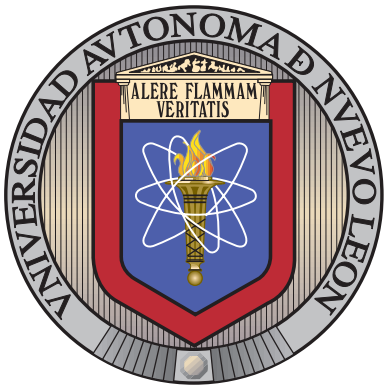
Universidad Autónoma de Nuevo León

Facultad de Ingeniería Mecánica y Eléctrica

División de Estudios de Posgrado



Una metaheurística para un problema de carga y descarga en la repartición de bebidas embotelladas

por

Andrés Castrillón Escobar

en opción al grado de

Maestría en Ciencias en Ingeniería de Sistemas

San Nicolás de los Garza, Nuevo León Marzo 2013

Universidad Autónoma de Nuevo León

Facultad de Ingeniería Mecánica y Eléctrica

División de Estudios de Posgrado

Una metaheurística para un problema de carga y descarga en la repartición de bebidas embotelladas

por

Andrés Castrillón Escobar

en opción al grado de

Maestría en Ciencias en Ingeniería de Sistemas

San Nicolás de los Garza, Nuevo León Marzo 2013

Universidad Autónoma de Nuevo León

Facultad de Ingeniería Mecánica y Eléctrica

División de Estudios de Posgrado

Los miembros del Comité de Tesis recomendamos que la Tesis «Una metaheurística para un problema de carga y descarga en la repartición de bebidas embotelladas», realizada por el alumno Andrés Castrillón Escobar, con número de matrícula 1434914, sea aceptada para su defensa como opción al grado de Maestría en Ciencias en Ingeniería de Sistemas.

El Comité de Tesis

Dr. Roger Z. Ríos Mercado
Director

Dra. Ada M. Álvarez Socarrás
Revisor

Dr. J. Fabián López Pérez
Revisor

Vo. Bo.

Dr. Moisés Hinojosa Rivera
Subdirector
División de Estudios de Posgrado

San Nicolás de los Garza, Nuevo León, Marzo 2013

Les dedico este trabajo a aquellas personas que han dado todo por mí y siempre de una manera incondicional me han apoyado en el transcurso de mi vida.

Mis papás,

El Sr. Julio Castrillón Valdes *y*

la Sra. Delia Eugenia Escobar Oyervides*.*

Índice general

Agradecimientos xii

Resumen xiv

1. Introducción 1

1.1. Descripción del problema 1

1.2. Motivación 4

1.3. Objetivo y alcance 5

1.4. Justificación 6

1.5. Contribuciones 6

2. Antecedentes y Revisión de Literatura 8

2.1. Problemas de ruteo de vehículos 9

2.2. Problemas de carga y descarga 11

2.3. Problema de estudio 16

2.4. Literatura relacionada con el problema de carga y descarga 17

3. Descripción del Problema y Modelación 22

3.1. Descripción del problema 22

3.2. Consideraciones y supuestos 25

3.3. Modelación matemática 26

3.4. Formulación de modelos alternativos 35

3.4.1. Modelo B 35

3.4.2. Modelo C 37

3.5. Resolución del Modelo A en forma exacta 43

3.6. Análisis de complejidad 49

4. Metodologia de Solución 50

4.1. Construcción de matriz de costos y tiempos 53

4.2. Generación de pedidos 54

4.3. Fase constructiva 56

4.3.1. Inicialización de vehículos 57

4.3.2. Asignar requerimientos a los vehículos 58

4.3.3. Devolver los vehículos a su lugar de origen 62

4.4. Fase de mejora 63

4.4.1. Convertir vehículos a dobles 64

4.4.2. Transferir entregas 65

4.4.3. Reparar andenes 68

5. Evaluación Empirica 71

5.1. Desarrollo experimental 72

5.1.1. Objetivos 72

5.1.2. Condiciones experimentales 72

5.1.3. Generación de instancias del problema 73

5.1.4. Características y parámetros de las instancias 76

5.1.5. Aspectos a evaluar del algoritmo 76

5.1.6. Diseño de experimentos 77

5.2. Análisis de resultados 78

5.2.1. Etapa 1: Calibración de parámetros de la heurística 78

5.2.2. Etapa 2: Verificar la calidad de las soluciones 89

5.2.3. Etapa 3: Comportamiento del algoritmo 95

5.2.4. Etapa 4: Comparación con caso de la industria 100

6. Conclusiones **102**

6.1. Conclusiones 102

6.2. Contribuciones 104

6.3. Recomendaciones y trabajo futuro 105

A. Análisis de Complejidad **115**

Índice de figuras

1.1. Configuración del vehículo sencillo o doble. 2

1.2. Representación gráfica de la carga de un vehículo con y sin entrepiso. 3

2.1. Esquema basico de un VRP. 9

2.2. Subclases del problema de carga y descarga. 11

3.1. Cumplimiento de la restricción de compartimientos en un vehículo sencillo. 34

3.2. Cumplimiento de la restricción de compartimientos en un vehículo doble. 34

3.3. Interacción entre el tiempo de ejecución y el número de nodos. 47

3.4. Interacción entre el tiempo de ejecución y el número de tráileres. . . . 47

3.5. Interacción entre el tiempo de ejecución y el número de productos. . . 48

4.1. Construcción de matriz de costos. 53

4.2. Ejemplo de una ruta de un vehículo. 57

4.3. Conversión de dos vehículos sencillos a uno doble. 65

4.4. Transferir entregas de un vehículo a otro. 67

4.5. Reparar factibilidad por uso de andenes. 69

4.6. Criterio para reparar factibilidad por uso de andenes. 69

5.1. Desempeño del algoritmo en cuanto al número de éxitos como función de α y β. 80

5.2. Desempeño del algoritmo en cuanto a calidad de solución como función de α y β. 82

5.3. Desempeño del algoritmo en cuanto a desviación de la solución enteramente miope ($\alpha = 0$) como función de α y β. 84

5.4. Desempeño del algoritmo en cuanto a calidad de solución como función de $\Delta\beta$. 86

5.5. Porcentaje de veces donde se obtuvo el mejor valor de la función objetivo. 87

5.6. Valoración del intervalo de desviación relativa como función del número de iteraciones. 89

5.7. Tiempos de ejecución para resolver los Modelos A y C. 92

5.8. IDR del Modelo A y la Heurística con respecto al Modelo C. 93

5.9. Rutas obtenidas por ambas metodologías. 101

Índice de tablas

2.1. Características de los diferentes tipos de VRP y PDP. 12

2.2. Acrónimos para diferentes tipos de problemas del VRP y PDP. 15

2.3. Clasificación de trabajos sobre problemas de carga y descarga. 19

2.4. Abreviaciones de los diferentes métodos de solución. 19

3.1. Cantidad de variables en función a la cantidad de nodos y tráileres para $|P| = 2$. 44

3.2. Número de restricciones en función a los nodos y a los tráileres para $|P| = 2$. 44

3.3. Resultados del análisis de varianza. 45

5.1. Comparación de la heurística y los Modelos A y C en instancias pequeñas. 91

5.2. Resultados del banco de instancias. 94

5.3. Porcentaje de iteraciones factibles. 96

5.4. Porcentaje de soluciones factibles. 96

5.5. Tiempo promedio (en segundos) al ejecutar mil iteraciones. 97

5.6. Porcentajes del tiempo de ejecución de los diferentes componentes. . . 98

5.7. Porcentaje de mejora al implementar vecindario: Convertir Vehículos a Dobles. 99

5.8. Porcentaje de mejora al implementar vecindario: Transferir Entregas. 100

5.9. Resultados obtenidos para la resolución del caso de la industria. . . . 101

Agradecimientos

Quisiera agradecer de una manera especial al Dr. Roger Z. Ríos por darme la oportunidad de trabajar con él y ser mi asesor, al guiarme en mi desarrollo académico por el esfuerzo y el apoyo que me dedicó y la paciencia con la que lo hizo, por ser una excelente persona ya que sin él no hubiera sido posible el término de este trabajo.

Agradezco al equipo de profesores de PISIS por brindarme la oportunidad de realizar mis estudios en esta facultad, especialmente a la Dra. Ada M. Álvarez por haber aceptado y participado como parte de mi comité de tesis. A la Universidad Autónoma de Nuevo León y a la Facultad de Ingeniería Mecánica y Eléctrica por la beca de inscripción y de colegiatura respectivamente.

También agradezco al Dr. Fabián López por a ver aceptado formar parte de mi comité de tesis, darle seguimiento y retroalimentación al estudio realizado además de ayudarnos en la realización del mismo.

También agradezco al Consejo Nacional de Ciencia y Tecnología (CONACyT) por la beca de manutención y el proyecto de investigación CB-2005-01-48499-Y, con las cuales fue posible la realización de mis estudios de maestría.

También agradezco al Programa de Apoyo a la Investigación Científica y Tecnología (PAICyT) por el apoyo económico a través del proyecto IT511-10, la cual contribuyó para mi formación y desarrollo académico.

También agradezco de manera especial al Profesor Jonathan Bard, catedrático de la Universidad de Texas en Austin, por haberme recibido y apoyado en una

estancia de investigación realizada en dicha universidad.

También agradezco de manera especial al equipo de Dead Optimizers Society (DOS) especialmente a mi compañera Mónica Elizondo, por apoyo y consejo a lo largo de mis estudios de maestría.

A mis papás que siempre me tuvieron presente en sus oraciones y siempre me han brindado su apoyo incondicional a lo largo de toda mi vida, a mis hermanos y amigos que siempre han estado presente para apoyarme.

Andrés Castrillón Escobar

San Nicolás de los Garza, Nuevo León, Marzo 2013

Resumen

Andrés Castrillón Escobar.

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

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

Título del estudio:

Una metaheurística para un problema de carga y descarga en la repartición de bebidas embotelladas

Número de páginas: 119.

El presente trabajo se enfoca en un problema de carga y descarga de productos proveniente de una empresa de bebidas ubicada en la ciudad de Monterrey, Nuevo León. La empresa desea encontrar la mejor configuración de rutas posible de tal modo que se minimicen los costos de distribución tanto los costos que se incurren por el uso de los vehículos como los costos por alquilar los vehículos; y que satisfaga la demanda de sus centros de distribución. Lo que hace diferente a este problema son las características particulares que lo distinguen como por ejemplo la decisión de la configuración de los vehículos y la decisión de cómo acomodar en cada tráiler los diferentes tipos de productos para evitar aplastarlos, entre otros. Debido a su

complejidad este tipo de problemas no se puede resolver con métodos exactos convencionales en un tiempo razonable para problemas de tamaño real.

Con base a lo anterior, se desarrolló un algoritmo basado en una metaheurística tipo GRASP. Esta técnica consiste en un proceso iterativo donde se construye una solución y posteriormente se mejora. No se tiene conocimiento de que exista algún trabajo en la literatura especializada que aborde un problema de carga y descarga con las características del trabajo de estudio; sin embargo, cabe aclarar que existen trabajos que abordan algunas de estas características.

El análisis de este problema, así como el estudio computacional que se realizó en este trabajo ha permitido lograr un mayor entendimiento de la problemática estudiada. La metodología desarrollada ha demostrado ser capaz de obtener soluciones de buena calidad en tiempos razonables para las instancias de tamaño real proporcionados por la industria.

OBJETIVOS Y MÉTODO DE ESTUDIO: El objetivo de esta tesis es de proponer un modelo matemático al problema de carga y descarga de producto así como una metodología de solución para poder resolver problemas de tamaño real encontrando de manera rápida e inteligente soluciones de buena calidad. Para esto se consideraron los siguientes objetivos particulares:

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

2. Proponer un modelo matemático que represente el problema expuesto, es decir, una abstracción matemática del problema, con la finalidad de determinar el tamaño máximo para el cual es posible encontrar soluciones exactas con métodos convencionales.

3. Desarrollar e implementar una metodología heurística que permita encontrar soluciones aproximadas al problema de carga de descarga para instancias de

tamaño real.

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

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

- Se logró un entendimiento detallado del problema abordado.
- Se realizó la formulación de varios modelos matemáticos que definen completamente al problema estudiado y algunas de sus variantes.
- Se demostró la complejidad computacional del problema.
- Se realizaron pruebas sobre el modelo matemático para determinar el tamaño máximo al cual es posible resolver el problema de manera óptima.
- Como aportación principal de este trabajo se diseñó y construyó una heurística basada en GRASP para encontrar soluciones de buena calidad al problema presentado. Esta heurística incorpora una serie de procedimientos específicos para la construcción y mejora de una solución.
- Se realizó un estudio empírico detallado, el cual logró probar que la metodología de solución propuesta obtuvo soluciones que cumplen con los requisitos establecidos.
- Se logró crear una herramienta para resolver el problema real de la empresa, como resultado de los conocimientos adquiridos a lo largo de los estudios de maestría.

Firma del director: _____

Dr. Roger Z. Ríos Mercado

Capítulo 1

Introducción

En nuestros días las empresas buscan darle a los clientes una calidad de servicio cada vez mejor para ser altamente competitivos; sin embargo, en la mayoría de los casos es posible mejorar la manera de como realizan estos procesos y poder así ser altamente competitivos, de tal manera que se pueda reducir el uso de los bienes de la empresa y ser capaz de brindarles un mejor servicio a sus clientes.

El problema que se aborda en esta tesis proviene de una empresa embotelladora de bebidas ubicada en la ciudad de Monterrey, N.L, México. Esta empresa necesita distribuir sus productos a lo largo de varios centros de distribución. Por otro lado, estos centros de distribución también solicitan que se retire cierto producto o que se reubiquen en otros centros. Este problema debido a sus características se clasifica como un problema de ruteo de vehículos con carga y descarga (PDP, por sus siglas en inglés, *Pickup and Delivery Problem*) sujeto a condiciones adicionales.

1.1 Descripción del problema

El problema básicamente consiste en determinar un diseño de rutas para la transportación y entrega de productos desde cierto número de plantas de manufactura hasta los centros de distribución a través de una red, las cuales seguirían un conjunto de vehículos para el cumplimiento de la demanda de los centros de distribución. Éste debe hacerse de tal manera que la realización de las entregas se lleve a cabo con el menor costo posible y se satisfaga una serie de restricciones especificadas por la

empresa.

Se cuenta con un conjunto de tráileres ubicados inicialmente en las plantas y cada una de éstos cuenta con una cabina. Con ello se debe de tomar la decisión de formar la flotilla de vehículos que serán utilizados para realizar las entregas. Un vehículo puede estar compuesto por un tráiler (vehículo sencillo) o dos tráileres (vehículo doble), tal como se muestra en la Figura 1.1. El número de tráileres en una planta es un dato conocido y la capacidad de cada tráiler de igual modo es un dato conocido. A cada tipo de vehículo distinto le toma tiempos distintos de viajar de un punto a otro y por consiguiente el costo también está en función del tipo de vehículo que se esté utilizando.

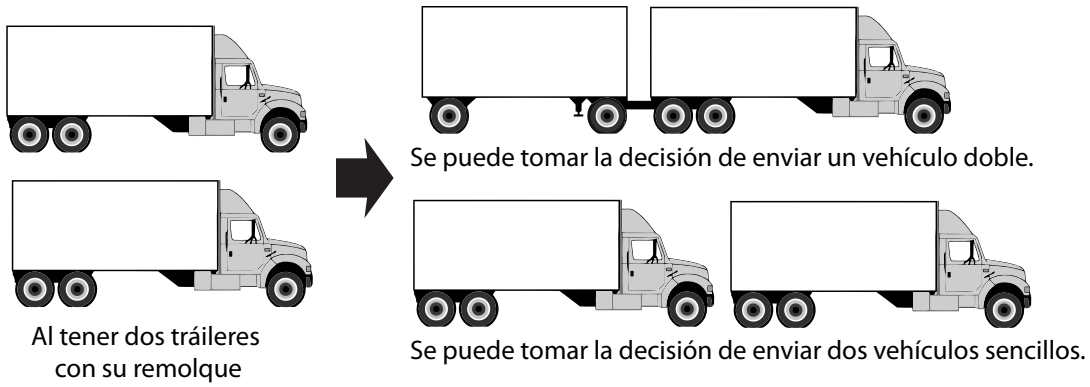


Figura 1.1: Configuración del vehículo sencillo o doble.

La capacidad de cada tráiler está dividida en dos partes (superior e inferior) como se puede observar en la Figura 1.2. A cada una de las partes se le conoce la cantidad de tarimas que tienen y también se conoce cuantas cajas caben por tarima dependiendo del tipo de producto.

Los productos que maneja la empresa son muchos pero es posible dividirlos en dos clases de productos: los productos de clase retornable (P_R) y los de clase no retornable (P_N). Esto último es importante ya que como se verá más adelante, cada tipo de producto puede ser puesto en los tráileres siguiendo unas reglas determinadas por la empresa.

