H

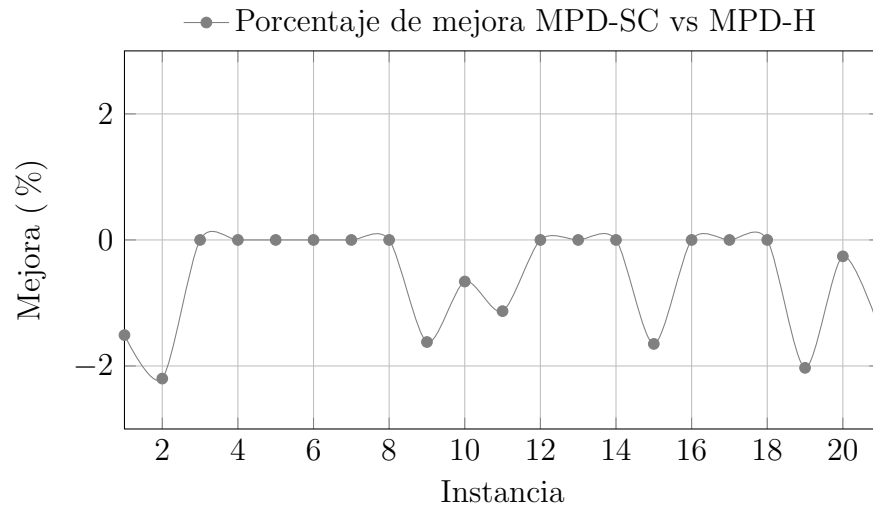


Figura 5.3: Resultados Clase 100C: MPD-SC vs MPD-H

En conclusión se puede observar que la heurística *MPD-H* genera soluciones de una calidad competitiva con las del modelo de cobertura de conjuntos, sin embargo, como se observa en la Tabla 5.1, en la clase *10C*, el porcentaje de mejora se mantiene bajo el 0% indicando así que el modelo *MPD-SC* provee mejores soluciones que la heurística *MPD-H*. Además se observa en las figuras anteriores un porcentaje de mejora negativo para múltiples instancias de las clases *50C* y *100C*; lo que nos indica que el modelo *MPD-SC* genera mejores soluciones. En la clase *200C* ambas metodologías dieron los mismos resultados.

5.3.2 MODELO DEL PROBLEMA MULTI-PRODUCTO Y MODELO DE COBERTURA DE CONJUNTOS

Se mencionaba anteriormente que el modelo de cobertura de conjuntos nos permite encontrar soluciones de mejor calidad que cuando usamos únicamente la heurística *MPD-H*. Ahora se desea comparar su desempeño con los resultados presentados en la Tabla 4.2 del modelo matemático para el problema de distribución multi-producto.

Se presenta a continuación la síntesis de los resultados para analizar ambas metodologías y posteriormente se presentan en una gráfica ambas soluciones, con el fin de medir la calidad de las soluciones que se obtienen utilizando el modelo de cobertura de conjuntos.

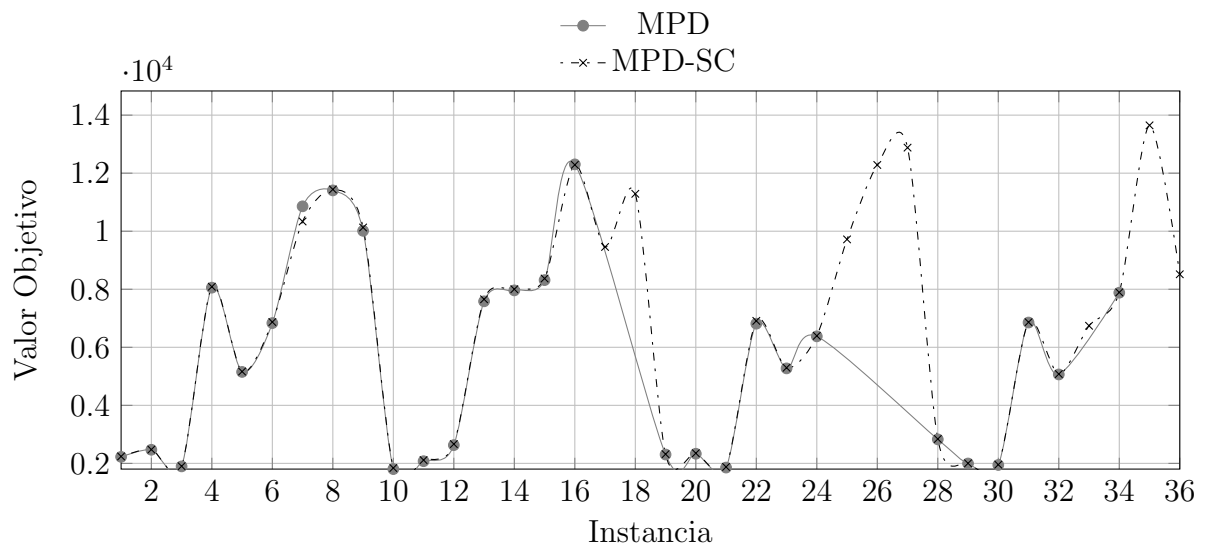


Figura 5.4: Resultados Clase 10C: MPD vs MPD-SC

Clase	Soluciones	MPD		Soluciones	MPD-SC		Tiempo (s)
		GAP (%)	Tiempo (s)		GAP (%)	Mejora (%)	
10C	28/36	0.40	2548.06	36/36	0.32	0.45	2347.22
50C	1/32	0	3600	32/32	10.58	1.95	3544.88
100C	0/21	-	3600	21/21	38.66	-	3458.90
200C	0/6	-	3600	6/6	2.59	-	3600

Tabla 5.2: Resultados promedio MPD vs MPD-SC

En la Tabla 5.2, la primera columna indica la clase a la que pertenecen las instancias, la columna 2 y 5 muestran la cantidad de instancias resueltas del total de cada clase (resueltas/totales), la columna 3 y 6 representan el GAP promedio que reportó el optimizador CPLEX, las columnas 4 y 8 señalan la cantidad de tiempo promedio en finalizar la ejecución de los modelos, por ultimo la séptima columna muestra el porcentaje de mejora en comparación con las soluciones que ha proveído el modelo *MPD*.

Se compara solamente la clase *10C*, debido a que es la clase que cuenta con mayor cantidad de instancias resueltas con el modelo matemático del problema multi-producto. Se puede observar la excelente calidad de las soluciones del modelo de cobertura de conjuntos en comparación con el modelo matemático del problema multi-producto, reportando un GAP promedio de 0.45 %, además se da solución a todas las instancias, como se observa en la Figura 5.4.