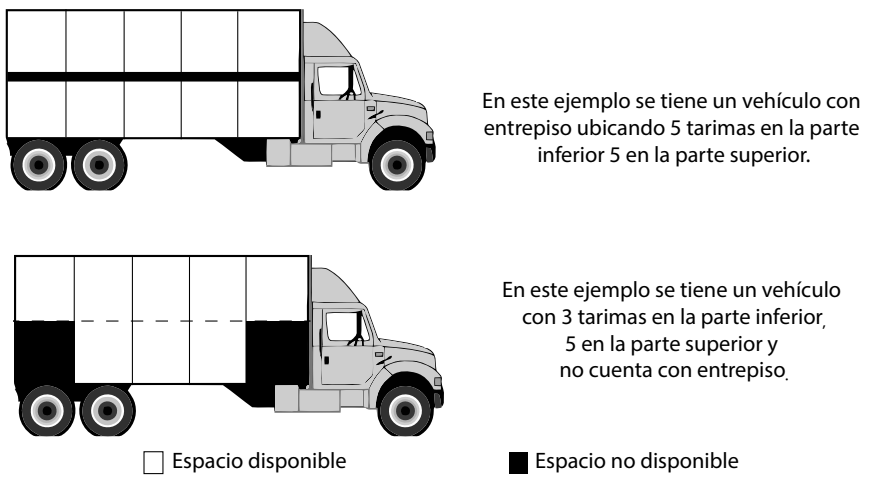


Figura 1.2: Representación gráfica de la carga de un vehículo con y sin entrepiso.

El objetivo consiste en determinar la mejor configuración de rutas posible que minimice el costo asociado al ruteo y el costo fijo por el uso de los vehículos. Dicha configuración debe cumplir una serie de lineamientos establecidos por la empresa para que pueda ser considerada como una configuración válida.

Por políticas de la empresa, los centros de distribución solo pueden ser atendidos a cierta hora del día y también los vehículos solo pueden ser usados en un cierto intervalo de tiempo corrido, esto último debido a que los vehículos son revisados diariamente en un chequeo de servicio y mantenimiento.

Además se pide que los vehículos no lleguen todos al mismo tiempo sino que vayan llegando en tiempos diferidos para así no saturar los andenes disponibles en las plantas y en los centros de distribución. Por políticas de la empresa, los vehículos inician su recorrido en una de las plantas disponibles y deben de regresar de nuevo a la misma planta al final del día.

No existe ninguna limitante de como se deben de realizar las entregas (como en el caso de PDP-LIFO o Dial-A-Ride [57]). Que por ejemplo el problema de ruteo de vehículos con carga y descarga con restricción LIFO, (PDP-LIFO, por sus siglas en inglés, *Pickup and Delivery Problem Last In First Out*) restringe que el último bien recolectado debe de entregarse antes que cualquier otro recolectado anteriormente.

Mientras que el problema con entrega inmediata (DARP, por sus siglas en inglés, *Dial A Ride Problem*) restringe que él se debe de entregar un bien inmediatamente después de ser cargado. Los pedidos se pueden dividir en varias entregas y a su vez varios vehículos pueden visitar varias veces a los centros de distribución si esto fuese necesario. El horizonte de planeación es diario de modo que la ruta de cualquier vehículo debe de ser realizada en menos de 24 horas cumpliendo las ventanas de tiempo.

Para que los productos puedan ser cargados en los vehículos sin que éstos sufran daños en la transportación es importante cumplir con los siguientes criterios establecidos por la empresa:

- Si el tráiler cuenta con entrepiso: Se puede poner tanto el producto de clase P_R como el producto de clase P_N en cualquier parte del tráiler.
- Si el tráiler no cuenta con entrepiso: Si se coloca en la parte inferior producto de clase P_R, es posible poner tanto producto tipo P_R o P_N en la parte de superior del tráiler. En cambio si se coloca producto de clase P_N en la parte inferior, no es posible poner nada encima.

Nota: El producto de clase P_R viene en cajas plásticas, lo cual permite que se le puedan poner cajas encima sin que se maltrate el producto.

Con todas las características mencionadas anteriormente, el problema de estudio forma parte de los problemas conocidos en la literatura como problemas de ruteo de vehículos con carga y descarga.

1.2 Motivación

Los problemas de ruteo de vehículos con carga y descarga son problemas de mucho interés, sin embargo en la literatura se observa una gran cantidad de variantes lo cual impide que sea posible el desarrollar una metodología única de solución,

por lo que se desarrollan algoritmos especializados para cada problema de estudio. En algunos casos es necesario construir herramientas especializadas para resolver problemas específicos debido a la complejidad y características de los mismos. Se llevó a cabo una revisión literaria de los trabajos relacionados al ruteo de vehículos el cual será expuesta en el Capítulo 2. Con base a esta revisión literaria no se tiene conocimiento de algún otro problema que contenga simultáneamente todas las características y restricciones presentadas anteriormente.

Se sabe además que debido a la naturaleza del problema, es posible formular este problema como un problema de programación lineal entera mixta, que puede ser resuelto con métodos de optimización disponibles en el mercado. Sin embargo, el tamaño de las instancias reales es tal que los algoritmos exactos no pueden dar soluciones óptimas en un período razonable de tiempo. Es por ello que para obtener soluciones satisfactorias para instancias reales es necesario una herramienta computacional que proporcione soluciones de alta calidad obtenidas en un tiempo razonable.

1.3 Objetivo y alcance

El objetivo de esta tesis es de proponer un modelo matemático para el problema que se está abordando así como también una metodología de solución para poder resolver problemas de escala real, que sea capaz de encontrar soluciones de buena calidad de manera rápida e inteligente. Para esto se consideran los siguientes objetivos particulares:

1. Proponer un modelo matemático del problema, con el fin de encontrar soluciones exactas al problema y encontrar el tamaño máximo al que es posible encontrar soluciones exactas.

2. Desarrollar e implementar una metodología heurística que permita encontrar soluciones al problema de carga de descarga.

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

4. Desarrollar una metodología la cual mejore los resultados obtenidos por la empresa.

El alcance de este trabajo es el de proporcionar una técnica heurística basada en GRASP para dar solución al problema de la empresa definido en la Sección 3.1. Esta técnica puede ser útil para resolver problemas que compartan todas o algunas de las características planteadas en la definición del problema.

1.4 Justificación

Con base al conocimiento adquirido que se obtuvo a través de la revisión literaria, no se tiene conocimiento de algún estudio donde se aborden todas las características del problema en forma simultanea, por lo que el desarrollo de esta tesis presentaría una nueva variante del problema de ruteo de vehículos.

Además el desarrollo de una metodología eficiente para la resolución de este problema sería un caso de éxito aplicado en la industria donde se implementen inteligentemente técnicas de optimización para la resolución de problemas de ruteo de vehículos con carga y descarga.

1.5 Contribuciones

En este trabajo se diseñó y construyó una heurística basada en GRASP, la cual resuelve el problema de carga y descarga con las restricciones propias del problema estudiado, al incorporar varios procedimientos clave para tratar al problema como un problema de carga y descarga con emparejamiento. Además se logró crear

una herramienta que logró obtener mejores resultados para el problema real de la industria en los casos de estudio analizados.

Capítulo 2

Antecedentes y Revisión de Literatura

Los problemas de ruteo de vehículos con carga y descarga (PDP, por sus siglas en inglés, *Pickup and Delivery Problem*) han sido muy estudiados por personas especializadas en el área de investigación de operaciones tanto a nivel académico como de aplicación industrial. Existen una gran cantidad de modelos para resolver este tipo de problemas debido a la variedad de características particulares que pueden tener cada uno de los problemas a tratar. Este capítulo tiene por finalidad ofrecer un panorama general de las variantes que tienen este tipo de problemas y sus aplicaciones en el mundo real.

El resolver problemas de ruteo de vehículos (VRP, por sus siglas en inglés, *Vehicle Routing Problem*) y sus variantes relacionadas es un importante campo de la investigación de operaciones. Tener herramientas para la toma de decisiones en la programación y planificación de rutas puede generar ahorros significativos en los costos de transportación. Típicamente se estiman ahorros entre el 5 % y el 20 % [73]. Por ejemplo, en las empresas de EUA, se estima que el costo asociado a la transportación de mercancías es de 47 billones de dólares al año y en Europa alrededor 168 billones de dólares, según King y Mast [36].

Este tipo de problemas es de una gran relevancia para las empresas, porque además es importante aclarar que dentro de los costos de distribución, aproximada-

mente el 50 % son costos logísticos, y si se habla de negocios relacionados con la comida o bebida pueden suponer hasta un 70 % del valor añadido del producto, según De Backer et al. [14] y Golden y Wasil [25].

2.1 Problemas de ruteo de vehículos

El ruteo de vehículos generalmente implica problemas que tienen que ver con la transportación de bienes o personas entre plantas y clientes, donde el objetivo es diseñar una programación óptima de uno o más vehículos para minimizar los costos de transportación.

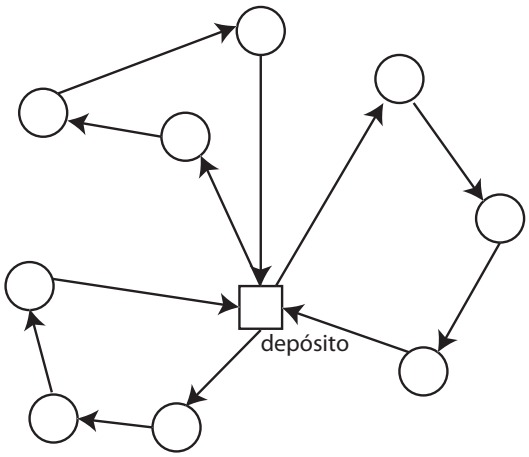


Figura 2.1: Esquema basico de un VRP.

En general el problema de ruteo de vehículos consiste en averiguar la mejor configuración de rutas de una flotilla de transporte para dar servicio a unos clientes distribuidos a través de una red, típicamente buscando minimizar costos o la cantidad de vehículos a utilizar. Dicho problema suele ser representado topológicamente por un grafo $G = (V, E)$, donde V representa el conjunto de clientes (nodos) que deben ser atendidos, mientras que E representa el conjunto de conecciones o caminos entre clientes, por los cuales los vehículos pueden realizar su ruta. Una solución a este tipo de problemas podría ser visualizado como se muestra en la Figura 2.1, el cual

cuenta con nueve clientes (identificado cada uno por un nodo circular) y un almacén (identificado por el nodo cuadrado) y la solución utiliza tres vehículos para realizar las entregas.

Por tratarse una generalización del famoso problema del agente viajero (TSP, por sus siglas en inglés, *Traveling Salesman Problem*), el VRP es un problema NP-duro [28]. Por mencionar algunas de las principales variantes del VRP tenemos:

- VRP capacitado (CVRP): En este tipo de problemas considera adicionalmente que los vehículos tienen una capacidad limitada para realizar las tareas.
- VRP con carga y descarga (VRPPD ó PDP): A diferencia del VRP clásico donde existe un único tipo de servicio (entrega), el VRP con carga y descarga cuenta con dos tipos de servicios: entrega y recolección.
- VRP con múltiples almacenes (MDVRP): En este tipo de problemas los vehículos están ubicados inicialmente en más de un almacén.
- VRP con carga y descarga simultánea (VRPSDP): En este problema es similar al problema de VRPPD con la diferencia de que es posible realizar ambos servicios (entrega y recolección) en una única visita a esta localización.
- VRP con entregas divididas (SDVRP): En este tipo de problemas considera la posibilidad de cumplir la demanda de una localización utilizando varios vehículos.
- VRP con vehículos heterogéneos (HF-VRP): En este problema es similar al problema de CVRP con la característica de que los vehículos pueden tener diferentes capacidades.
- VRP con flotilla mixta (MFVRP): En este tipo de problemas no solo es posible que los vehículos cuenten con diferentes capacidades (como el problema de HF-VRP) sino que es posible que cada uno cuente con distintos tiempos o costos de traslados o tengan alguna otra restricción adicional en función al tipo de vehículo.

- VRP con ventanas de tiempo (VRPTW): En este tipo de problemas se tiene restringido el tiempo en el cual es posible atender a un cliente.
- VRP estocástico (SVRP): En este tipo de problemas existe incertidumbre en uno o más de sus datos, típicamente la demanda, el tiempo de traslado y el costo en realizar una tarea.

2.2 Problemas de carga y descarga

Centrándonos en el problema del VRP con la variante carga y descarga (PDP). A diferencia del VRP donde todos los clientes requieren el mismo tiempo de servicio, el supuesto central en los PDP es que existen dos tipos diferentes de servicio que pueden ser realizados a un cliente, los cuales son: *carga* o *descarga*.

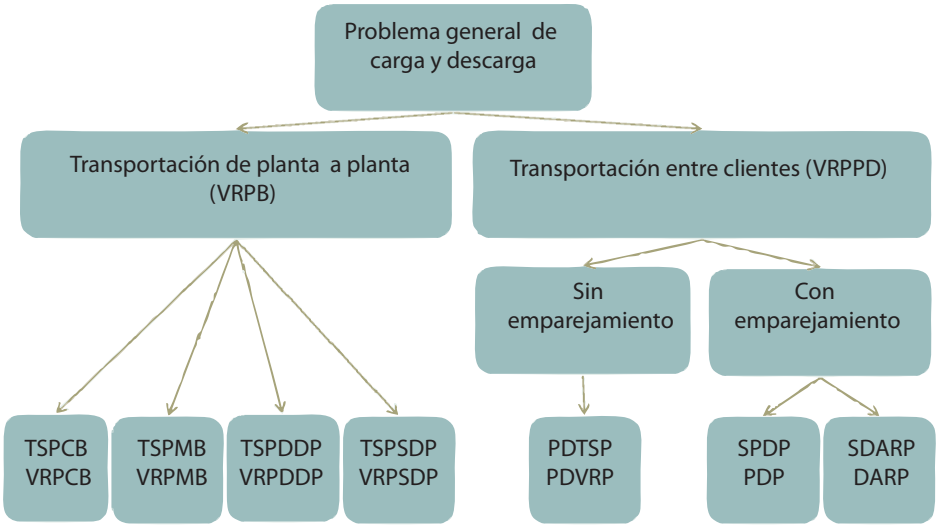


Diagrama de las variantes del problema general de carga y descarga.

VRP con carga de vuelta (VRPB)
VRP con consolidación de producto y carga de vuelta (VRPCB)
TSP con consolidación de producto y carga de vuelta (TSPCB)
VRP con combinación de entregas y cargas (VRPMB)
TSP con combinación de entregas y cargas (TSPMB)
VRP con entregas divididas (VRPDDP)
TSP con entregas divididas (TSPDDP)
VRP con carga y descarga simultánea (VRPSDP)
TSP con carga y descarga simultánea (TSPSDP)
VRP con carga y descarga (VRPPD)
PDP sin emparejamiento (PDVRP)
TSP con carga y descarga (PDTSP)
PDP con emparejamiento (PDP)
PDP con un solo vehículo (SPDP)
PDP con entrega inmediata (DARP)
DARP con un solo vehículo(SDARP)

Figura 2.2: Subclases del problema de carga y descarga.

Otra variante importante de los PDPs es cuando se conoce de antemano el origen y el destino de la carga y la descarga, respectivamente. En este caso se dice que es un problema con emparejamientos. Como se puede ver en la Figura 2.2, existen diversas variantes del problema de carga y descarga. Se menciona a continuación las diferentes clasificaciones en las que se puede dividir el problema de PDP realizado por Parragh, Doerner y Hartl [57].

	CVRP	DARP	DVRP	MFVRP	PDP	PDTSP	PDVRP	SDARP	SDVRP	SPDP	SVRP	TSPCB	TSPDDP	TSPMB	TSPSDP	VRPCB	VRPDDP	VRPMB	VRPSDP	MDVRP	OVRP	VRP
Un solo vehículo						•		•	•	•		•	•	•	•							
Múltiples vehículos	•	•	•	•	•		•				•					•	•	•	•	•	•	•
Una planta	•		•	•					•		•	•	•	•	•	•	•	•	•		•	•
Múltiples plantas				•							•	•	•	•	•	•	•	•	•	•		
Las órdenes son entre clientes		•			•	•	•	•		•												
Todas las entregas se cargan primero	○							○				•	•				•		○	○	○	
Solo se permite visitar a un cliente una vez	•		•						•						•				•	•	•	
Se permite visitar varias veces a un cliente													•				•					•
Se permite visitar varias veces a la planta													•				•					
La carga y descarga puede realizarse en la misma visita	•		•						•						•				•	•	•	
No se puede visitar otro cliente hasta realizar la descarga		•			•			•		•												
Punto de descarga asociado al punto de carga		•			•			•		•												
El cliente es de carga o descarga, pero no ambos											•	•		•		•		•				
El cliente es de carga o de descarga, incluso ambos			•						•				•		•		•		•	•	•	
Carga homogénea	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
Carga heterogénea	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
Vehículos homogéneos	○	○	○		○		○				○					○	○	○	○	○	○	○
Vehículos heterogéneos	○	○	○	•	○		○				○					○	○	○	○	○	○	○
Vehículos con compartimientos	○	○	○		○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
Decisión de configurar vehículo				•																		
Restricción de tiempo de viaje	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
Condición de tráfico durante el día	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
Preferencias del pasajero		○						○														
Entregas divididas	○	○	○	○	○	○	○	○	○	○	○	○	•	○	○	○	•	○	○	○	○	○
Ventanas de tiempo	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
Entrega con andenes	○	○	○	○	○	○	○		○		○	○	○	○	○	○	○	○	○	○	○	○
Estocástico											•											

• Restricción requerida ○ Restricción opcional

Tabla 2.1: Características de los diferentes tipos de VRP y PDP.