CAPÍTULO 6

EPÍLOGO

En esta tesis se abordó un problema de optimización proveniente de una empresa cervecera de la localidad, el cual puede verse como una combinación de un problema de asignación y uno de ruteo de vehículos. A continuación se presentan las conclusiones del trabajo realizado, así como la aportación científica y algunas áreas de oportunidad detectadas.

6.1 APORTACIÓN CIENTÍFICA

La investigación introdujo una variante del problema de ruteo de vehículos, la cual, a nuestro conocimiento, no se había estudiado anteriormente en la literatura. Se presentaron dos modelos de programación lineal entera mixta. Uno para el problema de estudio tomando en cuenta un solo producto y otro para el problema multi-producto.

Para resolver las variantes del problema de estudio, se propusieron dos procedimientos de solución, mismos que pueden usarse como base para resolver otros problemas de ruteo de vehículos. El primer procedimiento fue una metaheurística tipo GRASP reactivo para el problema mono-producto. El segundo procedimiento consistió en resolver, con CPLEX, un modelo de cobertura de conjuntos, utilizando únicamente un subconjunto de las variables (rutas). Estas variables (rutas) fueron generadas a partir de una heurística constructiva.

6.2 CONCLUSIONES

Principalmente se logró representar matemáticamente una situación real que enfrenta una compañía cervecera de la localidad. También se crearon varios conjuntos de instancias para la validación de los modelos y algoritmos propuestos.

Se implementaron dos modelos de programación lineal entera mixta en CPLEX 12.6, mediante el uso de el lenguaje de programación C++. Entonces se realizó la experimentación de dichos modelos sobre las instancias creadas. Gracias a estos resultados se elaboró un análisis que nos permitió determinar el alcance de las formulaciones matemáticas. Se observó que el uso de estos modelos en un optimizador comercial como CPLEX, queda limitado a la resolución de instancias pequeñas; para instancias grandes del problema, el esfuerzo computacional es demasiado costoso. Por esta razón, se procedió al diseño de procedimientos de solución heurística.

El primer método de solución que se realizó fue una metaheurística tipo GRASP reactivo, el cual se probó con las instancias creadas para el problema mono-producto logrando conseguir resultados competitivos, en comparación con las instancias que se pudieron resolver utilizando la formulación matemática propuesta, alcanzando soluciones con un GAP promedio de 0.02%. Posteriormente se utilizó la solución reportada por GRASP como solución inicial en la formulación matemática implementada en CPLEX. Con esto se lograron soluciones enteras para 118 de las 120 instancias de prueba, en un tiempo límite de 1 hora por instancia. Nótese que el GRASP propuesto fue capaz de generar soluciones factibles para todas las instancias de prueba.

El segundo método de solución consistió en resolver, con CPLEX, un modelo de cobertura de conjuntos, utilizando únicamente un subconjunto de las variables (rutas). Estas variables (rutas) fueron generadas a partir de una heurística constructiva. Con este procedimiento se encontraron soluciones para las 95 instancias de prueba, mientras que al utilizar el modelo de programación lineal entera mixta,

CPLEX reportó soluciones factibles únicamente en 29 de las 95 instancias de prueba.

6.3 ÁREAS DE OPORTUNIDAD

El método metaheurístico RGRASP, propuesto en la Sección 3.1, provee resultados de excelente calidad, compitiendo con los resultados óptimos con los que contamos. Sin embargo, es posible refinar cada uno de los componentes del método para generar mejores soluciones.

Por otro lado, en el problema de distribución multi-producto, el modelo de cobertura de conjuntos *MPD-SC*, detallado en la Sección 5.2.1 genera soluciones competitivas, no obstante, depende mucho de la calidad de las rutas generadas por la heurística *MPD-SC*. Actualmente la metodología implica generar múltiples rutas utilizando la heurística *MPD-H* para posteriormente alimentar al modelo *MPD-SC*. Por esta razón, se podría modificar la heurística *MPD-H*, de tal forma que, esta interaccione con el modelo *MPD-SC*, una vez finalizada la ejecución, y de esta manera instruir a la heurística a generar rutas que mejoren la solución. Esto se puede ver como un proceso iterativo entre la heurística y el modelo *MPD-SC*.

APÉNDICE A

RESULTADOS

En este apéndice se detallan los resultados computacionales que se realizaron a lo largo de esta investigación, así como terminología referente a los mismos.

En los resultados se hace referencia a el termino GAP, representando la calidad de la solución reportada, el cual nos provee el optimizador CPLEX 12.6 al terminar su ejecución. Un GAP de 0% indica que la solución que se encontró es óptima.

Por otra parte el termino *Mejora*, que aparece en algunas cabeceras de columnas, fue calculado y se utiliza como punto de referencia para medir el porcentaje de mejora de una solución α en base a una solución β . Un porcentaje de mejora de 0% indicaría que se obtuvo la misma solución, así mismo un resultado negativo indica que α es mejor solución que β y un resultado positivo indicaría lo opuesto.

Es importante mencionar que el obtener un porcentaje de mejora negativo, en comparación con el optimizador CPLEX, solo implica haber obtenido solución con mejor valor en la función objetivo que CPLEX, con los criterios de paro que se fijaron.

Los guiones largos(-), representan dos situaciones. La primera es que CPLEX no fue capaz de conseguir solución factible en un tiempo máximo de 3600 segundos, para dicho modelo matemático. La segunda situación indica que no se pudo calcular un porcentaje de mejoran, ya que no se contaba con un resultado para comparar en dicha instancia, esto para los casos en que una instancia no cuenta con solución reportada por CPLEX, utilizando los modelos matemáticos *SPD* o *MPD* , pero si

pudo ser solucionada con otra metodología.

A.1 EXPERIMENTACIÓN PROBLEMA MONO-PRODUCTO

La experimentación del problema mono-producto se realizó sobre una computadora con un CPU *Intel Xeon E5-2687W @ 3.10GHz* y 64GB de memoria RAM. Para la experimentación del modelo matemático se fijaron los siguientes parámetros en CPLEX:

- Tiempo Límite: 1 hora
- Hilos utilizados: 1 hilo
- GAP: 1 %
- Memoria de Trabajo(WorkMem): 1250 Megabytes

Posteriormente para la experimentación del modelo matemático, utilizando como generador de solución inicial el RGRASP, se modificó el criterio de paro de 1 % GAP a .5 %.

Se utilizará SPD_0 , para representar al modelo matemático SPD con solución inicial, la cual fue alimentada por la metaheurística RGRASP.