En la Tabla 2.1 se muestran algunos de los diferentes problemas que han sido abordados en la literatura relacionados a los problemas de ruteo de vehículos, junto con las características de cada uno de los trabajos que fueron estudiados para la

revisión literaria. Para ello primeramente se especifica una tabla donde fueron comparadas las diferentes características de cada uno de los problemas sobresalientes relacionados con el problema de carga y descarga. Adicionalmente, en la Tabla 2.2 se muestran los acrónimos para diversos problemas conocidos de ruteo de vehículos.

Un tipo de problema posee ciertas características específicas que lo identifican (expresado en la Tabla 2.1 como un punto obscuro) y ciertas características que pudiera o no poseer (expresado en la Tabla 2.1 como un punto claro). Varias de las características mostradas en esta tabla ya han sido descritas anteriormente por lo que a continuación se enlistan algunas de las características adicionales consideradas en esta tabla.

- Las órdenes son entre clientes: Esta característica se refiere a que se realizan entregas cuyos destinos son únicamente algunas localizaciones (clientes).
- Todas las entregas se cargan primero: Esta característica se refiere a que el bien a transportar solamente se puede cargar al inicio de la ruta.
- Punto de descarga asociado al punto de carga: Esta característica identifica a los problemas con emparejamiento, donde la decisión de origen y destino de los bienes a transportar es previamente establecido.
- Carga homogénea: Esta característica se refiere a que los bienes a transportar poseen las mismas características y no existe ninguna distinción entre un bien y otro.
- Carga heterogénea: Esta característica se refiere a que los bienes pueden poseer características particulares por lo que puede existir reglas específicas para la correcta transportación de los bienes. Un ejemplo de esta característica pudiera ser la de tener bienes que no pueden ser transportados simultáneamente, por ejemplo en el caso de la transportación de animales.
- Vehículos con compartimientos: Esta característica se refiere a que el vehículo posee características particulares y pudieran existir reglas adicionales para la

correcta transportación de los bienes. Un ejemplo de esta característica pudiera ser la de tener vehículos que solo permiten transportar cierto tipo de productos.

- Decisión de configurar el vehículo: Esta característica se refiere a tener que tomar la decisión de cómo enviar los vehículos para realizar las entregas. Esta característica por consiguiente implica también el poseer vehículos heterogéneos.

- Restricción de tiempo de viaje: Esta característica se refiere a que se tiene restringido a que los vehículos solo pueden ser usados solamente una cantidad de tiempo establecida. También suele establecerse como limitante la distancia de ruteo.

- Condición de tráfico durante el día: Esta característica se refiere a que los tiempos y costos de traslado no toman valor fijo, sino que están en función del momento en la cual se efectúa el traslado. Cabe aclarar que esta característica no implica que exista aleatoriedad en la información.

- Preferencias del pasajero: Esta característica se refiere a que se toma en consideración la preferencia del bien a transportar. Esta preferencia puede incluir el horario en el cual debe ser transportado, el vehículo en el cual será transportado, etc. Los bienes a transportar típicamente son personas.

- Entregas con andenes: Esta característica se refiere a que los clientes solo les es posible atender cierto número específico de vehículos simultáneamente.

Abreviaciones	Nombre
CVRP	VRP capacitado
DARP	PDP con entrega inmediata
DVRP	VRP con restricción de distancia
MDVRP	VRP con múltiples plantas
MFVRP	VRP con flotilla mixta
OVRP	VRP abierto
PDP	VRP con carga y descarga
PDTSP	TSP con carga y descarga
PDVRP	PDP sin emparejamientos
SDARP	DARP con un vehículo
SDVRP	VRP con dependencia de sitio
SPDP	PDP con un vehículo
SVRP	VRP estocástico
TSPCB	TSP con consolidación de producto y carga de vuelta
TSPDDP	TSP con entregas divididas
TSPMB	TSP con combinación de entregas y cargas
TSPSDP	TSP con carga y descarga simultanea
VRPCB	VRP con consolidación de producto y carga de vuelta
VRPDDP	VRP con entregas divididas
VRPMB	VRP con combinación de entregas y cargas
VRPSDP	VRP con carga y descarga simultanea
VRP	Problema de ruteo de vehículos
TSP	Problema del agente viajero

Tabla 2.2: Acrónimos para diferentes tipos de problemas del VRP y PDP.

2.3 Problema de estudio

El problema estudiado en esta tesis es un problema de la industria que ha sido modelado como un problema de carga y descarga de producto con las siguientes características:

- Múltiples almacenes: Los vehículos están ubicados inicialmente en más de una ubicación.
- Vehículos heterogéneos: Los vehículos tienen diferentes capacidades.
- Ventana de tiempos en los nodos y los vehículos: Los clientes tienen un estricto tiempo en el cual es posible atenderlos y lo vehículos tienen un horario en el cual es posible usarlos.
- Entregas divididas: Se es permitido que varios vehículos visiten a un cliente para satisfacer su demanda.
- Múltiples productos: Se considera que existen distintos tipos de productos y los clientes requieren cierto número de ellos.
- Carga y descarga simultánea: Cualquier vehículo le es posible entregar y recoger productos en una única visita a un cliente.
- Entregas con compartimientos: Los vehículos pueden transportar un producto si se satisface un criterio para no dañar los productos cargados previamente en el vehículo.
- Decisión de composición de flotilla: Se tiene la opción de enviar los vehículos como un vehículo con doble caja (vehículo doble) o un vehículo con una sola caja (vehículo sencillo).
- Restricción de andenes: Solo se permite un limitado número de vehículos para realizar de manera simultánea un servicio ya sea de carga o descarga.

- Soluciones tipo *Lasso*: Se es permitido que un vehículo pueda visitar varias veces a un cliente para satisfacer su demanda.

2.4 Literatura relacionada con el problema de carga y descarga

El problema de ruteo de vehículos junto con todas sus variantes ha sido ampliamente estudiado en la literatura. A continuación se muestra una tabla comparativa de algunos de los trabajos mas sobresalientes del problema de ruteo de vehículos o a alguna de sus variantes, donde se muestran las características propias del problema tratado en dichos trabajos junto con el método de solución que se propuso para resolverlo. En la Tabla 2.3 se muestra un resumen de algunos problemas de carga y descarga con características y restricciones similares a las del problema de estudio. En la primera columna se muestra el autor principal del trabajo citado. En la segunda columna se muestra el tipo de problema abordado. En las columnas 3-16 se indican las características del problema abordado en dichos trabajos. Finalmente la última columna muestra el método de solución desarrollado. Los métodos se definen en la Tabla 2.4.

Los artículos mencionados en esta tabla están ordenados alfabéticamente de acuerdo al nombre del artículo publicado. Posteriormente se mencionan algunos de los artículos relacionados con los problemas de carga y descarga que hacen una revisión de la literatura. Finalmente se agregan las características que han sido abordadas en el presente trabajo.

Actualmente no se tiene conocimiento que se haya abordado un problema con las características antes mencionadas, sin embargo hay una extensa cantidad de artículos que abordan problemas con algunas de estas características. A continuación se hará mención de algunos trabajos realizados donde se incluyen algunas de estas características antes mencionadas.

Artículo	Problema abordado	Todas las entregas se cargan primero	Carga homogénea	Carga heterogénea	Vehículos homogéneos	Vehículos heterogéneos	Vehículos con compartimientos	Decisión de configurar vehículo	Restricción de tiempo de viaje	Condición de tráfico durante el día	Preferencias del pasajero	Entregas divididas	Ventanas de tiempo	Entrega con andenes	Estocástico	Método de solución
Bard et al. [5]	VRP		✓		✓				✓				✓			RyA
Pisinger et al. [58]	VRP	✓	✓			✓			✓				✓			H-GV
Mladenović et al. [48]	PDTSP		✓		✓											H-GV
Ambrosini et al. [1]	PDP	✓	✓		✓											MH-G
Pankratz [54]	PDP	✓	✓		✓				✓				✓			AG
Li et al. [40]	PDP		✓		✓				✓				✓			MH-RS
Lu et al. [42]	PDP	✓	✓		✓				✓				✓			H
Prins [59]	VRP	✓	✓		✓											AG
Renaud et al. [61]	MFVRP	✓	✓			✓		✓	✓				✓			H
Archetti et al. [3]	VRPDDP		✓		✓							✓				MH-BT
Tang et al. [71]	VRPSDP		✓		✓				✓							MH-BT
Landrieu et al. [37]	SPDP	✓	✓		✓				✓				✓			MH-BT
Bent et al. [6]	PDP	✓	✓		✓				✓				✓			MH-RS
Ropke et al. [65]	PDP	✓	✓		✓				✓				✓			H-GV
Solomon [68]	VRP	✓			✓				✓				✓			H
Ropke et al. (2006) [64]	VRPDDP	✓	✓			✓			✓			✓	✓			H-GV
Burks [11]	PDP	✓		✓		✓			✓				✓			MH-BT
Créput et al. [13]	PDP	✓	✓		✓				✓				✓			AG
Lu et al. [41]	PDP	✓	✓			✓			✓				✓			RyA
Dong et al. [17]	PDP		✓		✓				✓				✓			MD
Haghani et al. [27]	VRP		✓		✓								✓			S
Ropke et al. [62]	PDP	✓	✓		✓				✓				✓			RyA
Mitrović et al. [46]	PDP	✓	✓			✓			✓				✓			H
Berbeglia et al. [8]	PDP		✓		✓				✓							H
Pacheco [52]	SPDP	✓	✓		✓											H
Hernández et al. [28]	PDTSP		✓		✓											H
Sáez et al. [67]	PDP	✓	✓		✓											AG
Srour et al. [69]	PDP	✓	✓		✓				✓				✓			RyA
Ropke et al. [63]	PDP	✓	✓		✓				✓				✓			RyA
Kerivin et al. [35]	SPDP	✓	✓		✓											RyA
Ralphs et al. [60]	CVRP		✓		✓											H
Dondo et al. [16]	PDP	✓	✓			✓			✓		✓		✓			H
Nowak et al. [51]	VRPDDP		✓		✓							✓				H
Pacheco et al. [53]	VRPSDP	✓	✓		✓				✓				✓			H
Yang et al. [77]	PDP	✓	✓		✓				✓				✓			S
Thangiah et al. [72]	PDP		✓		✓				✓			✓	✓			H
Dror et al. [18]	PDP	✓	✓		✓											H
Arbelaitz [2]	VRP		✓			✓			✓				✓			MH-RS
Carlsson et al. [12]	VRP		✓		✓											H
Nanry et al. [50]	PDP	✓	✓						✓				✓			MH-BT
Wassan et al. [75]	MFVRP		✓			✓			✓							MH-BT
Hernández et al. [29]	PDTSP	✓		✓	✓											RyA
Hosny et al. [32]	SPDP		✓		✓				✓				✓			AG
Nagy et al. [49]	VRPDDP	✓	✓		✓							✓				MH-BT
Baldacci et al. [4]	MFVRP		✓			✓										RyA
Derigs [15]	VRP	✓		✓	✓		✓									H
Wen et al. [76]	VRP		✓		✓				✓			✓		✓		MH-BT

Artículo	Problema abordado	Todas las entregas se cargan primero	Carga homogénea	Carga heterogénea	Vehículos homogéneos	Vehículos heterogéneos	Vehículos con compartimientos	Decisión de configurar vehículo	Restricción de tiempo de viaje	Condición de tráfico durante el día	Preferencias del pasajero	Entregas divididas	Ventanas de tiempo	Entrega con andenes	Estocástico	Método de solución
Mitrović [47]	PDP	✓	✓		✓				✓							H
Mendoza et al. [44]	SVRP			✓	✓		✓		✓						✓	AG
Bertsimas et al. [10]	SVRP		✓		✓										✓	PO
Bertsimas et al. [9]	SVRP		✓		✓										✓	PO
Mendoza et al. [45]	SVRP			✓	✓		✓								✓	H
Parragh et al. [57]																L
Hosny [31]																L
Berbeglia et al. [7]																L
Gendreau et al. [24]																L
Laporte [38]																L
Savelsbergh et al. [66]																L
Problema de estudio	PDP			✓		✓	✓	✓				✓	✓	✓		MH-G

Tabla 2.3: Clasificación de trabajos sobre problemas de carga y descarga.

Abreviaciones	Método de solución
RyA	Ramificación y acotamiento
H-GV	Heurística con búsqueda en grandes vecindarios
MH-G	Metaheurística basada en GRASP
AG	Algoritmos genético
MH-RS	Metaheurística basada en recocido simulado
MH-BT	Metaheurística basada en búsqueda tabú
H	Heurística
S	Simulación
PO	Política optima
MD	Método de descomposición
L	Repositorio de literatura

Tabla 2.4: Abreviaciones de los diferentes métodos de solución.

PDP con múltiples almacenes

Entre alguno de los trabajos relevantes, Carlsson et al. [12] desarrollaron un algoritmo de balanceo de carga para el problema del VRP con múltiples almacenes. Los autores presentaron el primer estudio min-max para dicho problema, en el cual fueron explorados algunos teoremas de las propiedades asintóticas de la solución óptima.

Por otro lado, Ropke y Pisinger [64] desarrollaron una heurística con una búsqueda local con un vecindario adaptativo para el PDPTW con múltiples almacenes. Su método es básicamente un proceso iterativo que destruye y construye parcialmente una solución. Se utilizaron diferentes movimientos de mejora los cuales eran guiados por un método de memoria adaptativa. Con esto fue posible encontrar soluciones de buena calidad en un tiempo razonable para instancias de hasta mil nodos.

VRP con compartimientos

Un problema de ruteo de vehículos con compartimientos fue estudiado por Derigs [15]. En éste se probaron diferentes heurísticas constructivas, de búsqueda local, con vecindarios a gran escala y diferentes metaheurísticas. En su análisis obtuvieron que los conceptos de diversificación e intensificación son esenciales para obtener soluciones de buena calidad. En este trabajo el vecindario de la búsqueda local consistía en el intercambio de las entregas en una ruta.

VRP con entregas divididas

Un problema de ruteo de vehículo con entregas divididas es abordado por Archetti, Speranza y Hertz [3]. En este trabajo se desarrolló un algoritmo que incorpora una búsqueda tabú, la cual hace uso de dos procedimientos, uno de ordenamiento de rutas y otro de seleccionar al mejor elemento de un vecindario el cual explora movimientos de intercambios tipo 2-OPT.

PDP con ventanas de tiempo

Ropke y Pisinger [64] estudian un problema de carga y descarga con ventanas de tiempo, donde utilizan una heurística de búsqueda local de vecindarios a gran escala que consiste en reasignar un pedido de un vehículo a otro vehículo. Cabe aclarar que este problema se maneja a base de pedidos, es decir, es un problema de PDP con emparejamientos, lo que significa que ya se tiene definido el origen y destino de cada pedido. Por otro lado existe una gran cantidad de literatura referente a los problemas de PDP con ventanas de tiempo. Para este tipo de problemas se han desarrollado diversas técnicas para su solución, como por ejemplo: métodos evolutivos [13], algoritmos de recocido simulado [74], algoritmos genéticos [34], algoritmo de colonia de hormigas [22], búsqueda tabú [50], por mencionar algunos.

PDP con carga y descarga simultanea

Tang y Diéguez [71] abordan un problema con cargas y descargas simultáneas. En su trabajo, los autores encontraron que la metaheurística de búsqueda tabú tuvo un excelente rendimiento para este VRP, comparado con otras técnicas de solución, produciendo soluciones de muy alta calidad en tiempos de cómputo razonables. En el trabajo que ellos realizaron definen un vecindario con cuatro tipos de movimientos tabú, tres de estos son movimientos fuera de la misma ruta (relocalización, intercambio y cruzamiento) y un movimiento más que se realiza dentro de la misma ruta (2-OPT).

PDP con múltiples productos

Finalmente referente a los problemas de PDP con múltiples productos, existen pocos trabajos realizados con estas características. Uno de ellos es el trabajo de Hernández-Pérez y Salazar-González [29] donde proponen técnicas de descomposición y algunas estrategias basadas en el procedimiento de poda y corte, o la tesis doctoral de Burks [11] que propone una búsqueda tabú donde el vecindario representa los posibles movimientos que se pueden hacer en una secuenciación de un vehículo, y un movimiento tabú consiste en fijar una configuración ya visitada.

Capítulo 3

Descripción del Problema y Modelación

El problema abordado puede ser modelado a través de la investigación de operaciones. En muchos casos el uso de modelos matemáticos se ha realizado de forma exitosa hasta su implementación.

Este capítulo tiene por objetivo mostrar la descripción detallada del modelo así como presentar la formulación propuesta del problema abordado en esta tesis. Primeramente se expresan las ventajas y las desventajas que ofrece desarrollar un modelo matemático para un problema. Posteriormente se presenta detalladamente el problema, los supuestos, las variables, parámetros y las notaciones usadas en el modelo. Después, se formula el modelo de programación entera mixta resultante y se explican detalladamente cada una de sus partes.

3.1 Descripción del problema

El problema consiste en determinar un diseño para la transportación y entrega de varios tipos de productos $p \in P$ a través de una red de distribución definida por un grafo $G = (V, E)$ donde cada vértice $i \in V$ representa una planta o un centro de distribución, y una arista $(i, j) \in E$ representa el camino (si es que existe) que conecta al nodo i con el j.

Cada nodo cuenta con estrictos horarios de servicio tal que a_i y b_i son los tiempos mínimo y máximo en el que se puede iniciar a atender al nodo i, respectivamente. Además, cada nodo puede requerir o disponer cierta cantidad de producto, de tal modo que n_{ip}^{+} expresa la cantidad de cajas del producto p que dispone el nodo i y n_{ip}^{-} expresa la cantidad de cajas del producto p que requiere el nodo i.

Para la descripción del problema se ha establecido que, una planta es aquel nodo que dispone con al menos un tipo de producto. Mientras que, un centro de distribución es aquel nodo que requiere al menos un tipo de producto. Cabe aclarar que es posible que un nodo pueda actuar simultáneamente como una planta y un centro de distribución.

Dado un conjunto total de tráileres T, cada planta cuenta con una flotilla de tráileres tal que T_i es el conjunto de tráileres que inicialmente se encuentran ubicados en el nodo i. Debido a políticas de la empresa los vehículos solamente pueden ser usados en un cierto intervalo de tiempo corrido, debido a que los vehículos son revisados diariamente en un chequeo de servicio y mantenimiento. Por lo que τ_t^l es el tiempo en el que el tráiler t puede iniciar su recorrido y τ_t^u es el tiempo en el que debe de finalizar su recorrido.

Cada tráiler además posee distinta capacidad y ésta está definida por dos elementos, de tal modo que c_t^{above} expresa la capacidad del tráiler t entendida como la cantidad de tarimas en la parte superior y c_t^{below} expresa la cantidad de tarimas del tráiler t en la parte inferior. Además, existen dos tipos de tráileres que maneja la empresa, los tráileres con entrepiso $T^e \subseteq T$ y los tráileres que no cuentan con entrepiso $T^w \subseteq T$. Esto último es de suma importancia ya que para la transportación de los productos es necesario tener en cuenta lo siguiente: A pesar de que existe una variedad considerable de productos que maneja la empresa, es posible dividir estos productos en dos clasificaciones, aquellos productos que son retornables $P_R \subseteq P$ y aquellos que no son retornables $P_N \subseteq P$. Para la correcta transportación de los productos es necesario cuidar que los productos no retornables no sean aplastados ya que estos productos son empacados con un material más liviano que los productos

retornables. Para proteger a estos productos se han definido los siguientes lineamientos para poder cargar producto a los tráileres:

- Si el tráiler cuenta con entrepiso: Se puede poner tanto el producto retornable como el producto no retornable en cualquier parte del tráiler.
- Si el tráiler no cuenta con entrepiso: Si se coloca en la parte inferior producto retornable, es posible poner tanto producto retornable como no retornable en la parte de superior del tráiler. En cambio si se coloca producto no retornable en la parte inferior, no es posible poner nada encima.

Los productos que maneja la empresa son muy variados y de diversos tamaños, sin embargo se sabe de antemano la cantidad de cajas de cualquier producto p que caben en una tarima definido como o_p.

Con los tráileres distribuidos por la red, es posible tomar la decisión de enviarlos como un vehículo sencillo, los cuales con aquellos que están formados por un único tráiler ó enviarlos como un vehículo doble, los cuales son aquellos que están formados por exactamente dos tráileres. Se cuenta con suficientes cabinas para poder enviar todos los tráileres como vehículos sencillos. Por otro lado la única condición para enviar dos tráileres como un vehículo doble es que ambos tráileres estén ubicados inicialmente en la misma planta.

Existen ciertas ventajas y desventajas referentes a los tipos de vehículos, ya que por ejemplo, existe un costo fijo C que es el mismo para un vehículo sencillo y uno doble. Además, al realizar las entregas con vehículos dobles se tiene cierto beneficio ya que cada planta o centro de distribución i cuenta con cierto número limitado de andenes para atender a los vehículos, tal que f_{ih} es la cantidad de andenes disponibles en el nodo i a la hora h que a su vez es la cantidad máxima de vehículos que pueden arribar a dicho nodo en esa hora.

Los tiempos y los costos de traslados son distintos para cada tipo de vehículo, y se asume que estos datos son estáticos y conocidos, de tal manera que s^s_{ij} y s^d_{ij}

expresa el tiempo necesario para trasladarse por la arista $(i, j) \in E$ para un vehículo sencillo y uno doble, respectivamente. De igual forma se tiene que c_{ij}^s y c_{ij}^d expresa el costo necesario para trasladarse por la arista $(i, j) \in E$ para un vehículo sencillo y uno doble, respectivamente. Además se asume que el tiempo de servicio S en el que un vehículo permanece en un nodo para cargar o descargar los productos es un dato conocido y no depende de la cantidad de producto que se carga o se descarga.

En la práctica el tiempo de servicio está directamente relacionado con la cantidad de producto que se carga y se descarga. Sin embargo para la resolución de este problema fue considerado como un tiempo fijo se servicio para hacerlo mas tratable y facilitar su resolución.

Por políticas de la empresa los vehículos inician su recorrido en una de las plantas disponibles y deben regresar de nuevo a la misma planta al final del día. Los pedidos se pueden dividir en varias entregas y a su vez varios vehículos pueden visitar varias veces a los centros de distribución si esto fuese necesario. El horizonte de planeación es diario de modo que la ruta de cualquier vehículo debe de ser realizada en menos de 24 horas cumpliendo las ventanas de tiempo.

Finalmente, el objetivo es determinar la mejor configuración de rutas posible de tal modo que se minimice el costo asociado al ruteo y el costo asociado por el uso de los vehículos.

A continuación se desarrollan tres variantes del problema descrito anteriormente. Para cada una de éstas se han considerado supuestos diferentes. De manera que, cada uno de los modelos matemáticos que se describen a continuación representan problemas con distintas características.

3.2 Consideraciones y supuestos

Para la formulación del primer modelo matemático se han establecido los siguientes supuestos:

- Cada vehículo puede visitar a lo más una vez a cada centro de distribución. Dado a que los vehículos deben de regresar a su localización inicial, los recorridos de los vehículos puede ser representado como una ruta.
- No existe estocacidad en las demandas y se sabe con certeza todos los datos del problema.
- Existe al menos la misma cantidad de cabinas que de cajas de tráiler con lo que es posible formar a lo más tantos vehículos como tráileres existan en el problema.
- Un centro de distribución no puede recibir en una hora más vehículos que los andenes que se tengan en dicho centro de distribución.
- Los costos así como los tiempos de traslado de viajar de un punto a otro dependen únicamente de si el vehículo es doble o sencillo.

3.3 Modelación matemática

El uso de modelos matemáticos para la representación de un problema es muy usado en la investigación de operaciones ya que provee muchas ventajas. Por mencionar algunas se tiene que el uso de los modelos matematicos:

- Permite obtener un mejor conocimiento del problema.
- Permite una descripción concisa del problema lo cual hace más comprensible la estructura del mismo.
- A partir del modelo se suele ver con claridad los datos que son importantes para el problema.
- Para las instancias pequeñas es posible en muchos casos resolverlos con el uso de paquetes computacionales. Ésto permite obtener una idea del comportamiento del problema en este tipo de instancias.

Por otro lado, existen también algunas desventajas que deben mencionarse, como por ejemplo, se tiene que un modelo es por definición una idealización abstracta del problema, es decir, que el modelo es una simplificación del problema real implicando ciertas suposiciones que deben justificarse. Por lo tanto se debe tener cuidado que el modelo sea una representación lo más apegada posible al problema real.

En esta sección se presenta la formulación del problema como un modelo de programación entera mixta. El modelo presentado en esta sección es llamado de ahora en adelante como el Modelo A. Para la modelación se tomó como base el modelo presentado por Dondo, Méndez y Cerdá [16] quienes abordan un problema de PDP con emparejamientos. A pesar que el Modelo A resuelve un problema sin emparejamientos, con múltiples productos, entre otras restricciones, el trabajo [16] comparte algunas características similares como múltiples almacenes y ventanas de tiempo.

Conjuntos

T Conjunto de tráileres.

T_i Subconjunto de tráileres que se ubican inicialmente en el nodo i.

T^e Subconjunto de tráileres con entrepiso.

T^w Subconjunto de tráileres sin entrepiso.

K Conjunto de vehículos.

K_i Subconjunto de vehículos que se ubican inicialmente en el nodo i.

V Conjunto de plantas y centros de distribución (nodos).

E Conjunto de aristas.

P Conjunto de productos.

P_N Subconjunto de productos no retornables.

P_R Subconjunto de productos retornables.

H Conjunto de horas en el horizonte de planeación.

Parámetros

c_t^{above} Capacidad (número de tarimas) en la parte superior del tráiler t; $t \in T$

c_t^{below} Capacidad (número de tarimas) en la parte inferior del tráiler t; $t \in T$

τ_t^l Tiempo en el que el tráiler t puede iniciar su recorrido; $t \in T$

τ_t^u Tiempo en el que el tráiler t debe de regresar a la planta; $t \in T$

$\sigma(t)$ Ubicación inicial del tráiler t; $t \in T$

$\sigma(k)$ Ubicación inicial del vehículo k; $k \in K$

n_{ip}^+ Cajas del producto p que se deben recoger del nodo i; $p \in P, i \in V$

n_{ip}^- Cajas del producto p que se deben entregar al nodo i; $p \in P, i \in V$

a_i Tiempo límite inferior en el cual se puede atender al nodo i; $i \in V$

b_i Tiempo límite superior en el cual se puede atender al nodo i; $i \in V$

o_p Cantidad de cajas del producto p que caben en una tarima; $p \in P$

c_{ij}^s Costo por viajar del nodo i al nodo j usando un vehículo sencillo; $(i, j) \in E$

c_{ij}^d Costo por viajar del nodo i al nodo j usando un vehículo doble; $(i, j) \in E$

s_{ij}^s Tiempo por viajar del nodo i al nodo j usando un vehículo sencillo; $(i, j) \in E$

s_{ij}^d Tiempo por viajar del nodo i al nodo j usando un vehículo doble; $(i, j) \in E$

S Tiempo de servicio para atender a un cliente.

f_{ih} Cantidad máxima de vehículos que pueden llegar a la hora h en el nodo $h \in H, i \in V$

C Costo fijo por el uso del vehículo.

Nota: adicionalmente se utilizan parámetros auxiliares para la modelación matemática, estos son: M_T que representa el tiempo máximo de viaje de un nodo a otro, M_L que representa la máxima capacidad de un vehículo y M_C que representa el máximo costo de una ruta.

Variables Binarias

$$w_{kt} = \begin{cases} 1 & \text{Si se asigna el tráiler } t \text{ al vehículo } k; k \in K, t \in T \,|\sigma(k) = \sigma(t) \\ 0 & \text{en otro caso} \end{cases}$$

$$x_{ijk} = \begin{cases} 1 & \text{Si el vehículo } k \text{ viaja directamente de } i \text{ a } j; k \in K, (i,j) \in E \\ 0 & \text{en otro caso} \end{cases}$$

$$y_{ikh} = \begin{cases} 1 & \text{Si el vehículo } k \text{ llega al nodo } i \text{ en la hora } h; k \in K, i \in V, h \in H \\ 0 & \text{en otro caso} \end{cases}$$

$$z_k = \begin{cases} 1 & \text{Si se va usar el vehículo } k; k \in K \\ 0 & \text{en otro caso} \end{cases}$$

Variables Enteras

v_k^{above} Número de tarimas en la parte superior del vehículo k; $k \in K$

v_k^{below} Número de tarimas en la parte inferior del vehículo k; $k \in K$

v_k^{type} Variable auxiliar para determinar la configuración (sencillo/doble) del vehículo k. Toma valor de -1, 0 y 1 cuando es un vehículo sin asignar, sencillo y doble, respectivamente; $k \in K$

g_{ikp}^{-} Cantidad de cajas del producto p que entregará el vehículo k al nodo i; $p \in P, k \in K, i \in V$

g_{ikp}^{+} Cantidad de cajas del producto p que recogerá el vehículo k en el nodo i; $p \in P, k \in K, i \in V$

q_{ikp} Cantidad de cajas del producto p que tiene el vehículo k al terminar de visitar al nodo i; $p \in P, k \in K, i \in V$

Nota:

g_{ikp}^{-} No está definida para el inicio de la ruta, es decir: $g_{\sigma(k)kp}^{-} = 0$; $k \in k, p \in P$

q_{ikp} Cuando es el inicio de la ruta toma el valor de la cantidad de producto cargado en el nodo inicial, es decir: $q_{\sigma(k)kp} = g_{\sigma(k)kp}^{+}$; $k \in k, p \in P$

Variables Continuas

l_k Tiempo en el que el vehículo k puede iniciar su recorrido; $k \in K$

u_k Tiempo en el que el vehículo k debe de regresar a más tardar a su lugar de origen; $k \in K$

s_{ik} Instante en que el vehículo k empieza a atender al nodo i; $i \in V, k \in K$

c_{ik} Costo acumulado de viajar al nodo i para el vehículo k; $i \in V, k \in K$

OC_k Costo Total de viaje para el vehículo k; $k \in K$

OT_k Tiempo total de viaje para el vehículo k; $k \in K$

Modelo

$$\text{Minimizar} \quad \sum_{k \in K} OC_k + \sum_{k \in K} C z_k \tag{3.1}$$

Sujeto a:

$$\sum_{t \in T_{\sigma(k)}} w_{kt} \leq 2 \qquad k \in K \tag{3.2}$$

$$\sum_{k \in K_{\sigma(t)}} w_{kt} \leq 1 \qquad t \in T \tag{3.3}$$

$$v_k^{\text{type}} = \sum_{t \in T_{\sigma(k)}} w_{kt} - 1 \qquad k \in K \tag{3.4}$$

$$u_k = \min_{t \in T_{\sigma(k)}} \{\tau_t^u + M_T(1 - w_{kt})\} \qquad k \in K \tag{3.5}$$

$$l_k = \max_{t \in T_{\sigma(k)}} \left\{\tau_t^l w_{kt}\right\} \qquad k \in K \tag{3.6}$$

$$v_k^{\text{above}} = \sum_{t \in T_{\sigma(k)}} c_t^{\text{above}} w_{kt} \qquad k \in K \tag{3.7}$$

$$v_k^{\text{below}} = \sum_{t \in T_{\sigma(k)}} c_t^{\text{below}} w_{kt} \qquad k \in K \tag{3.8}$$

$$x_{ijk} \leq z_k \qquad i, j \in V, k \in K \tag{3.9}$$

$$\sum_{t \in T_{\sigma(k)}} w_{kt} \geq z_k \qquad k \in K \tag{3.10}$$

$$\sum_{i:(i,j) \in E} x_{ijk} - \sum_{i:(j,i) \in E} x_{jik} = 0 \qquad j \in V, k \in K \tag{3.11}$$

$$\sum_{i:(i,j) \in E} x_{ijk} \leq 1 \qquad j \in V, k \in K \tag{3.12}$$

$$a_i \sum_{j:(j,i) \in E} x_{jik} \leq s_{ik} \leq b_i \sum_{j:(j,i) \in E} x_{jik} \qquad i \in V, k \in K \tag{3.13}$$

$$a_{\sigma(k)} z_k \leq OT_k \leq b_{\sigma(k)} z_k \qquad k \in K \tag{3.14}$$

$$l_k - M_T\left(1 - z_k\right) \leq s_{\sigma(k)k} \leq u_k$$
$$+M_T\left(1 - z_k\right) \qquad k \in K \tag{3.15}$$
$$l_k - M_T\left(1 - z_k\right) \leq OT_k \leq u_k$$
$$+M_T\left(1 - z_k\right) \qquad k \in K \tag{3.16}$$
$$\sum_{h \in H} y_{ikh} h \leq \frac{s_{ik}}{60} \leq \sum_{h \in H} y_{ikh}\left(h + 1\right) \qquad k \in K, i \in V \backslash \{\sigma(k)\} \tag{3.17}$$
$$\sum_{h \in H} y_{\sigma(k)kh} h \leq \frac{OT_k}{60} < \sum_{h \in H} y_{\sigma(k)kh}\left(h + 1\right) \qquad k \in K \tag{3.18}$$
$$\sum_{k \in K} y_{ikh} \leq f_{ih} \qquad i \in V, h \in H \tag{3.19}$$
$$s_{ik} + S + s^s_{ij}\left(1 - v^{\text{type}}_k\right) + s^d_{ij} v^t_k \qquad (i,j) \in E | j \neq \sigma(k),$$
$$-M_T\left(1 - x_{ijk}\right) \leq s_{jk} \qquad k \in K \tag{3.20}$$
$$s_{ik} + S + s^s_{i\sigma(k)}\left(1 - v^{\text{type}}_k\right) + s^d_{i\sigma(k)} v^t_k$$
$$-M_T\left(1 - x_{i\sigma(k)k}\right) \leq OT_k \qquad k \in K, i \in V \tag{3.21}$$
$$\sum_{k \in K} g^+_{ikp} \leq n^+_{ip} \qquad i \in V, p \in P \tag{3.22}$$
$$\sum_{k \in K} g^-_{ikp} \geq n^-_{ip} \qquad i \in V, p \in P \tag{3.23}$$
$$g^-_{ikp} \leq n^-_{ip} \sum_{j:(j,i) \in E} x_{jik} \qquad i \in V, k \in K, p \in P \tag{3.24}$$
$$g^+_{ikp} \leq n^+_{ip} \sum_{j:(j,i) \in E} x_{jik} \qquad i \in V, k \in K, p \in P \tag{3.25}$$
$$v^{\text{above}}_k + v^{\text{below}}_k \geq \sum_{p \in P} \frac{q_{ikp}}{o_p} \qquad i \in V, k \in K \tag{3.26}$$
$$c_{ik} + c^s_{ij}\left(1 - v^{\text{type}}_k\right) + c^d_{ij} v^t_k \qquad k \in K, i \in V,$$
$$-M_C\left(1 - x_{ijk}\right) \leq c_{jk} \qquad j \in V \backslash \{\sigma(k)\} \tag{3.27}$$
$$c_{ik} + c^s_{i\sigma(k)}\left(1 - v^{\text{type}}_k\right) + c^d_{i\sigma(k)} v^t_k$$
$$-M_C\left(1 - x_{i\sigma(k)k}\right) \leq OC_k \qquad k \in K, i \in V \tag{3.28}$$
$$q_{ikp} + g^+_{jkp} - g^-_{jkp} \qquad k \in K, i \in V, p \in P,$$
$$-M_L\left(1 - x_{ijk}\right) \leq q_{jkp} \qquad j \in V \backslash \{\sigma(k)\} \tag{3.29}$$
$$q_{ikp} + g^+_{jkp} - g^-_{jkp} \qquad k \in K, i \in V, p \in P,$$
$$+M_L\left(1 - x_{ijk}\right) \geq q_{jkp} \qquad j \in V \backslash \{\sigma(k)\} \tag{3.30}$$
$$\sum_{p \in P_N} \frac{q_{ikp}}{o_p} \leq v^{\text{below}}_k + \sum_{t \in T_{\sigma(k)} \cap T^e} c^{\text{above}}_t w_{kt} \qquad i \in V, k \in K \tag{3.31}$$