No.	Clase: 3P25C Instancia	CPLEX – Modelo SPD			RGRASP			Modelo SPD_0		
		Sol	GAP (%)	Tiempo (s)	Sol	Mejora (%)	Tiempo (s)	Sol	GAP (%)	Tiempo (s)
1	3P25C-3TW-45K1	6609	0.54	103	6641	0.48	3	6597	0.40	87
2	3P25C-3TW-45K2	5336	0.76	174	5321	-0.28	2	5309	0.44	53
3	3P25C-3TW-45K3	5644	0.30	88	5648	0.07	3	5637	0.35	60
4	3P25C-3TW-45K4	4539	0.94	104	4506	-0.72	2	4506	0.22	61
5	3P25C-3TW-45K5	5396	0.18	147	5436	0.74	2	5397	0.41	94
6	3P25C-3TW-45K6	5990	0.41	189	6006	0.26	3	5977	0.36	62
7	3P25C-3TW-45K7	6576	0.45	100	6608	0.48	3	6577	0.49	103
8	3P25C-3TW-45K8	5660	0.91	170	5631	-0.51	2	5631	0.47	70
9	3P25C-3TW-45K9	4268	0	145	4279	0.25	2	4279	0.35	64
10	3P25C-3TW-45K10	5585	0.35	111	5587	0.03	3	5575	0.34	57
11	3P25C-5TW-45K1	5164	0.15	96	5169	0.09	3	5169	0.30	63
12	3P25C-5TW-45K2	5272	0.03	86	5278	0.11	2	5278	0.15	72
13	3P25C-5TW-45K3	-	-	3603	5748	-	3	5748	0.24	67
14	3P25C-5TW-45K4	6075	0.30	94	6092	0.27	2	6078	0.46	80
15	3P25C-5TW-45K5	5662	0.77	556	5640	-0.38	3	5638	0.48	67
16	3P25C-5TW-45K6	5881	0.98	1969	5862	-0.32	2	5835	0.45	96
17	3P25C-5TW-45K7	6238	0.92	231	6215	-0.36	3	6193	0.32	60
18	3P25C-5TW-45K8	5120	0.83	396	5119	-0.01	3	5105	0.49	183
19	3P25C-5TW-45K9	4996	0.56	88	4978	-0.36	3	4978	0.20	64
20	3P25C-5TW-45K10	4990	0.20	98	4997	0.14	3	4997	0.34	80
21	3P25C-8TW-45K1	5843	0.17	331	5843	0	3	5843	0.17	57
22	3P25C-8TW-45K2	6277	0.70	1411	6256	-0.33	3	6256	0.36	55
23	3P25C-8TW-45K3	5866	0.66	3108	5858	-0.13	2	5841	0.32	42
24	3P25C-8TW-45K4	5681	0.70	730	5687	0.10	2	5663	0.49	89
25	3P25C-8TW-45K5	5232	0.05	48	5243	0.21	2	5243	0.26	41
26	3P25C-8TW-45K6	6285	0.64	131	6286	0.01	3	6273	0.46	75
27	3P25C-8TW-45K7	5532	0.10	372	5538	0.10	2	5538	0.21	60
28	3P25C-8TW-45K8	4946	0.02	191	4957	0.22	3	4957	0.48	48
29	3P25C-8TW-45K9	5581	0	272	5605	0.43	3	5605	0.42	59
30	3P25C-8TW-45K10	4640	0.56	188	4644	0.08	2	4621	0.35	74

Tabla A.1: Experimentación: Problema mono-producto – 3P25C

No.	Clase: 3P50C Instancia	CPLEX – Modelo SPD			RGRASP			Modelo SPD_0		
		Sol	GAP (%)	Tiempo (s)	Sol	Mejora (%)	Tiempo (s)	Sol	GAP (%)	Tiempo (s)
1	3P50C-3TW-60K1	-	-	3601	10033	-	10	10006	0.45	454
2	3P50C-3TW-60K2	-	-	3601	11377	-	10	11290	0.45	1539
3	3P50C-3TW-60K3	-	-	3600	10911	-	9	10895	0.42	253
4	3P50C-3TW-60K4	-	-	3601	10390	-	9	10286	0.73	3600
5	3P50C-3TW-60K5	-	-	3600	12520	-	12	12224	0.49	1459
6	3P50C-3TW-60K6	-	-	3600	9850	-	8	9850	0.49	135
7	3P50C-3TW-60K7	-	-	3600	10446	-	8	10423	0.44	227
8	3P50C-3TW-60K8	-	-	3601	12181	-	8	12099	0.28	1292
9	3P50C-3TW-60K9	-	-	3600	10171	-	9	10171	0.47	195
10	3P50C-3TW-60K10	-	-	3601	12086	-	10	11953	0.41	2177
11	3P50C-5TW-60K1	-	-	3601	10100	-	8	10100	0.46	132
12	3P50C-5TW-60K2	-	-	3601	11195	-	9	11137	0.32	602
13	3P50C-5TW-60K3	-	-	3600	12318	-	9	12283	0.46	262
14	3P50C-5TW-60K4	-	-	3601	10576	-	10	10529	0.51	3601
15	3P50C-5TW-60K5	-	-	3600	11027	-	9	11001	0.47	2092
16	3P50C-5TW-60K6	-	-	3600	10887	-	10	10828	1.15	3600
17	3P50C-5TW-60K7	-	-	3600	10803	-	10	10803	0.44	192
18	3P50C-5TW-60K8	-	-	3606	11631	-	9	11604	0.49	1229
19	3P50C-5TW-60K9	-	-	3600	11543	-	11	11467	0.49	2402
20	3P50C-5TW-60K10	-	-	3601	11891	-	10	11862	0.44	351
21	3P50C-8TW-60K1	-	-	3601	12020	-	10	12020	0.38	132
22	3P50C-8TW-60K2	-	-	3600	10762	-	10	10720	0.48	258
23	3P50C-8TW-60K3	-	-	3600	10723	-	9	10723	0.17	140
24	3P50C-8TW-60K4	-	-	3601	9874	-	8	9874	0.40	173
25	3P50C-8TW-60K5	-	-	3601	10820	-	7	10820	0.39	139
26	3P50C-8TW-60K6	-	-	3601	9227	-	7	9227	0.21	168
27	3P50C-8TW-60K7	-	-	3600	11650	-	10	11592	0.49	2168
28	3P50C-8TW-60K8	-	-	3601	9997	-	8	9997	0.25	176
29	3P50C-8TW-60K9	-	-	3600	11349	-	8	11349	0.48	176
30	3P50C-8TW-60K10	-	-	3600	10394	-	8	10394	0.47	136