$$\sum_{h \in H} y_{ikh} \leq 1 \qquad i \in V, k \in K \tag{3.32}$$

$$w_{kt} = 0 \qquad t \in T \,|\sigma(k) \neq \sigma(t), \; k \in K \tag{3.33}$$

$$g^{-}_{\sigma(k)kp} = 0 \qquad k \in K, p \in P \tag{3.34}$$

$$q_{\sigma(k)kp} = g^{+}_{\sigma(k)kp} \qquad k \in K, p \in P \tag{3.35}$$

$$w_{kt}, x_{ijk}, y_{ikh}, z_k \in \{0,1\} \qquad i,j \in V, t \in T, k \in K, h \in H \tag{3.36}$$

$$x_{ijk} \in \{0,1\} \qquad k \in K, (i,j) \in E \tag{3.37}$$

$$v^{\text{above}}_k, v^{\text{below}}_k, g^{-}_{ikp}, g^{+}_{ikp}, q_{ikp} \in Z^{+} \qquad i \in V, k \in K, p \in P \tag{3.38}$$

$$l_k, u_k, s_{ik}, c_{ik}, OC_k, OT_k \geq 0 \qquad i \in V, k \in K \tag{3.39}$$

$$v^{\text{type}}_k \in \{-1, 0, 1\} \qquad k \in K \tag{3.40}$$

El objetivo (3.1) es de minimizar el costo asociado al ruteo de vehículos (que está en función al tipo de vehículo) aunado al costo asociado al costo fijo por el uso del vehículo. Las restricciones (3.2) limitan a que el número de tráileres que se le pueden asignar a un vehículo sea a lo más dos. Las restricciones (3.3) aseguran que un tráiler solo se le puede asignar a un vehículo. Para poder identificar el tipo de configuración del vehículo se establece la variable v^{type}_k la cual queda definida en las restricciones (3.4). Las restricciones (3.5) y (3.6) definen las ventanas de tiempo de cada vehículo, las cuales deben de cumplir con las ventanas de tiempo de cada tráiler que le es asignado. Las restricciones (3.7) y (3.8) de manera similar garantizan que la cantidad de tarimas en la parte superior e inferior respectivamente de un vehículo sea igual a la suma de las tarimas de los tráileres asignados al vehículo. Las restricciones (3.9) garantizan que solo se pueden asignar rutas a vehículos que están habilitados. Las restricciones (3.10) establecen que sólo se pueden habilitar vehículos que tienen tráiler. Las restricciones (3.11) y (3.12) sirven para cumplir el criterio de conservación de flujo y restringen a que un vehículo pueda solo visitar una vez a un centro de distribución. Las restricciones (3.13) y (3.14) garantizan el cumplimiento de las ventanas de tiempo de los nodos. Las restricciones (3.15) y (3.16) garantizan el cumplimiento de las ventanas de tiempo de los vehículos. Con las

restricciones (3.17) y (3.18) el vehículo debe de llegar a la hora que le fue asignada. Las restricciones (3.19) garantizan el cumplimiento de no exceder el máximo número de andenes en un centro de distribución. Las restricciones (3.20) y (3.21) sirven para asegurar que las variables de tiempo son consistentes con los viajes y los tiempos de servicio. Las restricciones (3.22) y (3.23) garantizan el cumplimiento de la demanda sin exceder la capacidad de las plantas. Las restricciones (3.24) y (3.25) garantizan que un vehículo solo puede entregar o recoger producto en un nodo siempre y cuando este nodo haya sido visitado por el vehículo. Las restricciones (3.26) sirven para no exceder la capacidad del vehículo. Con las restricciones (3.27) y (3.28) se asegura que las variables de costo sean consistentes. Las restricciones (3.29) y (3.30) se refieren a que la carga del vehículo k en el nodo j depende de la carga que tenía en el nodo i si el vehículo viajo directamente de i a j. Para cumplir la restricción del acomodo de las cajas en los vehículos basta con que el número máximo de tarimas ocupadas por las cajas tipo P_N sea menor igual al número de tarimas debajo del vehículo en caso de que no tenga entrepiso (3.31). Con esto se garantiza que existe al menos un modo de cómo acomodar las cajas de tal manera que sea factible la solución como se puede apreciar en la Figura 3.1 y la Figura 3.2. Las restricciones (3.32) establecen que un vehículo sólo puede llegar a lo más una vez a un nodo. Con las restricciones (3.33) se imposibilita que se pueda asignar un tráiler que no esté en el mismo lugar inicial que del vehículo. Con las restricciones (3.34) se establece que la cantidad que tiene cargado el vehículo al momento de partir es la misma cantidad que el vehículo cargo en la planta de inicio. Con las restricciones (3.35) se imposibilita a que un vehículo pueda dejar producto al inicio de su ruta. Esto es debido a que inicialmente el vehículo no cuenta con material cargado sino que en el lugar de inicio carga los productos para después repartirlos. Finalmente las restricciones (3.36)-(3.40) definen la naturaleza de las variables del modelo matemático.

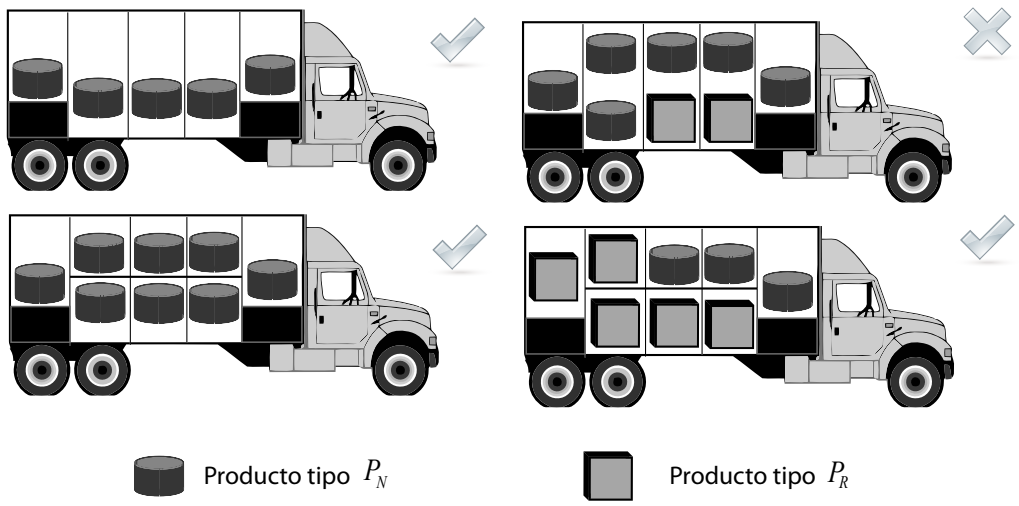


Figura 3.1: Cumplimiento de la restricción de compartimientos en un vehículo sencillo.

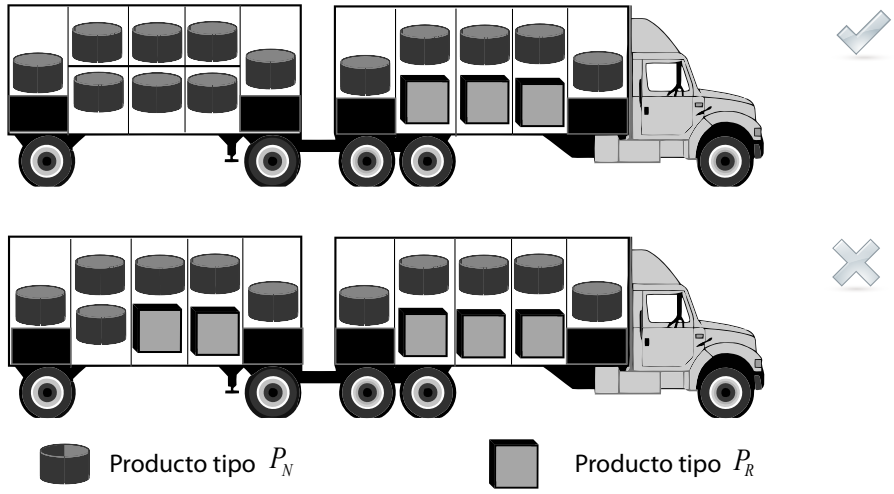


Figura 3.2: Cumplimiento de la restricción de compartimientos en un vehículo doble.

3.4 Formulación de modelos alternativos

La formulación presentada anteriormente parte del hecho de que se conocen las cantidades de producto que demandan los centro de distribución y las capacidades de las plantas, y es planteado como un problema de carga y descarga sin emparejamiento. Mas no en todas las formulaciones resulta práctico expresarlo de este modo por la complejidad para resolver este tipo de modelos. Típicamente suele expresarse como un problema de carga y descarga con emparejamientos.

Sin embargo si previamente se conociera la cantidad de producto (p) que entregará una planta (i) para atender a un centro de distribución (j), es decir, que ésta decisión haya sido tomada en una etapa anterior, el problema de estudio formaría parte de los problemas catalogados como problemas de carga y descarga de producto con emparejamientos.

3.4.1 Modelo B

Como se ha mencionado anteriormente, resulta impráctico la formulación del Modelo A por el esfuerzo computacional que se requiere para resolverlo, por lo que se propone una nueva formulación donde ya se tiene asignado el origen y destino de las entregas para satisfacer la demanda de los centros de distribución, es decir, un modelo de carga y descarga con emparejamientos.

El problema expuesto podría expresarse como un problema de carga y descarga con emparejamientos. Para ello basándonos en la formulación del Modelo A propuesto anteriormente, se propone un modelo de carga y descarga con emparejamientos que será llamado de ahora en adelante como el Modelo B.

Para ello se parte de que se cuenta con un conjunto R que contiene todos los requerimientos ($r_{ijp} \in R$) donde r_{ijp} es la cantidad de producto p que se recoge en el nodo i y que será entregada directamente en el nodo j. Con base a la formulación del Modelo A se remplazan las variables g^{+}_{ikp}, g^{-}_{ikp}, y q_{ikp} por las variables g_{ijpk}

que representan la cantidad de cajas que transportará el vehículo k para atender el requerimiento r_{ijp}. Se sustituyen las restricciones (3.22) y (3.23) que aseguran el complimiento de las demandas sin exceder la capacidad de las plantas, por las restricciones (3.41) que aseguran que todos los requerimientos sean cumplidos en su totalidad.

$$\sum_{k \in K} g_{ijpk} = r_{ijp} \quad i, j \in V, p \in P \tag{3.41}$$

Además las restricciones (3.24) y (3.25) que aseguran que un vehículo puede cumplir la demanda del nodo i solamente si visitó al nodo i, se remplazan por las restricciones (3.42) que aseguran que un vehículo puede atender todos los requerimientos que se cargan en el nodo i y se descargan en el nodo j siempre y cuando se haya viajado de i a j.

$$\sum_{p \in P} g_{ijpk} \leq M_L x_{ijk} \quad i, j \in V, k \in K \tag{3.42}$$

Las restricciones (3.26) son reemplazadas por las restricciones (3.43) para asegurar no exceder la capacidad de los vehículos.

$$v_k^{\text{above}} + v_k^{\text{below}} \geq \sum_{p \in P} \sum_{j \in V} \frac{g_{ijpk}}{o_p} \quad i \in V, k \in K \tag{3.43}$$

Finalmente las restricciones (3.31) son reemplazadas por las restricciones (3.44) para asegurar que exista al menos una configuración valida de tal modo que no se aplasten los productos tipo P_N.

$$\sum_{p \in P_N} \sum_{j \in V} \frac{g_{ijpk}}{o_p} \leq v_k^{\text{below}} + \sum_{t \in T_{\sigma(k)} \cap T^e} c_t^{\text{above}} w_{kt} \quad i \in V, k \in K \tag{3.44}$$

Ya que esta formulación con emparejamientos fija algunas decisiones, el Modelo B omite las restricciones (3.29), (3.30),(3.34) y (3.35) que están incluidas en el Modelo A.

3.4.2 Modelo C

Los modelos presentados anteriormente presentan la restricción de que un vehículo no le es posible visitar varias veces al mismo centro de distribución. Al eliminar esta restricción los vehículos tienen la capacidad de visitar varias veces un centro de distribución, respetando la restricción que establece que deben de iniciar y terminar su recorrido en la planta de origen.

En la literatura se ha abordado este criterio en varios trabajos como por ejemplo Gribkovskaia, Halskau y Bugge [26] quienes presentan una heurística para resolver un VRP o el trabajo presentado por Hoff et al. [30] que presentan una solución basada en búsqueda local para resolver el problema de ruteo de vehículos con soluciones tipo *Lasso*. En las soluciones tipo *Lasso* es posible visitar un nodo más de una vez con un mismo vehículo, por lo que los recorridos de los vehículos son representados como caminos cerrados en vez de rutas.

Con base a lo anterior se presenta un modelo donde se relaja la restricción de solo poder visitar a un centro de distribución a lo más una vez con un vehículo. Para la formulación de este modelo se ha establecido un parámetro $\hat{e}$ tal que el $\hat{e}$-ésimo nodo visitado por el vehículo k es el último nodo visitado por dicho vehículo. El valor del parámetro $\hat{e}$ toma un valor lo suficientemente grande para garantizar que sea posible formar rutas que incluyan a todos los nodos. Se ha observado que típicamente la cantidad de nodos que conforman una ruta no exceden más de los diez elementos. Con este tipo de formulación es posible formar rutas cuya cantidad de visitas sea a lo más de $\hat{e}$, sin que exista la restricción en que estrictamente la cantidad de nodos sea de $\hat{e}$ elementos.

Variables Binarias

$$w_{kt} = \begin{cases} 1 & \text{Si se asigna el tráiler } t \text{ al vehículo } k;\ k \in K, t \in T\,|\sigma(k) = \sigma(t) \\ 0 & \text{en otro caso} \end{cases}$$

$$x_{iek} = \begin{cases} 1 & \text{Si el nodo } i \text{ es el } e\text{-ésimo nodo visitado por el vehículo } k; \\ 0 & \text{en otro caso; } i \in V, e \in [0, \hat{e}] \cap Z^+ \end{cases}$$

$$y'_{ekh} = \begin{cases} 1 & \text{Si el vehículo } k \text{ llega al } e\text{-ésimo nodo en la hora } h; \\ 0 & \text{en otro caso; } k \in K, h \in H, e \in [0, \hat{e}] \cap Z^+ \end{cases}$$

$$y_{ikh} = \begin{cases} 1 & \text{Si el vehículo } k \text{ llega al nodo } i \text{ en la hora } h;\ k \in K, i \in V, h \in H \\ 0 & \text{en otro caso} \end{cases}$$

$$z_k = \begin{cases} 1 & \text{Si se utiliza el vehículo } k;\ k \in K \\ 0 & \text{en otro caso} \end{cases}$$

Variables Enteras

v_k^{above} Número de tarimas en la parte superior del vehículo k; $k \in K$

v_k^{below} Número de tarimas en la parte inferior del vehículo k; $k \in K$

v_k^{type} Variable auxiliar para determinar la configuración (sencillo/doble) del vehículo k. Toma valor de -1, 0 y 1 cuando es un vehículo sin asignar, sencillo y doble, respectivamente; $k \in K$

g^-_{iekp} Cantidad de cajas del producto p que entregara el vehículo k en el e-ésimo nodo visitado; $i \in V, e \in [0, \hat{e}] \cap Z^+, k \in K, p \in P$

g^+_{iekp} Cantidad de cajas del producto p que recogerá el vehículo k en el e-ésimo nodo visitado; $i \in V, e \in [0, \hat{e}] \cap Z^+, k \in K, p \in P$

q_{ekp} Carga del producto p que tiene el vehículo k al terminar de visitar al e-ésimo nodo de su ruta; $e \in [0, \hat{e}] \cap Z^+, k \in K, p \in P$

Nota:

g^-_{iekp} No está definida para el inicio de la ruta, es decir: $g^-_{i0kp} = 0;\ k \in k,$ $p \in P, i \in V$

q_{ekp} Cuando es el inicio de la ruta toma el valor de la cantidad de producto cargado en el nodo inicial, es decir: $q_{0kp} = \sum_{i \in V} g^+_{i0kp};\ k \in k, p \in P$

Variables Continuas

l_k Tiempo en el que el vehículo k puede iniciar su recorrido; $k \in K$

u_k Tiempo en el que el vehículo k debe de regresar a más tardar a su lugar

de origen; $k \in K$

s_{ek} Instante en que el vehículo k empieza a atender al e-ésimo nodo visitado; $e \in [0, \hat{e}] \cap Z^+, k \in K$

c_{ek} Costo acumulado de viajar al e-ésimo nodo visitado para el vehículo k; $e \in [0, \hat{e}] \cap Z^+, k \in K$

Modelo

$$\text{Minimizar} \quad \sum_{k \in K} c_{\hat{e}k} + \sum_{k \in K} C z_k \tag{3.45}$$

Sujeto a:

$$\sum_{t \in T_{\sigma(k)}} w_{kt} \leq 2 \qquad k \in K \tag{3.46}$$

$$\sum_{k \in K_{\sigma(t)}} w_{kt} \leq 1 \qquad t \in T \tag{3.47}$$

$$v_k^{\text{type}} = \sum_{t \in T_{\sigma(k)}} w_{kt} - 1 \qquad k \in K \tag{3.48}$$

$$u_k = \min_{t \in T_{\sigma(k)}} \{\tau_t^u + M_T(1 - w_{kt})\} \qquad k \in K \tag{3.49}$$

$$l_k = \max_{t \in T_{\sigma(k)}} \left\{\tau_t^l w_{kt}\right\} \qquad k \in K \tag{3.50}$$

$$v_k^{\text{above}} = \sum_{t \in T_{\sigma(k)}} c_t^{\text{above}} w_{kt} \qquad k \in K \tag{3.51}$$

$$v_k^{\text{below}} = \sum_{t \in T_{\sigma(k)}} c_t^{\text{below}} w_{kt} \qquad k \in K \tag{3.52}$$

$$x_{iek} \leq z_k \qquad i \in V, k \in K, \; e \in [0, \hat{e}] \cap Z^+ \tag{3.53}$$

$$\sum_{t \in T_{\sigma(k)}} w_{kt} \geq z_k \qquad k \in K \tag{3.54}$$

$$x_{\sigma(k)0k} = x_{\sigma(k)\hat{e}k} = z_k \qquad k \in K \tag{3.55}$$

$$\sum_{i \in V} x_{iek} = z_k \qquad k \in K, e \in [0, \hat{e}] \cap Z^+ \tag{3.56}$$

$$\sum_{i \in V} a_i x_{iek} \leq s_{ek} \leq \sum_{i \in V} b_i x_{iek} \qquad k \in K, e \in [0, \hat{e}] \cap Z^+ \tag{3.57}$$

$$l_k \leq s_{0k} \qquad k \in K \tag{3.58}$$

$$u_k \geq s_{\hat{e}k} \qquad k \in K \tag{3.59}$$

$$\sum_{h \in H} y'_{ekh} h \leq \frac{s_{ek}}{60} \leq \sum_{h \in H} y'_{ekh} (h+1) \qquad k \in K, e \in [0, \hat{e}] \cap Z^+ \tag{3.60}$$

$$\sum_{h\in H} y'_{ekh} \leq 1 \qquad k \in K, e \in [0,\hat{e}] \cap Z^+ \qquad (3.61)$$

$$y'_{ekh} + x_{iek} \leq y_{ikh} + 1 \qquad i \in V, k \in K, e \in [0,\hat{e}] \cap Z^+, h \in H \qquad (3.62)$$

$$\sum_{k\in K} y_{ikh} \leq f_{ih} \qquad i \in V, h \in H \qquad (3.63)$$

$$s_{ek} + S + s^s_{ij}\left(1 - v^{\text{type}}_k\right) + s^d_{ij} v^t_k - M_T\left(2 - x_{iek} - x_{j(e+1)k}\right) \leq s_{(e+1)k} \qquad k \in K, i \neq j \in V, e \in [0,\hat{e}-1] \cap Z^+ \qquad (3.64)$$

$$s_{ek} - M_T\left(2 - x_{iek} - x_{i(e+1)k}\right) \leq s_{(e+1)k} \qquad k \in K, i \in V, e \in [0,\hat{e}-1] \cap Z^+ \qquad (3.65)$$

$$\sum_{e\in[0,\hat{e}]\cap Z} \sum_{k\in K} g^+_{iekp} \leq n^+_{ip} \qquad i \in V, p \in P \qquad (3.66)$$

$$\sum_{e\in[0,\hat{e}]\cap Z} \sum_{k\in K} g^-_{iekp} \geq n^-_{ip} \qquad i \in V, p \in P \qquad (3.67)$$

$$g^+_{iekp} \leq n^+_{ip} x_{iek} \qquad i \in V, k \in K, p \in P, e \in [0,\hat{e}] \cap Z^+ \qquad (3.68)$$

$$g^-_{iekp} \leq n^-_{ip} x_{iek} \qquad i \in V, k \in K, p \in P, e \in [0,\hat{e}] \cap Z^+ \qquad (3.69)$$

$$q_{ekp} + \sum_{i\in V} g^+_{i(e+1)kp} - \sum_{i\in V} g^-_{i(e+1)kp} = q_{(e+1)kp} \qquad k \in K, p \in P, e \in [0,\hat{e}-1] \cap Z^+ \qquad (3.70)$$

$$v^{\text{above}}_k + v^{\text{below}}_k \geq \sum_{p\in P} \frac{q_{ekp}}{o_p} \qquad k \in K, e \in [0,\hat{e}] \cap Z^+ \qquad (3.71)$$

$$\sum_{p\in P_N} \frac{q_{ekp}}{o_p} \leq v^{\text{below}}_k + \sum_{t\in T_{\sigma(k)}\cap T^e} c^{\text{above}}_t w_{kt} \qquad k \in K, e \in [0,\hat{e}] \cap Z^+ \qquad (3.72)$$

$$c_{ek} + c^s_{ij}\left(1 - v^{\text{type}}_k\right) + c^d_{ij} v^t_k - M_C\left(2 - x_{iek} - x_{j(e+1)k}\right) \leq c_{(e+1)k} \qquad k \in K, i, j \in V, e \in [0,\hat{e}-1] \cap Z^+ \qquad (3.73)$$

$$w_{kt} = 0 \qquad k \in K, t \in \{T \,|\sigma(k) \neq \sigma(t)\} \qquad (3.74)$$

$$g^-_{i0kp} = 0 \qquad k \in K, p \in P, i \in V \qquad (3.75)$$

$$q_{0kp} = \sum_{i \in V} g^{+}_{i0kp} \qquad k \in K, p \in P \tag{3.76}$$

$$w_{kt}, x_{iek}, y'_{ekh}, y_{ikh}, z_k \in \{0,1\} \qquad i \in V, t \in T, k \in K, e \in [0, \hat{e}] \cap Z^{+}, h \in H \tag{3.77}$$

$$v^{\text{above}}_{k}, v^{\text{below}}_{k}, g^{-}_{iekp}, g^{+}_{iekp}, q_{ekp} \in Z^{+} \qquad i \in V, k \in K, p \in P, e \in [0, \hat{e}] \cap Z^{+} \tag{3.78}$$

$$l_k, u_k, s_{ek}, c_{ek} \geq 0 \qquad k \in K, e \in [0, \hat{e}] \cap Z^{+} \tag{3.79}$$

$$v^{\text{type}}_{k} \in \{-1, 0, 1\} \qquad k \in K \tag{3.80}$$

El objetivo (3.45) es de minimizar el costo asociado al ruteo de vehículos (que está en función del tipo de vehículo) aunado al costo asociado al costo fijo por el uso del vehículo. Las restricciones (3.46) limitan a que el número de tráileres que se le pueden asignar a un vehículo sea a lo más dos. Las restricciones (3.47) aseguran que un tráiler solo se le puede asignar a un vehículo. Para poder identificar el tipo de configuración del vehículo se establece la variable v^{type}_{k} el cual queda definida en las restricciones (3.48). Las restricciones (3.49) y (3.50) declaran que las ventanas de tiempo de un vehículo deben de cumplir con las ventanas de tiempo de cada tráiler que fue asignado a ese vehículo. Las restricciones (3.51) y (3.52) de manera similar garantizan que la cantidad de tarimas en la parte superior e inferior respectivamente de un vehículo sean igual a la suma de las tarimas de los tráileres asignados al vehículo. Con las restricciones (3.53) se restringe a que solo se puedan asignar rutas a vehículos que están habilitados. Las restricciones (3.54) indican que sólo se pueden habilitar vehículos que tienen tráiler. Las restricciones (3.55) garantizan que cada vehículo inicia y termina en su nodo origen. Con las restricciones (3.56) se asegura que para cada vehículo se tenga exactamente una sola visita en cada posición de su ruta. Con las restricciones (3.57) se garantiza cumplir con las horas de llegada a los centros de distribución. Con las restricciones (3.58) y (3.59) se garantiza cumplir con horarios del vehículo. Con las restricciones (3.60) se garantiza que el vehículo llegue a la hora que le fue asignada. Las restricciones (3.61) sirven para que cada posición de la ruta de un vehículo solamente se pueda llegar a lo más en un horario. Las restricciones (3.62) sirven para indicar en que horario el vehículo k visitó al nodo i.

Las restricciones (3.63) garantizan el cumplimiento de respetar el máximo número de llegadas en la hora h. Las restricciones (3.64) y (3.65) sirven para asegurar que las variables de tiempo sean consistentes con los viajes y los tiempos de servicio. Las restricciones (3.66) y (3.67) sirven para cumplir la demanda de los centros de distribución sin exceder la capacidad de las plantas. Con las restricciones (3.68) y (3.69) se garantiza que un vehículo solo puede cargar o descargar producto al nodo i siempre y cuando haya visitado al nodo i. Con las restricciones (3.70) se garantiza que la carga del vehículo k en el nodo j depende de la carga que tenía en el nodo i si el vehículo viajo directamente de i a j. Las restricciones (3.71) garantizan que no se exceda la capacidad del vehículo. Para cumplir la restricción del acomodo de las cajas en los vehículos basta con que el número máximo de tarimas ocupadas por la cajas tipo P_N sea menor igual al número de tarimas debajo del vehículo en caso de que no tenga entrepiso. De este modo las restricciones (3.72) garantizan que exista al menos una manera de acomodar las cajas de tal manera que la solución sea factible. Las restricciones (3.73) sirven para asegurar que las variables de costo sean consistentes. Con las restricciones (3.74) se imposibilita que se pueda asignar un tráiler que no esté en el mismo lugar inicial que del vehículo. Con las restricciones (3.75) se imposibilita a que un vehículo pueda dejar producto al inicio de su ruta. Esto es debido a que inicialmente el vehículo no cuenta con producto cargado. Con las restricciones (3.76) se establece que la cantidad que tiene cargado el vehículo al momento de partir es la misma cantidad que el vehículo cargó en la planta de inicio. Las restricciones (3.77)-(3.80) definen la naturaleza de las variables del modelo.

Cada una de las formulaciones matemáticas planteadas anteriormente responde a un problema determinado. La formulación del Modelo A ha sido planteada para abordar al problema como uno sin emparejamientos y restringiendo a que un vehículo puede visitar a lo más una vez un nodo. La formulación del Modelo B ha sido planteada para abordar el problema como uno con emparejamientos y restringiendo a que un vehículo pueda visitar a lo mas una vez a un nodo. La formulación del Modelo C ha sido planteada para abordar al problema como uno sin emparejamientos pero con la posibilidad de que un vehículo pueda visitar más de una vez a un nodo.

3.5 Resolución del Modelo A en forma exacta

El número de variables y de restricciones del Modelo A presentado está en función al número de tráileres ($|T|$), al número de nodos ($|V|$) y al número de productos ($|P|$) y está dado de la siguiente manera:

- Variables binarias: $|T|^2 + |V|^2 |T| + 24 |V| |T| + |T|$
- Variables enteras: $3 |T| + 3 |V| |T| |P|$
- Variables continuas: $4 |T| + 2 |V| |T|$
- Restricciones: $11 |T| + |V|^2 |T| + 8 |V| |T| + (|V| - 1) |T| + 24 |V| + 2 |V| (|V| - 1) |T| + 2 |V| |P| + 2 |V| |T| |P| + 2 |V| (|V| - 1) |T| |P|$

En las Tabla 3.1 y 3.2 se aprecia cómo crece el número de variables (binarias y enteras) y el número de restricciones con respecto al número de tráileres y nodos.

En una primera inspección es posible ver que el número de variables aumenta mas rápidamente que el número de restricciones cuando se incrementa la cantidad de tráileres. Por otro lado se observa que el número de restricciones aumenta mas rápidamente que el número de variables cuando se incrementa la cantidad de nodos aunque el número de restricciones aumentan más rápidamente aumentando el número de nodos que el número de tráileres.

Se realizó una prueba estadística que consistió en un diseño factorial con tres factores para determinar el impacto que tiene el tamaño de la instancia en el tiempo de ejecución del método exacto.

Se analizaron los siguientes factores:

- Número de nodos (4, 5, 6)

		$\|T\| = 1$	$\|T\| = 5$	$\|T\| = 10$	$\|T\| = 20$	$\|T\| = 100$
$\|V\| = 1$	Binarias	27	155	360	920	12600
	Enteras	9	45	90	180	900
	Continuas	6	30	60	120	600
$\|V\| = 5$	Binarias	147	755	1560	3320	24600
	Enteras	33	165	330	660	3300
	Continuas	14	70	140	280	1400
$\|V\| = 10$	Binarias	342	1730	3510	7220	44100
	Enteras	63	315	630	1260	6300
	Continuas	24	120	240	480	2400
$\|V\| = 20$	Binarias	882	4430	8910	18020	98100
	Enteras	123	615	1230	2460	12300
	Continuas	44	220	440	880	4400
$\|V\| = 100$	Binarias	12402	62030	124110	248420	1250100
	Enteras	603	3015	6030	12060	60300
	Continuas	204	1020	2040	4080	20400

Tabla 3.1: Cantidad de variables en función a la cantidad de nodos y tráileres para $|P| = 2$.

	$\|T\| = 1$	$\|T\| = 5$	$\|T\| = 10$	$\|T\| = 20$	$\|T\| = 100$
$\|V\| = 1$	52	148	268	508	2428
$\|V\| = 5$	360	1240	2340	4540	22140
$\|V\| = 10$	1060	4180	8080	15880	78280
$\|V\| = 20$	3510	15310	30060	59560	295560
$\|V\| = 100$	73510	356350	709900	1417000	7073800

Tabla 3.2: Número de restricciones en función a los nodos y a los tráileres para $|P| = 2$.

- Número de tráileres (4, 5, 6, 7)
- Número de productos (2, 3, 4, 5)

Para cada combinación de dichos factores se generaron tres réplicas dando un total de 144 instancias a analizar. Fueron resueltas estas instancias utilizando el método de ramificación y acotamiento implementado por la herramienta computacional CPLEX 11.2 [33], llevados a cabo estos experimentos en un procesador Intel Core i7 bajo el sistema operativo Windows 7, obteniendo así los tiempos de ejecución para la realización del análisis estadístico. El software estadístico que se utilizó para analizar los resultados fue el Design-Expert 8.0.6.1 [70] en su versión de prueba.

En un primer experimento se busca determinar si existe una correlación en los tiempos de ejecución. Para el análisis de varianza (ANOVA) fue utilizada la prueba estadística de Fisher [20]. Dicha prueba es útil para probar el grado de significancia estadística cuando el tamaño de la muestra es reducido. Para dicha experimentación se estableció como hipótesis nula que no existe una correlación en los tiempos de ejecución y en el tamaño de las instancias; y como hipótesis alternativa se estableció que si existe una correlación en los tiempos de ejecución y en el tamaño de las instancias.

Entrada	Suma de los cuadrados	Grados de libertad	Cuadrado Medio	Valor F	Prob > F
V	$2.782E+006$	2	$1.393E+006$	19.13	<0.0001
T	$6.176E+006$	3	$2.059E+006$	28.26	<0.0001
P	$2.260E+006$	3	$7.534E+005$	10.34	<0.0001
$V-T$	$7.157E+006$	6	$1.193E+006$	16.37	<0.0001
$V-P$	$2.073E+006$	6	$3.454E+005$	4.74	0.0003
$T-P$	$5.746E+006$	9	$6.384E+005$	8.76	<0.0001
$V-T-P$	$5.381E+006$	18	$2.989E+005$	4.10	<0.0001

Tabla 3.3: Resultados del análisis de varianza.

En la Tabla 3.3 se muestran los resultados de la prueba estadística. Cada renglón expresa una prueba de significancia para uno o varios de los factores analizados, a manera de ejemplo se tiene que, el primer renglón se expone la prueba de significancia para el factor $|V|$ (la cantidad de nodos). Los resultados de dicha prueba determinan estadísticamente si los tiempos de ejecución obtenidos fueron producto del azar o si el factor analizado tiene un impacto significativo en los tiempos de ejecución. Si se obtiene un valor de probabilidad relativamente bajo (usualmente menor al 5 %) se rechaza la hipótesis nula. Como se puede observar en la última columna de la tabla, es posible concluir que estadísticamente cada factor analizado es significativo y tiene un impacto en los tiempos de ejecución.

Como se puede ver en la Figura 3.3 a medida que la cantidad de nodos aumenta el tiempo de ejecución aumenta, mientras que en la Figura 3.4 se puede observar que se dispara el tiempo de ejecución con una cantidad mínima de tráileres. En la Figura 3.5 se puede ver que la cantidad de productos no influye tanto como los otros dos parámetros, una posible justificación es el hecho de que el número de productos no incrementa tanto a la cantidad de variables y de restricciones como el número de tráileres y de nodos.