Tabla A.2: Experimentación: Problema mono-producto – 3P50C

No.	Clase: 4P100C Instancia	CPLEX – Modelo SPD			RGRASP			Modelo SPD ₀		
		Sol	GAP (%)	Tiempo (s)	Sol	Mejora (%)	Tiempo (s)	Sol	GAP (%)	Tiempo (s)
1	4P100C-3TW-105K1	-	-	3604	20531	-	47	20531	0.18	1607
2	4P100C-3TW-105K2	-	-	3610	21731	-	61	21731	0.61	3602
3	4P100C-3TW-105K3	-	-	3603	20524	-	52	20524	0.48	2616
4	4P100C-3TW-105K4	-	-	3602	21821	-	58	21775	0.77	3601
5	4P100C-3TW-105K5	-	-	3603	24177	-	64	24177	1.11	3600
6	4P100C-3TW-105K6	-	-	3602	23767	-	63	23690	1.28	3600
7	4P100C-3TW-105K7	-	-	3637	22302	-	51	22302	0.31	2145
8	4P100C-3TW-105K8	-	-	3688	21356	-	57	21312	0.36	3194
9	4P100C-3TW-105K9	-	-	3602	20074	-	51	20074	0.30	1961
10	4P100C-3TW-105K10	-	-	3603	23091	-	61	23091	0.97	3601
11	4P100C-5TW-105K1	-	-	3628	21557	-	49	21557	0.38	1660
12	4P100C-5TW-105K2	-	-	3604	18949	-	45	18949	0.21	1498
13	4P100C-5TW-105K3	-	-	3602	22459	-	51	22459	0.34	1499
14	4P100C-5TW-105K4	-	-	3605	20876	-	48	20876	0.25	1987
15	4P100C-5TW-105K5	-	-	3730	19463	-	48	19463	0.51	3654
16	4P100C-5TW-105K6	-	-	3621	21619	-	53	21598	0.47	2661
17	4P100C-5TW-105K7	-	-	3602	22688	-	49	22688	0.42	1600
18	4P100C-5TW-105K8	-	-	3713	21516	-	47	21516	0.40	2057
19	4P100C-5TW-105K9	-	-	3602	22708	-	48	22643	0.48	2561
20	4P100C-5TW-105K10	-	-	3635	22764	-	59	22686	2.43	3654
21	4P100C-8TW-105K1	-	-	3602	21650	-	43	21650	0.45	974
22	4P100C-8TW-105K2	-	-	3604	22598	-	45	22598	0.30	1588
23	4P100C-8TW-105K3	-	-	3655	20725	-	48	20725	0.42	1747
24	4P100C-8TW-105K4	-	-	3605	18816	-	43	18816	0.15	1449
25	4P100C-8TW-105K5	-	-	3639	20693	-	42	20693	0.33	1186
26	4P100C-8TW-105K6	-	-	3602	22802	-	46	22802	0.25	1513
27	4P100C-8TW-105K7	-	-	3603	20603	-	46	20603	0.39	1063
28	4P100C-8TW-105K8	-	-	3603	19586	-	44	19586	0.14	1672
29	4P100C-8TW-105K9	-	-	3602	24249	-	48	24249	0.29	1260
30	4P100C-8TW-105K10	-	-	3602	22318	-	48	22318	0.25	1314

Tabla A.3: Experimentación: Problema mono-producto – 4P100C

No.	Clase: 5P150C Instancia	CPLEX – Modelo SPD			RGRASP			Modelo SPD ₀		
		Sol	GAP (%)	Tiempo (s)	Sol	Mejora (%)	Tiempo (s)	Sol	GAP (%)	Tiempo (s)
1	5P150C-3TW-300K1	-	-	3615	32683	-	463	32683	100	3600
2	5P150C-3TW-300K2	-	-	3609	34494	-	472	34494	100	3660
3	5P150C-3TW-300K3	-	-	3611	28410	-	494	28410	100	3612
4	5P150C-3TW-300K4	-	-	3611	31396	-	486	31396	100	3619
5	5P150C-3TW-300K5	-	-	3614	30775	-	508	30775	100	3604
6	5P150C-3TW-300K6	-	-	3610	32896	-	459	32896	100	3686
7	5P150C-3TW-300K7	-	-	3706	32029	-	481	32029	100	3703
8	5P150C-3TW-300K8	-	-	3618	30923	-	482	30923	100	3661
9	5P150C-3TW-300K9	-	-	3616	32452	-	490	32452	100	3601
10	5P150C-3TW-300K10	-	-	3614	31877	-	443	-	-	3609
11	5P150C-5TW-300K1	-	-	3614	31184	-	452	31184	100	3600
12	5P150C-5TW-300K2	-	-	3614	30729	-	437	30729	100	3603
13	5P150C-5TW-300K3	-	-	3614	33389	-	450	33389	100	3600
14	5P150C-5TW-300K4	-	-	3614	32349	-	430	32349	100	3601
15	5P150C-5TW-300K5	-	-	3614	35512	-	453	35512	100	3601
16	5P150C-5TW-300K6	-	-	3615	30589	-	426	30589	100	3680
17	5P150C-5TW-300K7	-	-	3615	31985	-	443	-	-	3609
18	5P150C-5TW-300K8	-	-	3616	33068	-	465	33068	100	3601
19	5P150C-5TW-300K9	-	-	3614	30714	-	441	30714	100	3601
20	5P150C-5TW-300K10	-	-	3615	32561	-	449	32561	100	3601
21	5P150C-8TW-300K1	-	-	3615	32016	-	433	32016	100	3601
22	5P150C-8TW-300K2	-	-	3614	33727	-	417	33727	100	3603
23	5P150C-8TW-300K3	-	-	3614	32054	-	429	32054	100	3607
24	5P150C-8TW-300K4	-	-	3614	33776	-	431	33776	100	3608
25	5P150C-8TW-300K5	-	-	3614	32723	-	422	32723	100	3601
26	5P150C-8TW-300K6	-	-	3614	31922	-	433	31922	100	3601
27	5P150C-8TW-300K7	-	-	3615	31249	-	430	31249	100	3607
28	5P150C-8TW-300K8	-	-	3615	32338	-	426	32338	100	3601
29	5P150C-8TW-300K9	-	-	3615	34285	-	425	34285	100	3608
30	5P150C-8TW-300K10	-	-	3614	31064	-	423	31064	100	3608

Tabla A.4: Experimentación: Problema mono-producto – 5P150C

A.2 EXPERIMENTACIÓN PROBLEMA MULTI-PRODUCTO

La experimentación del problema multi-producto se realizó sobre una computadora con *Intel Xeon E5450 @ 3.00GHz* y *32GB* de memoria RAM. Para la experimentación del modelo matemático se fijaron parámetros en CPLEX, de tal manera que se utilizaron los siguientes criterios de paro, una hora como tiempo de ejecución máximo o alcanzar el óptimo (GAP 0%). Los modelos matemáticos y la heurística constructiva fueron codificados utilizando el lenguaje de programación C++.

La configuración que se utilizó en CPLEX para la ejecución los modelos matemáticos fue la siguiente:

- Tiempo Límite: 1 hora
- Hilos utilizados: 1 hilo
- GAP: 0%
- Memoria de Trabajo(WorkMem): 1250 Megabytes

No.	Clase: 10C Instancia	CPLEX – Modelo MPD			Heurística MPD-H			CPLEX – Modelo MPD-SC			
		Sol	GAP (%)	Tiempo (s)	Sol	Mejora (%)	Tiempo (s)	Sol	GAP (%)	Mejora (%)	Tiempo (s)
1	1P10C-1H1TW30K	2224	0	8	2286	2.78	0	2239	0	0.67	0
2	1P10C-1H2TW20K	2469	0	3	2507	1.53	0	2476	0.12	0.28	3606
3	1P10C-1H3TW20K	1896	0	6	1937	2.16	0	1902	0	0.31	11
4	1P10C-3H1TW30K	8055	0.33	3813	8193	1.71	1	8087	0	0.39	1827
5	1P10C-3H2TW30K	5151	0.34	3601	5182	0.60	0	5164	0.25	0.25	3900
6	1P10C-3H3TW30K	6834	0.39	3616	6920	1.25	0	6870	0.87	0.52	3616
7	1P10C-5H1TW30K	10858	5.58	3602	10414	-4.08	1	10335	0.31	-4.81	3600
8	1P10C-5H2TW30K	11406	0.40	3657	11532	1.10	0	11440	0.65	0.29	3603
9	1P10C-5H3TW30K	10013	0.45	3602	10128	1.14	1	10128	1.52	1.14	3609
10	3P10C-1H1TW30K	1803	0	11	1847	2.44	0	1840	0	2.05	0
11	3P10C-1H2TW20K	2075	0	213	2113	1.83	0	2113	0	1.83	4
12	3P10C-1H3TW20K	2629	0	10	2697	2.58	0	2665	0	1.36	3
13	3P10C-3H1TW30K	7584	0.31	3608	7706	1.60	1	7654	0.22	0.92	4114
14	3P10C-3H2TW30K	7961	0.26	4399	8003	0.52	16	8001	0.65	0.50	3611
15	3P10C-3H3TW30K	8315	0.45	3600	8461	1.75	1	8370	0.95	0.66	3604
16	3P10C-5H1TW200K	12303	0.61	4464	12344	0.33	18	12286	0.20	-0.13	3601
17	3P10C-5H2TW30K	-	-	3601	9457	-	3	9457	1.07	-	3605
18	3P10C-5H3TW30K	-	-	3601	11490	-	1	11288	0.02	-	3691
19	5P10C-1H1TW30K	2305	0	19	2356	2.21	0	2317	0	0.52	2
20	5P10C-1H2TW20K	2332	0	179	2385	2.27	1	2350	0	0.77	12
21	5P10C-1H3TW20K	1855	0	27	1908	2.85	0	1877	0	1.18	1
22	5P10C-3H1TW30K	6814	0.34	3610	6988	2.55	0	6910	0.37	1.40	3604
23	5P10C-3H2TW30K	5273	0.44	3729	5303	0.56	1	5295	0.64	0.41	3791
24	5P10C-3H3TW30K	6377	0.40	3614	6414	0.58	0	6395	0.58	0.28	3614
25	5P10C-5H1TW30K	-	-	3600	9806	-	1	9718	0.32	-	3601
26	5P10C-5H2TW30K	-	-	3601	12493	-	1	12283	0	-	173
27	5P10C-5H3TW30K	-	-	5755	12924	-	36	12880	0.34	-	3601
28	7P10C-1H1TW30K	2828	0	27	2905	2.72	0	2840	0	0.42	1
29	7P10C-1H2TW20K	1996	0	67	2014	0.90	0	2002	0	0.30	3
30	7P10C-1H3TW20K	1945	0	92	1983	1.95	0	1956	0	0.56	1
31	7P10C-3H1TW30K	6853	0.13	3600	6903	0.72	1	6864	0	0.16	2037
32	7P10C-3H2TW30K	5064	0.35	3973	5088	0.47	2	5078	0.63	0.27	3603
33	7P10C-3H3TW30K	-	-	3601	6791	-	0	6742	0.48	-	3602
34	7P10C-5H1TW200K	7881	0.41	3603	7920	0.49	1	7892	0.19	0.13	3600
35	7P10C-5H2TW30K	-	-	3616	13765	-	2	13648	0.00	-	3643
36	7P10C-5H3TW30K	-	-	3602	8521	-	3	8516	1.16	-	3606

Tabla A.5: Experimentación: Problema multi-producto – 10C

No.	Clase: 50C Instancia	CPLEX – Modelo MPD			Heurística MPD-H			CPLEX – Modelo MPD-SC			
		Sol	GAP (%)	Tiempo (s)	Sol	Mejora (%)	Tiempo (s)	Sol	GAP (%)	Mejora (%)	Tiempo (s)
1	1P50C-1H1TW75K	-	-	3627	11172	-	2	10941	0.04	-	3601
2	1P50C-1H2TW50K	-	-	3601	11225	-	2	11020	0.60	-	4054
3	1P50C-1H3TW50K	11081	0	281	11297	1.94	1	11297	1.31	1.94	3642
4	1P50C-3H1TW75K	-	-	3600	34933	-	7	34933	1.89	-	3628
5	1P50C-3H2TW75K	-	-	3603	37370	-	10	37370	2.81	-	3641
6	1P50C-3H3TW75K	-	-	3602	32372	-	6	32372	1.48	-	4857
7	1P50C-5H1TW500K	-	-	3604	56892	-	11	56892	0.48	-	3601
8	1P50C-5H2TW500K	-	-	3837	52668	-	15	52668	100	-	3705
9	1P50C-5H3TW500K	-	-	3838	54144	-	53	54144	100	-	3612
10	3P50C-1H1TW75K	-	-	6514	11960	-	7	11770	0	-	2076
11	3P50C-1H2TW50K	-	-	3602	12190	-	3	12019	0.74	-	3627
12	3P50C-1H3TW50K	-	-	5378	11298	-	12	11231	0.87	-	3655
13	3P50C-3H1TW75K	-	-	3601	31842	-	5	31842	2.90	-	3600
14	3P50C-3H2TW75K	-	-	3603	37303	-	8	37303	2.28	-	3648
15	3P50C-3H3TW75K	-	-	3956	31754	-	38	31754	1.82	-	4872
16	3P50C-5H1TW500K	-	-	3906	53436	-	14	53224	0.32	-	3600
17	3P50C-5H2TW500K	-	-	3601	50508	-	12	50497	0.56	-	3600
18	3P50C-5H3TW500K	-	-	3666	54204	-	9	54163	0.58	-	3601
19	5P50C-1H1TW100K	-	-	3607	11774	-	2	11553	0	-	74
20	5P50C-1H2TW50K	-	-	3605	10867	-	7	10867	1.93	-	3762
21	5P50C-1H3TW50K	-	-	3941	10817	-	34	10731	1.14	-	3623
22	5P50C-3H1TW75K	-	-	3601	33235	-	10	33235	3.72	-	3600
23	5P50C-3H2TW75K	-	-	4253	32324	-	20	32324	2.18	-	3867
24	5P50C-3H3TW75K	-	-	5131	33085	-	23	33085	2.42	-	5105
25	5P50C-5H3TW500K	-	-	3641	53844	-	14	53844	0.67	-	3928
26	7P50C-1H1TW75K	-	-	6438	11533	-	18	11332	0	-	415
27	7P50C-1H2TW50K	-	-	3608	10648	-	2	10600	1.48	-	3602
28	7P50C-1H3TW50K	-	-	3602	10379	-	2	10291	1.33	-	3633
29	7P50C-3H1TW75K	-	-	3601	37011	-	18	37011	2.68	-	3621
30	7P50C-3H2TW75K	-	-	3605	29573	-	10	29573	1.48	-	3603
31	7P50C-3H3TW75K	-	-	4118	31971	-	19	31971	100	-	4359
32	7P50C-5H3TW500K	-	-	3616	55788	-	17	55788	0.62	-	3624