El problema real cuenta con: 24 nodos, 153 tráileres y 362 tipos de productos. Lo que significa que en el Modelo A, la cantidad de variables y restricciones son 4,196,025 y 64,114,242 respectivamente. Está más que claro que la instancia real es intratable, lo que con la experimentación previa se podría anticipar que no será posible encontrar la solución en un tiempo razonable.

Se intentaron algunas estrategias con la finalidad de encontrar una solución factible en un tiempo razonable utilizando la herramienta computacional CPLEX, para ello se modificaron los siguientes parámetros propios de la herramienta:

- FeasOptMode = 1 (Encontrar un punto factible para un ambiente relajado).
- FPHeur = 1 (Si la heurística se centran en la búsqueda de una solución factible).

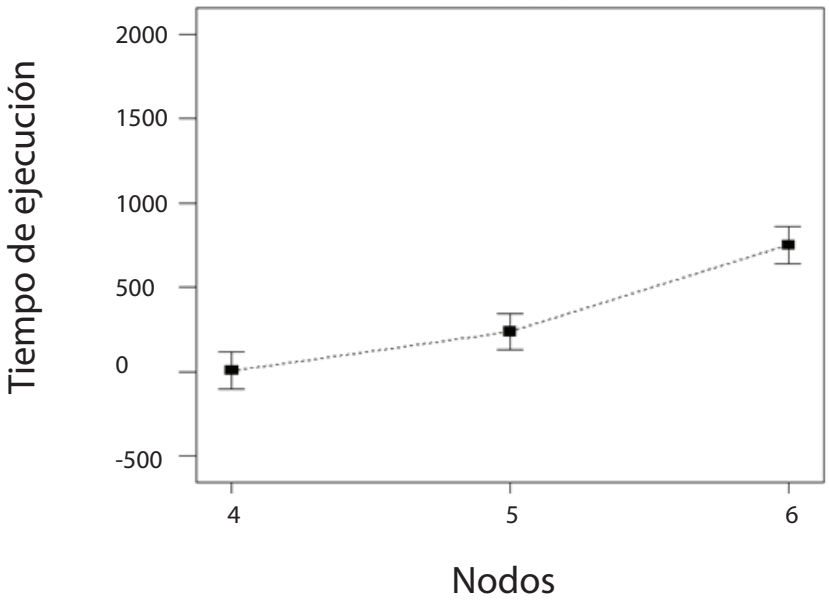


Figura 3.3: Interacción entre el tiempo de ejecución y el número de nodos.

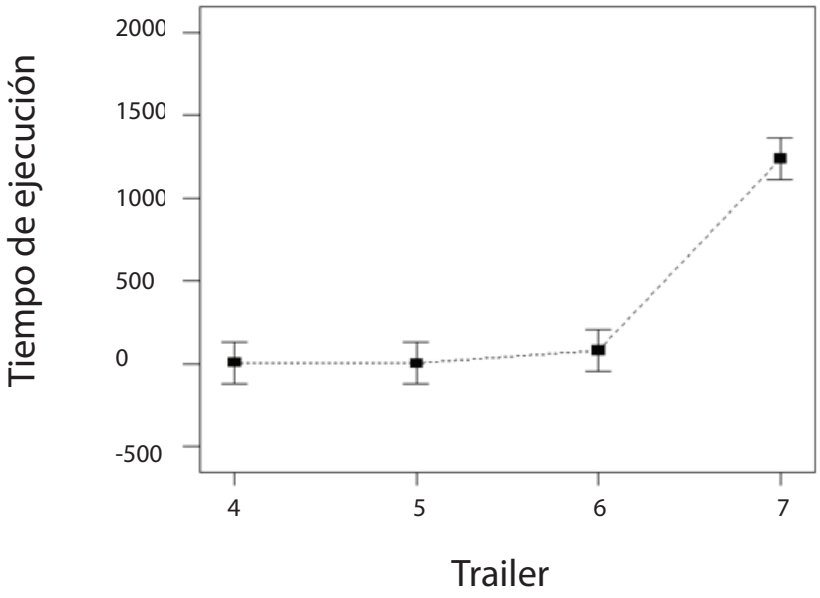


Figura 3.4: Interacción entre el tiempo de ejecución y el número de tráileres.

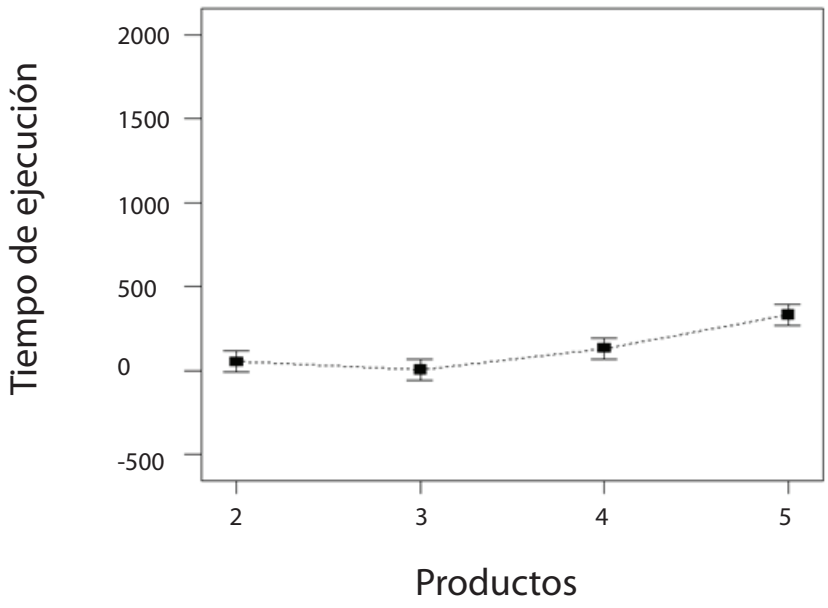


Figura 3.5: Interacción entre el tiempo de ejecución y el número de productos.

- MIPEmphasis = 1 (Para hacer hincapié en la búsqueda de soluciones factibles).

Lamentablemente no fue posible obtener siquiera una solución factible en donde cuyo error fue la falta de memoria para cargar el modelo matemático. A este respecto, se intentó adecuar el sistema de computo para que usara la máxima memoria disponible en un sistema de 32 bits (4GB), sin embargo se obtuvieron los mismos resultados.

Debido a estos resultados se refuerza la necesidad de utilizar un método rápido que otorgue buenas soluciones como es el caso de las heurísticas para instancias de tamaño real ya que, como se ha expuesto, es muy difícil obtener la solución factible para los problemas de tamaño real.

3.6 Análisis de complejidad

La teoría de la complejidad computacional establece que el grado de complejidad que puede existir, principalmente en el esfuerzo computacional requerido para resolver en forma óptima problemas de optimización. Un problema que pertenece a la clase NP-completo [23], en términos simples, significa que no se conoce un método exacto para resolver el problema en tiempo polinomial. Probar que un problema es NP-completo no es un resultado negativo, sino más bien es un resultado importante para los investigadores, ya que esto determina que es prácticamente imposible encontrar soluciones exactas en tiempo razonable para instancias reales del problema en cuestión. Esto da pauta a que los esfuerzos se centren en el desarrollo de métodos heurísticos con la finalidad de hallar soluciones de buena calidad en tiempos razonables. Se desarrolló un análisis de complejidad para el Modelo A, obteniendo una reducción del problema del problema VRPTW al Modelo A, con lo cual es posible concluir que el Modelo A pertenece a la clase de problemas NP-completo. Para una explicación más detallada se puede ver en el Apéndice A el desarrollo del análisis de complejidad para el Modelo A.

Cabe aclarar que solamente se desarrolló el análisis de complejidad para el Modelo A, debido a que es más frecuente encontrar en la literatura problemas de carga y descarga que no permitan soluciones tipo *Lasso*; sin embargo, es de esperarse que los Modelos B y C pertenezcan a la clase de problemas NP-completo. El análisis de complejidad para estos modelos no formó parte del alcance de este estudio.

Capítulo 4

Metodologia de Solución

En este capítulo se expone la descripción y desarrollo del método de solución para el problema descrito en la Sección 3.4.2. Este método tiene su justificación en el hecho de que se ha demostrado que no es posible resolver un problema de tamaño real de manera exacta con un tiempo de resolución razonable. Ni siquiera fue posible obtener alguna solución factible como se presentó en el Capitulo 3.

La metodología propuesta para la solución del problema es la implementación de una metaheurística GRASP. Esta metaheurística, propuesta por Feo y Resende [19], es un procedimiento voraz aleatorizado comúnmente utilizado para resolver problemas de optimización combinatoria. Dicha metaheurística se compone de dos fases: una fase de construcción y otra fase de búsqueda local, las cuales son ejecutadas cierto número de iteraciones ($maxIterations$) obteniendo una solución diferente en cada iteración para reportar finalmente la mejor solución encontrada en el procedimiento. La fase constructiva, como su nombre lo dice, lleva a cabo la construcción de una solución insertando parcialmente elementos que componen la solución, los cuales son clasificados por una función de costo. Para obtener diversas soluciones de buena calidad, cierto porcentaje de los mejores elementos son colocados en una lista restringida de candidatos (RCL, por sus siglas en inglés, *Restricted Candidate List*) ajustada por un parámetro (α). Finalizada la fase constructiva se tiene una solución para el problema, la cual puede ser factible o infactible. De tal forma que en una segunda fase (la fase de búsqueda local) se tiene como objetivo reparar dicha factibilidad y mejorar la solución obtenida. Esta fase consiste básicamente en realizar

una serie de movimientos en la solución definidos en uno o varios vecindarios.

Para poder implementar el método de solución propuesto es necesario llevar a cabo una etapa de pre procesamiento en donde la información de las demandas de los centros de distribución (n_{ip}^{-}) y las capacidades de las plantas (n_{ip}^{+}) son asignadas a través de la resolución de un problema de asignación a un conjunto de pedidos $(i, j, p, r_{ijp}) \in R$. Un pedido se refiere a la cantidad de cajas del producto p que se planea enviar desde la planta i al centro de distribución j. Al ya tener definido previamente esta asignación es posible tratar el problema como si fuese un problema de carga y descarga de productos con emparejamientos.

Para resolver el problema planteado se ha propuesto un procedimiento basado en GRASP que consta básicamente de los siguientes componentes:

Cálculo de matriz de tiempos y costos: Este proceso del algoritmo tiene la finalidad de determinar el costo y el tiempo de traslado desde cualquier nodo i hasta cualquier nodo j del grafo. Este proceso se ejecuta una única vez y se refiere al Paso 2 del Algoritmo 1.

Generación de pedidos: Este proceso del algoritmo tiene la finalidad de construir diferentes pedidos para tratar al problema como si fuese un problema de PDP con emparejamientos. Este procedimiento es ejecutado un número determinado de veces para darle diversidad al algoritmo. Dicha diversidad está en función de un parámetro (β). Este proceso se refiere al Paso 5 del Algoritmo 1.

Fase constructiva: Este procedimiento forma parte del procedimiento GRASP el cual corresponde en su forma más básica desde el Paso 6 hasta el Paso 12 del Algoritmo 1. En el procedimiento de la fase constructiva se genera una solución al insertar elementos a la solución final a través de una lista restringida de candidatos. Este proceso se refiere al Paso 7 del Algoritmo 1.

Fase de búsqueda local: Este procedimiento forma parte del procedimiento GRASP, el cual, al recibir una solución realiza una serie de movimientos definidos en uno o varios vecindarios para mejorar la solución. Este proceso se refiere al Paso 8 del Algoritmo 1.

Algoritmo 1 solutionProcess($\Delta\beta$,α,$maxIterations$)

Input:

$\Delta\beta$:= Parámetro para la generación de pedidos.

α := Parámetro para la lista restringida de candidatos en la fase constructiva.

$maxIterations$:= Número de iteraciones.

1: $X_{\text{best}} \leftarrow \emptyset$; $f(X_{\text{best}}) \leftarrow +\infty$
2: $solveFloyd_Warshall(G, C^s, C^d)$
3: $\beta \leftarrow 0$
4: **while** ($\beta \leq 1$) **do**
5: $R \leftarrow solve\ AP\ (\beta)$
6: **for** $iter = 1$ to $maxIterations$ **do**
7: $X \leftarrow constructSolution(\alpha, R)$
8: $X \leftarrow localSearch(X)$
9: **if** (Xes mejor que X_{best}) **then**
10: $X_{\text{best}} \leftarrow X$
11: **end if**
12: **end for**
13: $\beta \leftarrow \beta + \Delta\beta$
14: **end while**

return X_{best}

En cada iteración se construye una nueva solución y ésta es comparada contra la mejor solución encontrada como se muestra en el Paso 9 del Algoritmo 1. Cabe aclarar que esta comparación primeramente evalúa la factibilidad de la solución y posteriormente, en caso de igual factibilidad, evalúa el valor de la función objetivo. A continuación se explica detalladamente cada uno de estos componentes que conforman el algoritmo.

4.1 Construcción de matriz de costos y tiempos

Esta etapa tiene por objetivo construir las matrices de tiempos y costos de traslados entre dos puntos para cualquier tipo de vehículo buscando la ruta más corta entre estos puntos. Para ello se vale del algoritmo de Floyd-Warshall [55] para encontrar la ruta más corta entre todos los nodos del problema.

Se asume que se recibe un grafo no dirigido donde las aristas poseen dos tipos de costos ($c^s_{ij} \in C^s$ y $c^d_{ij} \in C^d$). Con dicha información se construye una matriz de costos C' utilizando el algoritmo de Floyd-Warshall encontrando la ruta más corta con respecto al precio entre dos nodos del grafo para cada uno de los vehículos. De tal modo que $c^s_{[ij]}$ es el costo más barato de viajar desde el nodo i hasta el nodo j utilizando un vehículo sencillo y de $c^d_{[ij]}$ es el costo más barato de viajar del nodo i al nodo j utilizando un vehículo doble. La ruta que sigue el vehículo en la red para realizar el viaje del nodo i al nodo j utilizando la ruta más corta está definida como $m^s_{[ij]} = \langle i, \ldots, j \rangle$ y $m^d_{[ij]}$ para un vehículo sencillo y un vehículo doble, respectivamente. El tiempo para realizar dicha ruta está definido como $s^s_{[ij]}$ y $s^d_{[ij]}$ para un vehículo sencillo y un vehículo doble, respectivamente, y se obtiene calculando la suma de las rutas parciales de $m^s_{[ij]}$ y $m^d_{[ij]}$.

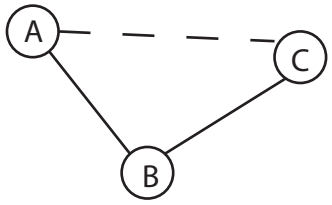


Figura 4.1: Construcción de matriz de costos.

Para tener más claro este concepto, se ilustra con un grafo compuesto por tres nodos $\{A, B, C\}$ y dos aristas $\{AB, BC\}$ como se muestra en la Figura 4.1. El costo de viajar desde el nodo A hasta el nodo C sería la suma del costo de $AB + BC$ (para cada tipo correspondiente de vehículo) y de igual manera el tiempo de viaje sería la suma de los tiempos de $AB + BC$. En el problema sería válido aplicar este

razonamiento puesto que si el vehículo quisiera viajar desde el nodo A hasta el nodo C tendría que viajar de A a B y de B a C sin detenerse en el nodo B por lo que no incurriría en el tiempo en atender al nodo B ni utilizaría ninguno de los andenes del nodo B.

4.2 Generación de pedidos

En esta etapa se asignan los productos que se enviarán desde el nodo i hasta el nodo j resolviendo un problema de asignación que se explica a continuación. En esta asignación se determina la cantidad r_{ijp} de producto p que la planta i atenderá al centro de distribución j.

En esta etapa de pre procesamiento se generan los pedidos a partir de la información de las capacidades y demandas de las plantas y centros de distribución respectivamente. Un pedido $(i, j, p, r_{ijp}) \in R$ lo definimos como una solicitud de r_{ijp} cajas de producto p que se recogerá en el nodo i para ser entregado en el nodo j. Para generar estos pedidos se propone resolver un problema de asignación cuyo modelo se expone a continuación:

Conjuntos

V Conjunto de nodos.

P Conjunto de tipos de productos.

Parámetros

β Parámetro para ponderar el costo de una arista.

$c^s_{[ij]}$ Costo de viajar desde el nodo i hasta le nodo j utilizando un vehículo sencillo.

$c^d_{[ij]}$ Costo de viajar desde el nodo i hasta le nodo j utilizando un vehículo doble.

n^+_{ip} Cantidad de cajas del producto p que pueden recogerse del nodo i.

n^-_{ip} Cantidad de cajas del producto p que deben de entregarse al nodo i.

Variables

r_{ijp} Cantidad de producto p que se recogerá en el nodo i y se entregara en el nodo j.

y_{ij} Variable binaria que toma valor de uno en caso de que se recoja producto en el nodo i para ser entregado en el nodo j, cero en otro caso.