Tabla A.6: Experimentación: Problema multi-producto – 50C

No.	Clase: 100C Instancia	Cplex – Modelo MPD			Heurística MPD-H			Cplex – Modelo MPD-SC			
		Sol	GAP (%)	Tiempo (s)	Sol	Mejora (%)	Tiempo (s)	Sol	GAP (%)	Mejora (%)	Tiempo (s)
1	1P100C-1H1TW200K	–	–	3601	23160	–	4	22811	0.21	–	3600
2	1P100C-1H2TW200K	–	–	3601	24673	–	5	24131	0	–	2983
3	1P100C-1H3TW200K	–	–	3602	22668	–	10	22668	2.15	–	3602
4	1P100C-3H1TW600K	–	–	4110	65652	–	39	65652	100	–	3603
5	1P100C-3H2TW200K	–	–	3603	65443	–	35	65443	100	–	3603
6	1P100C-3H3TW200K	–	–	3602	66840	–	35	66840	100	–	3603
7	1P100C-5H2TW200K	–	–	3607	115336	–	62	115336	100	–	3608
8	1P100C-5H3TW200K	–	–	3601	102902	–	51	102902	100	–	3608
9	3P100C-1H1TW200K	–	–	3678	22503	–	6	22138	0	–	1672
10	3P100C-1H2TW200K	–	–	3602	23544	–	7	23389	1.35	–	3603
11	3P100C-1H3TW200K	–	–	3602	21420	–	5	21178	0.32	–	3600
12	3P100C-3H1TW200K	–	–	5021	66860	–	100	66860	2.74	–	3928
13	3P100C-3H3TW200K	–	–	3602	66291	–	38	66291	1.05	–	3601
14	3P100C-5H2TW200K	–	–	3600	107777	–	68	107777	100	–	3604
15	5P100C-1H2TW200K	–	–	3601	20904	–	10	20560	0.26	–	3602
16	5P100C-1H3TW200K	–	–	4925	19500	–	24	19500	1.98	–	3608
17	5P100C-3H2TW200K	–	–	3601	62683	–	69	62683	100	–	3602
18	5P100C-3H3TW200K	–	–	3603	58358	–	64	58358	100	–	3603
19	7P100C-1H1TW200K	–	–	3809	21881	–	18	21437	0	–	2782
20	7P100C-1H2TW200K	–	–	3669	21547	–	10	21490	1.64	–	3622
21	7P100C-1H3TW200K	–	–	3600	22656	–	7	22324	0.20	–	3600

Tabla A.7: Experimentación: Problema multi-producto – 100C

No.	Clase: 200C Instancia	CPLEX – Modelo MPD			Heurística MPD-H			CPLEX – Modelo MPD-SC			
		Sol	GAP (%)	Tiempo (s)	Sol	Mejora (%)	Tiempo (s)	Sol	GAP (%)	Mejora (%)	Tiempo (s)
1	1P200C-1H1TW400K	–	–	4693	41664	–	54	41664	3.21	–	3823
2	1P200C-1H2TW400K	–	–	4050	44244	–	63	44244	3.17	–	3726
3	1P200C-1H3TW400K	–	–	3606	44532	–	66	44532	3.03	–	3601
4	3P200C-1H3TW400K	–	–	3606	45912	–	92	45912	2.05	–	3602
5	5P200C-1H3TW400K	–	–	3704	42672	–	213	42672	1.84	–	4081
6	7P200C-1H3TW400K	–	–	3652	44004	–	211	44004	2.20	–	3626

Tabla A.8: Experimentación: Problema multi-producto – 200C

APÉNDICE B

NOTAS COMPLEMENTARIAS

En este apéndice se detalla información complementaria la cual puede ayudar a un mejor entendimiento de los temas tratados a lo largo de la tesis.

B.1 PROBLEMA DE COBERTURA DE CONJUNTOS

El problema de cobertura de conjuntos conocido como *Set Covering Problem* (SCP), es un problema en el que dado un conjunto de elementos $\{1, 2, \dots, m\}$ (llamado universo), n conjuntos cuya unión comprende el universo y un costo c_j asociado a cada conjunto s_j . Se desea identificar los conjuntos con el menor costo, cuya unión genere como resultado el conjunto universo.

El SCP puede ser descrito formalmente de la siguiente manera. Sea A una matriz de 0-1 de $m \times n$, cuyos elementos a_{ij} toman el valor 1 si el conjunto j contiene al elemento i y sea C un vector de n elementos, de tal manera que el elemento c_j representa el costo asociado al conjunto j . Sea $M = \{1, 2, \dots, m\}$ y $N = \{1, 2, \dots, n\}$, además se asume que $c_j > 0$ para cada $j \in N$. El modelo matemático para el problema de cobertura de conjuntos se puede establecer de la siguiente manera:

Parámetros

a_{ij} Tomará el valor de 1 si el conjunto j contiene al elemento i , 0 en otro caso.

c_j El costo asociado al conjunto j .

Variables de decisión

$$x_j = \begin{cases} 1 & \text{Si el conjunto } j \text{ es seleccionado.} \\ 0 & \text{En otro caso.} \end{cases}$$

$$\text{Min } \sum_{j \in N} c_j x_j \quad (\text{B.1})$$

sujeto a:

$$\sum_{j \in N} a_{ij} x_j \geq 1 \quad \forall i \in M \quad (\text{B.2})$$

$$x_j \in \{0, 1\} \quad \forall j \in N \quad (\text{B.3})$$

La función objetivo (B.1) busca minimizar el costo de los conjuntos que son seleccionados, mientras que la restricción (B.2) asegura que los conjuntos seleccionados generen el universo. La ecuación (B.3) establece la naturaleza de las variables x como binarias.

B.2 HEURÍSTICAS Y METAHEURÍSTICAS

La metodología de solución heurística, comúnmente llamada solamente heurística, es un enfoque simple para resolver un problema, que no garantiza obtener resultados óptimos, pero provee una solución para objetivos inmediatos. Las heurísticas son estrategias derivadas de la experiencia y conocimiento del problema, cuyo objetivo principal es encontrar una solución factible al problema en un tiempo que se considere razonable.

Los métodos heurísticos son utilizados usualmente cuando es imposible o no es viable encontrar una solución óptima. Así mismo las heurísticas son utilizadas en algunas experimentaciones para proveer una solución inicial a metodologías exactas con el fin de acelerar el proceso de obtención de una solución óptima o satisfactoria.

Una metaheurística es aquel método heurístico de objetivo general, el cual fija un patrón abstracto, esperando resultados eficientes, otorgando libertad al usuario de realizar cambios sobre los parámetros y heurísticos internos para resolver un problema en específico.

Algunos procedimientos metaheurísticos que se encuentran en la literatura son algoritmos genéticos, recocido simulado, búsqueda tabú, optimización por colonia de hormigas, GRASP, algoritmo de cúmulo de partículas, búsqueda dispersa, búsqueda de grandes vecindarios, etc.

Las metaheurísticas se utilizan comúnmente para resolver problemas en los que no se dispone de un algoritmo de solución eficiente para el mismo. Para el problema de estudio, en esta tesis, se diseñó e implementó una metaheurística tipo GRASP, específicamente un RGRASP (Reactive Greedy Randomized Adaptive Search Procedure).

BIBLIOGRAFÍA

- Behrouz Afshar-Nadjafi and Alireza Afshar-Nadjafi. A constructive heuristic for time-dependent multi-depot vehicle routing problem with time-windows and heterogeneous fleet. *Journal of King Saud University-Engineering Sciences*, 2014.
- Claudia Archetti, Nicola Bianchessi, and M Grazia Speranza. A branch-price-and-cut algorithm for the commodity constrained split delivery vehicle routing problem. *Computers & Operations Research*, 64:1–10, 2015.
- Nabila Azi, Michel Gendreau, and Jean-Yves Potvin. An exact algorithm for a vehicle routing problem with time windows and multiple use of vehicles. *European Journal of Operational Research*, 202(3):756–763, 2010.
- Farah Belmecheri, Christian Prins, Farouk Yalaoui, and Lionel Amodeo. Particle swarm optimization algorithm for a vehicle routing problem with heterogeneous fleet, mixed backhauls, and time windows. *Journal of intelligent manufacturing*, 24(4):775–789, 2013.
- Andrea Bettinelli, Alberto Ceselli, and Giovanni Righini. A branch-and-cut-and-price algorithm for the multi-depot heterogeneous vehicle routing problem with time windows. *Transportation Research Part C: Emerging Technologies*, 19(5):723–740, 2011.
- Andrea Bettinelli, Alberto Ceselli, and Giovanni Righini. A branch-and-price algorithm for the multi-depot heterogeneous-fleet pickup and delivery problem with soft time windows. *Mathematical Programming Computation*, 6(2):171–197, 2014.

- Marie-Claude Bolduc, Jacques Renaud, Fayez Boctor, and Gilbert Laporte. A perturbation metaheuristic for the vehicle routing problem with private fleet and common carriers. *Journal of the Operational Research Society*, 59(6):776–787, 2008.
- Mourad Boudia, Mohamed Aly Ould Louly, and Christian Prins. A reactive grasp and path relinking for a combined production–distribution problem. *Computers & Operations Research*, 34(11):3402–3419, 2007.
- Jose Brandao and Alan Mercer. A tabu search algorithm for the multi-trip vehicle routing and scheduling problem. *European journal of operational research*, 100(1):180–191, 1997.
- Diego Cattaruzza, Nabil Absi, Dominique Feillet, and Daniele Vigo. An iterated local search for the multi-commodity multi-trip vehicle routing problem with time windows. *Computers & Operations Research*, 51:257–267, 2014.
- Sara Ceschia, Luca Di Gaspero, and Andrea Schaerf. Tabu search techniques for the heterogeneous vehicle routing problem with time windows and carrier-dependent costs. *Journal of Scheduling*, 14(6):601–615, 2011.
- George B Dantzig and John H Ramser. The truck dispatching problem. *Management science*, 6(1):80–91, 1959.
- Iman Dayarian, Teodor Gabriel Crainic, Michel Gendreau, and Walter Rei. A column generation approach for a multi-attribute vehicle routing problem. *European Journal of Operational Research*, 241(3):888–906, 2015.
- Rodolfo Dondo and Jaime Cerdá. A cluster-based optimization approach for the multi-depot heterogeneous fleet vehicle routing problem with time windows. *European Journal of Operational Research*, 176(3):1478–1507, 2007.
- Thomas A Feo and Jonathan F Bard. Flight scheduling and maintenance base planning. *Management Science*, 35(12):1415–1432, 1989.

- Thomas A Feo and Mauricio GC Resende. Greedy randomized adaptive search procedures. *Journal of global optimization*, 6(2):109–133, 1995.
- Luciana B Gonçalves, Luiz S Ochi, and Simone L Martins. A grasp with adaptive memory for a period vehicle routing problem. In *Computational Intelligence for Modelling, Control and Automation, 2005 and International Conference on Intelligent Agents, Web Technologies and Internet Commerce, International Conference on*, volume 1, pages 721–727. IEEE, 2005.
- Hipólito Hernández-Pérez, Inmaculada Rodríguez-Martín, and Juan-José Salazar-González. A hybrid heuristic approach for the multi-commodity pickup-and-delivery traveling salesman problem. *European Journal of Operational Research*, 2015.
- J Jair, Carlos D Paternina-Arboleda, Victor Cantillo, and Jairo R Montoya-Torres. A two-phormone trail ant colony system-tabu search approach for the heterogeneous vehicle routing problem with time windows and multiple products. *Journal of Heuristics*, 19(2):233–252, 2013.
- Jun Jiang, Kien Ming Ng, Kim Leng Poh, and Kwong Meng Teo. Vehicle routing problem with a heterogeneous fleet and time windows. *Expert Systems with Applications*, 41(8):3748–3760, 2014.
- Angel A Juan, Jarrod Goentzel, and Tolga Bektaş. Routing fleets with multiple driving ranges: Is it possible to use greener fleet configurations? *Applied Soft Computing*, 21:84–94, 2014.
- Philip Kilby and Tommaso Urli. Fleet design optimisation from historical data using constraint programming and large neighbourhood search. *Constraints*, 21(1):2–21, 2016.
- Manolis N Kritikos and George Ioannou. The heterogeneous fleet vehicle routing problem with overloads and time windows. *International Journal of Production Economics*, 144(1):68–75, 2013.

- Yong-Ju Kwon, Young-Jae Choi, and Dong-Ho Lee. Heterogeneous fixed fleet vehicle routing considering carbon emission. *Transportation Research Part D: Transport and Environment*, 23:81–89, 2013.
- Rahma Lahyani, Leandro C Coelho, Mahdi Khemakhem, Gilbert Laporte, and Frédéric Semet. A multi-compartment vehicle routing problem arising in the collection of olive oil in tunisia. *Omega*, 51:1–10, 2015.
- Michela Lai, Teodor Gabriel Crainic, Massimo Di Francesco, and Paola Zuddas. An heuristic search for the routing of heterogeneous trucks with single and double container loads. *Transportation Research Part E: Logistics and Transportation Review*, 56:108–118, 2013.
- David Levy, Kaarthik Sundar, and Sivakumar Rathinam. Heuristics for routing heterogeneous unmanned vehicles with fuel constraints. *Mathematical Problems in Engineering*, 2014, 2014.
- Shuguang Liu. A hybrid population heuristic for the heterogeneous vehicle routing problems. *Transportation Research Part E: Logistics and Transportation Review*, 54:67–78, 2013.
- Dražen Popović, Milorad Vidović, and Gordana Radivojević. Variable neighborhood search heuristic for the inventory routing problem in fuel delivery. *Expert Systems with Applications*, 39(18):13390–13398, 2012.
- Marcelo Prais and Celso C Ribeiro. Reactive grasp: An application to a matrix decomposition problem in tdma traffic assignment. *INFORMS Journal on Computing*, 12(3):164–176, 2000.
- Yuan Qu and Jonathan F Bard. The heterogeneous pickup and delivery problem with configurable vehicle capacity. *Transportation Research Part C: Emerging Technologies*, 32:1–20, 2013.
- N Ramkumar, P Subramanian, TT Narendran, and K Ganesh. Mixed integer linear

- programming model for multi-commodity multi-depot inventory routing problem. *Opsearch*, 49(4):413–429, 2012.
- Said Salhi, Arif Imran, and Niaz A Wassan. The multi-depot vehicle routing problem with heterogeneous vehicle fleet: Formulation and a variable neighborhood search implementation. *Computers & Operations Research*, 52:315–325, 2014.
- Michel Povlovitsch Seixas and André Bergsten Mendes. Column generation for a multitrip vehicle routing problem with time windows, driver work hours, and heterogeneous fleet. *Mathematical Problems in Engineering*, 2013, 2013.
- R Tavakkoli-Moghaddam, N Safaei, MMO Kah, and M Rabbani. A new capacitated vehicle routing problem with split service for minimizing fleet cost by simulated annealing. *Journal of the Franklin Institute*, 344(5):406–425, 2007.
- Thibaut Vidal, Teodor Gabriel Crainic, Michel Gendreau, Nadia Lahrichi, and Walter Rei. A hybrid genetic algorithm for multidepot and periodic vehicle routing problems. *Operations Research*, 60(3):611–624, 2012.
- Thibaut Vidal, Teodor Gabriel Crainic, Michel Gendreau, and Christian Prins. A unified solution framework for multi-attribute vehicle routing problems. *European Journal of Operational Research*, 234(3):658–673, 2014.
- Qian Wang, Qingkai Ji, and Chun-Hung Chiu. Optimal routing for heterogeneous fixed fleets of multicompartment vehicles. *Mathematical Problems in Engineering*, 2014, 2014.
- Yingcheng Xu and Wei Jiang. An improved variable neighborhood search algorithm for multi depot heterogeneous vehicle routing problem based on hybrid operators. *International Journal of Control and Automation*, 7(3):299–316, 2014.
- Yingcheng Xu, Li Wang, and Yuexiang Yang. A new variable neighborhood search algorithm for the multi depot heterogeneous vehicle routing problem with time windows. *Electronic Notes in Discrete Mathematics*, 39:289–296, 2012.

Baozhen Yao, Bin Yu, Ping Hu, Junjie Gao, and Mingheng Zhang. An improved particle swarm optimization for carton heterogeneous vehicle routing problem with a collection depot. *Annals of Operations Research*, pages 1–18, 2014.

Hugo Tsugunobu Yoshida Yoshizaki et al. Scatter search for a real-life heterogeneous fleet vehicle routing problem with time windows and split deliveries in brazil. *European Journal of Operational Research*, 199(3):750–758, 2009.

FICHA AUTOBIOGRÁFICA

Roberto Cantu Funes

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

Universidad Autónoma de Nuevo León

Facultad de Ingeniería Mecánica y Eléctrica

Tesis:

OPTIMIZACIÓN DE LA LOGÍSTICA DE TRANSPORTE Y DISTRIBUCIÓN DE PRODUCTOS

Nací el 24 de mayo de 1992 en Monterrey, Nuevo León, primogénito de Laura Alicia Funes García y Roberto Cantu Cavazos.

Concluí mi educación básica en el año 2007, iniciando el mismo año con mi bachillerato general en la Preparatoria No. 2, de la Universidad Autónoma de Nuevo León. Durante el último semestre de mis estudios en la preparatoria, se me impartió la clase Información Profesional y Laboral en la cual decidí estudiar la carrera de Licenciatura en Ciencias Computacionales en la Facultad de Ciencias Físico Matemáticas.

En el año 2009 ingresé a la Facultad de Ciencias Físico Matemáticas de la Universidad Autónoma de Nuevo León. En el séptimo semestre el año 2012 cursé la

materia de investigación de operaciones, la cual atrajo mi atención. Posteriormente realicé mi servicio social como asesor en el área de ciencias computacionales en la Facultad de Ciencias Físico Matemáticas. Simultáneamente apliqué a una beca para certificarme como Asociado en el desarrollo - ABAP con SAP NetWeaver 7.02, en el Centro para el Desarrollo de la Industria del Software, la cual me fue otorgada, concluyendo así mi certificación y servicio social en junio del 2013.

Posteriormente trabajé como Consultor en Inteligencia de Negocios Junior, en la empresa HITSS, el período de Julio del 2013 a Enero del 2014, desempeñando labores de estimación de diseño y desarrollo de proyectos, diseño de almacenes de datos y elaboración de software en java.

En Enero del 2014 ingresé como estudiante de maestría al Programa de Posgrado en Ingeniería de Sistemas de la Facultad de Ingeniería Mecánica y Eléctricade la Universidad Autónoma de Nuevo León.

Durante mis estudios de posgrado he trabajado en conjunto con mi supervisora la Dra. M. Angélica Salazar Aguilar, quien me oriento y impulso a congresos nacionales e internacionales. Así mismo tuve la oportunidad de impartir cursos y talleres en la Facultad de Ingeniería Mecánica y Eléctrica. Posteriormente se me presentó la oportunidad de realizar una estancia de investigación en la Universidad LAVAL, en Quebec, Canada. Durante dicha estancia de investigación trabajé en conjunto con el Dr. Leandro C. Coelho, realizando una investigación en el área de ruteo de vehículos.