Modelo

$$AP(\beta) \quad \text{Minimizar} \quad \sum_{i,j \in V} \left(\beta c^s_{[ij]} + (1-\beta) c^d_{[ij]}\right) y_{ij} \tag{4.1}$$

$$\text{Sujeto a:} \quad \sum_{j \in V} r_{ijp} \leq n^+_{ip} \qquad i \in V, p \in P \tag{4.2}$$

$$\sum_{i \in V} r_{ijp} \geq n^-_{jp} \qquad j \in V, p \in P \tag{4.3}$$

$$\sum_{p \in P} r_{ijp} \leq M y_{ij} \qquad i, j \in V \tag{4.4}$$

$$r_{ijp} \geq 0 \qquad i, j \in V, p \in P \tag{4.5}$$

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

El objetivo (4.1) es minimizar el costo relacionado por el uso de las aristas para trasladar el producto del nodo i al j. El parámetro β en la función objetivo tiene la función de ponderar la relevancia entre los costos para los vehículos sencillos y los vehículos dobles. Las restricciones (4.2) restringen que no se pueda exceder la capacidad de la planta. Las restricciones (4.3) restringen el cumplimiento de la demanda de los centros de distribución. Finalmente en las restricciones (4.4) se indica si se asignan pedidos que van desde el nodo i hasta el nodo j, donde M es un número lo suficientemente grande tal que en caso de que y_{ij} tome valor de uno la restricción siempre se cumpla. Un valor de M podría ser la máxima capacidad de una planta ya que no es posible enviar más producto del que se tiene disponible de tal modo que M podría expresarse como $M = \max_{i \in V, p \in P} \left\{n^+_{ip}\right\}$. Finalmente las restricciones (4.5) y (4.6) establecen la condición de no negatividad e integralidad, respectivamente, de las variables de decisión.

Este modelo matemático es muy similar al problema de asignación y éste es un modelo fácil de resolver. En la evaluación empírica, usualmente este modelo

matemático era resuelto en cuestión de segundos para las instancias de tamaño real, utilizando el método de ramificación y acotamiento implementado por CPLEX.

4.3 Fase constructiva

En esta etapa del algoritmo es donde se construye una solución (X). Para ello en una primera parte se construyen todos los vehículos que se van a poder llegar a usar, posteriormente se asignan los requerimientos que atenderán los vehículos por lo que también se asigna la ruta que seguirán dichos vehículos. Después debido a la forma de cómo se realizan las inserciones de los requerimientos a los vehículos se requiere un proceso adicional para regresar los vehículos a su lugar de origen para cumplir con el criterio de que cada vehículo al concluir su jornada deben de regresar a su lugar de origen. Cabe mencionar que en la fase constructiva no se toma en cuenta el criterio de no exceder el número de vehículos que puede atender simultáneamente un centro de distribución, este criterio es considerado en la fase de búsqueda local, el cual buscará reparar la solución para que este criterio no sea violado.

Para la formulación del algoritmo se establece que $\mathrm{X}_{\Pi} = \{\pi_k | k \in K\}$ es una colección de elementos o rutas donde $\pi_k = \langle \ell_1, ..., \ell_n \rangle$ es un conjunto de elementos ordenados que representan los nodos visitados por el vehículo k, siendo el primer elemento el nodo origen del vehículo k y el último elemento la posición final del vehículo, tal como puede verse en la Figura 4.2.

Con este elemento además de saber la ruta del vehículo es posible determinar el costo de la ruta, el tiempo que se requiere en realizar la ruta y la posición actual del vehículo. Es indispensable saber esto para la construcción de la solución.

En el Algoritmo 2 se muestra de manera general el procedimiento de la fase constructiva. A continuación se explican cada uno de los componentes que lo conforman.

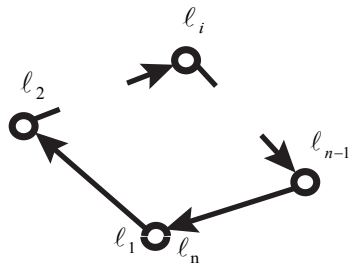


Figura 4.2: Ejemplo de una ruta de un vehículo.

Algoritmo 2 constructSolution(α,R)

1: $X_\Pi \leftarrow \emptyset$
2: $K \leftarrow T$
3: $R' = \{\langle i, j\rangle : (i, j, p, r_{ijp}) \in R\}$
4: $\bar{P} \leftarrow getFeasiblePaths(K, R')$
5: **while** $R \neq \emptyset$ **do**
6: $\quad \varphi^{\text{mín}} \leftarrow \text{mín}\left\{\varphi(i, j, k) : (i, j, k) \in \bar{P}\right\}$
7: $\quad \varphi^{\text{máx}} \leftarrow \text{máx}\left\{\varphi(i, j, k) : (i, j, k) \in \bar{P}\right\}$
8: $\quad RCL = \left\{(i, j, k) \in \bar{P} : \varphi(i, j, k) \leq \varphi^{\text{mín}} + \alpha(\varphi^{\text{máx}} - \varphi^{\text{mín}})\right\}$
9: $\quad (i, j, k) \leftarrow$ elegir un elemento aleatoriamente de RCL
10: $\quad (X_k, R) \leftarrow assignProducts(i, j, k, R, X)$
11: **end while**
12: $returnToDepot(X_\Pi)$
return X

4.3.1 Inicialización de vehículos

Inicialmente cada tráiler ($t \in T$) es asignado a un vehículo ($k \in K_{\sigma(t)}$). Esto es posible porque parte de los supuestos del problema es que existe una cabina disponible para cada tráiler, de tal forma que cada tráiler será asignado a un solo

vehículo y así las ventanas de tiempo y las capacidades de cada vehículo serán las mismas que el tráiler asignado. Cabe mencionar que en una solución no necesariamente se deban usar todos los vehículos. En una etapa posterior se buscará reducir la cantidad de vehículos ya sea convirtiéndolos en vehículos dobles o negar el uso de algún vehículo. Esta parte del algoritmo corresponde al Paso 2 del Algoritmo 2.

4.3.2 Asignar requerimientos a los vehículos

En esta etapa es donde se asignan los pedidos a los vehículos y por consiguiente donde se lleva acabo el ruteo. Esta etapa es la parte medular de la fase constructiva y consiste en generar las rutas de los vehículos al mismo tiempo, es decir, se construyen todas las rutas de manera paralela. La manera de cómo asignar los pedidos a los vehículos es determinar el vehículo k y la ruta (i,j) que va tomar este vehículo para después asignarle los productos que entregara en esta ruta. Este proceso se sigue ejecutando hasta que se hayan asignado todo los requerimientos del problema a los vehículos.

Inicialmente se construye un conjunto de rutas factibles definido como $\bar{P} = \{(i,j,k) : (i,j) \in R' \wedge (i,j,k) \text{ es una ruta factible}\}$, tal que $(i,j,k) \in \bar{P}$ es un elemento que indica si el vehículo k le es posible realizar un pedido donde se recoja producto en el nodo i y se entregue el producto en el nodo j. Para que sea posible que el vehículo pueda realizar esos pedidos se requiere que el vehículo pueda viajar de i a j sin que se violen las ventanas de tiempo por lo que es necesario que se cumplan con las siguientes características tomando como referencia el tiempo que ha utilizado el vehículo para realizar sus entregas previas:

1. Que le sea posible llegar al nodo i sin que la ventana de tiempo del nodo i sea violada.

2. Que le sea posible viajar del nodo i al nodo j sin que la ventana de tiempo del nodo j sea violada.

3. Que le sea posible regresarse a su ubicación inicial sin que la ventana de tiempo de la ubicación inicial y del vehículo sean violadas.

El cumplimiento de estas características hace que esa ruta se le considere una ruta factible, de tal forma que en el Paso 4 del Algoritmo 2 se generan todas las rutas factibles, además se establece que $\bar{P}_k \subseteq \bar{P}$ son las rutas factibles del vehículo k.

Obtención de ruta

La etapa de obtención de ruta es donde se determina que vehículo se va usar y hacia donde se va a dirigir. Cabe aclarar que en esta etapa no se asignan los productos que el vehículo trasladará ya que eso se realiza en una etapa posterior.

En esta etapa se busca determinar la ruta (i, j, k) de tal modo que el vehículo k realice entregas que inician en i y terminan en j, por lo que, para seleccionar esta ruta se selecciona una ruta de manera aleatoria cuyo costo $\varphi\,(i, j, k)$ esté a un $\alpha\,\%$ del costo de la ruta factible de menor costo.

El costo proyectado de la ruta (función de mérito de la fase constructiva) esta definida por la ecuación (4.7).

$$\varphi(i,j,k) = \left(1 - v_k^{\text{type}}\right)\left(c^s_{[u(k)i]} + c^s_{[ij]} + c^s_{[j\sigma(k)]}\right) + v_k^{\text{type}}\left(c^d_{[u(k)i]} + c^d_{[ij]} + c^d_{[j\sigma(k)]}\right) \quad (4.7)$$

Consiste en el costo de viajar desde la ubicación actual del vehículo $u(k)$ hacia la ubicación i (donde se recoge el producto) más el costo de viajar de la ubicación i a la ubicación j (donde se entrega el producto) más el costo de regresar de la ubicación j a la ubicación inicial del vehículo $\sigma(k)$, que está en función a la configuración del vehículo k ya sea doble ($v_k^{\text{type}} = 1$) o sencillo ($v_k^{\text{type}} = 0$).

Al tener el costo proyectado de las rutas factibles y siguiendo la filosofía de la metaheurística GRASP, se genera una lista restringida de candidatos (RCL), que contiene únicamente los mejores elementos, es decir, aquellos elementos cuya función

de mérito esté dentro de un $\alpha\,\%$ del mejor costo posible. Esta parte corresponde al Paso 8 del Algoritmo 2.

Posteriormente se elige un elemento de la RCL de manera aleatoria. Esta parte corresponde al Paso 9 del Algoritmo 2. A diferencia de la metodología puramente voraz, donde se selecciona al mejor candidato, la metaheurística GRASP permite seleccionar entre los mejores candidatos, dando diversidad en las soluciones y obteniendo mejores resultados en la práctica que los procedimientos puramente voraces.

Asignación de productos

Una vez que se selecciona la ruta (i, j, k) que va seguir el vehículo k, se le asignan todos los productos hasta llenar la capacidad del vehículo o hasta que ya no haya requerimientos que se realicen de i a j, utilizando el siguiente criterio: Primero asigna los productos no retornables (R_n^{ij}) sin que se exceda el número máximo de tarimas de producto no retornable. Este proceso corresponde a los pasos del 6 al 17 del Algoritmo 3, en el cual básicamente se selecciona un pedido (Paso 9) del conjunto de pedidos no asignados y se prosigue a evaluar si es posible que el vehículo puede realizarlo en su totalidad (Paso 10). En caso de no ser posible, solamente se asigna la mayor cantidad de cajas que es posible atender (Paso 14) y se devuelve el resto del pedido al conjunto de pedidos no asignados (Paso 14). Este proceso se repetirá mientras no se haya excedido el número máximo de tarimas del producto no retornable ($capacity > 0$) y todavía existan requerimientos sin asignar con origen en el nodo i y destino en el nodo j de los productos no retornables ($R_n^{ij} \neq \emptyset$). Este proceso corresponde al Paso 8 del Algoritmo 3.

Al terminar la asignación de los productos no retornables se termina de llenar el vehículo con los productos retornables (R_r^{ij}). Este proceso corresponde al Paso 18 al 29 del Algoritmo 3. Dicho proceso se realiza de manera similar a la asignación de producto no retornable descrito anteriormente. Al generar las rutas de esta forma se garantiza que siempre se cumpla la restricción de que no se aplasten los productos

no retornables.

Para la formulación de esta sección se establece que $\pi_k = \langle \ell_1, \ell_2, \ldots, \ell_n \rangle$ es la ruta que sigue el vehículo k aclarando que en la ruta solo se establecen los nodos donde se carga y descarga producto. Dado que se tiene la ruta que debe de seguir el vehículo utilizando la ruta más corta $(m^s_{[ij]}, m^d_{[ij]})$, fácilmente se obtiene la ruta real del vehículo. $g^+_{[n]kp}$ es la cantidad de producto p que recoge el vehículo k en el n-ésimo nodo de su ruta y de manera homóloga $g^-_{[n]kp}$ es la cantidad de producto p que descarga el vehículo k en el n-ésimo nodo de su ruta. Finalmente $h(k) = \{t \in T : w_{kt} = 1\}$ es una función que esta definida como el conjunto de tráileres que conforman al vehículo k.

Algoritmo 3 assignProducts(i, j, k, R, X)

Input: $\varphi = (i, j, k)\,, \quad \pi_k = \langle \ell_1, \ldots, \ell_n \rangle$

if $(\ell_n \neq i)$ **then**
  $\pi_k \leftarrow \langle \ell_1, \ldots, \ell_n, i, j \rangle$
else
  $\pi_k \leftarrow \langle \ell_1, \ldots, \ell_n, j \rangle$
end if
$capacity \leftarrow v_k^{\text{below}} + \sum_{t \in h(k) \cap T^e} c_t^{\text{above}}$
$R^{ij}_n = \{(i, j, p, r_{ijp}) \in R : p \in P^n\}$
while $(capacity > 0 \wedge R^{ij}_n \neq \emptyset)$ **do**
  $r \leftarrow$ elegir un elemento de R^{ij}_n; $R^{ij}_n \leftarrow R^{ij}_n \backslash \{r\}$
  if $(o_p \cdot capacity \geq r_{ijp})$ **then**
    $g^+_{ikp} = r_{ijp}; R \leftarrow R \backslash \{r\}$
    $capacity \leftarrow capacity - r_{ijp}/o_p$
  else
    $g^+_{ikp} = o_p \cdot capacity; R \leftarrow (R \backslash \{r\}) \cup \{(i, j, p, r_{ijp} - o_p \cdot capacity)\}$
    $capacity \leftarrow 0$
  end if

17: **end while**

18: $capacity \leftarrow v_k^{\text{below}} + v_k^{\text{above}} - \sum_{p \in P_n} g_{ijp}^{+}$

19: $R_r^{ij} = \{(i, j, p, r_{ijp}) \in R : p \in P^r\}$

20: **while** $(capacity > 0 \wedge R_r^{ij} \neq \emptyset)$ **do**

21: $r \leftarrow$ elegir un elemento de $R_r^{ij}; R_r^{ij} \leftarrow R_r^{ij} \setminus \{r\}$

22: **if** $(o_p \cdot capacity \geq r_{ijp})$ **then**

23: $g_{ikp}^{+} = r_{ijp}; R \leftarrow R \setminus \{r\}$

24: $capacity \leftarrow capacity - r_{ijp}/o_p$

25: **else**

26: $g_{ikp}^{+} = o_p \cdot capacity; R \leftarrow (R \setminus \{r\}) \cup \{(i, j, p, r_{ijp} - o_p \cdot capacity)\}$

27: $capacity \leftarrow 0$

28: **end if**

29: **end while**

30: $\bar{P}_k \leftarrow \{(i, j, k) : (i, j) \in R' \wedge (i, j, k)$ es una ruta factible$\}$

return (X_k, R)

4.3.3 Devolver los vehículos a su lugar de origen

Una vez asignados todos los requerimientos. Los vehículos son devueltos a su lugar de origen. Cabe recalcar que en el proceso de asignación de los requerimientos se ha garantizado que todo vehículo le es posible regresar a su lugar de origen sin violar las ventanas de tiempo.

Algoritmo 4 returnToDepot(X_Π))

Input: $\pi_k = \langle \ell_1, \ldots, \ell_n \rangle, \quad \pi_k \in X_\Pi$

1: **for** $(\pi_k \in X_\Pi)$ **do**

2: **if** $(\ell_n \neq \sigma(k))$ **then**

3: $\pi_k \leftarrow \langle \ell_1, \ldots, \ell_n, \sigma(k) \rangle$

4: **end if**

5: **end for**

return

4.4 Fase de mejora

Al finalizar la fase constructiva se cuenta con una solución donde se tienen asignados todos los pedidos a los vehículos. Cabe mencionar que se cuenta con una solución donde únicamente fueron utilizados vehículos sencillos y dicha solución cumple con todas las restricciones del problema salvo las restricciones del cumplimiento de los andenes.

Es por ello que en la fase de búsqueda local se pretende mejorar el valor de la función objetivo y reparar factibilidad de la solución, para ello en una primera etapa se analiza existen vehículos sencillos que tengan las mismas rutas y se busca verificar que, en el supuesto de enviar estos vehículos como un solo vehículo doble fuese posible reducir los costos sin violar las restricciones de las ventanas de tiempo. Esta etapa corresponde al Paso 1 del Algoritmo 5.

Posteriormente se propone verificar si es posible que un vehículo que terminó a tiempo sus entregas le sea posible realizar las entregas de otro vehículo. Para ello se analizan todos los vehículos que iniciaron en la misma planta y se determina que, si al incorporarle las nuevas entregas, este vehículo puede regresar a su lugar de origen sin que viole las ventanas de tiempo. Si es posible, se transfiere la entrega reduciendo con esto el costo asociado al uso de un vehículo. Esta estapa corresponde al Paso 2 del Algoritmo 5.

Finalmente se cuenta con una última etapa donde se busca asignar un tiempo de espera a los vehículos a lo largo de su recorrido con el fin de que le sea posible cumplir sus entregas sin exceder la cantidad de andenes disponibles. Esta estapa corresponde al Paso 3 del Algoritmo 5.

Algoritmo 5 localSearch()

1: $convertToDoubles()$

2: $transferShipment()$

3: $repairDocks()$

return

A continuación se explica detalladamente cada uno de los procesos propuestos en la búsqueda local.

4.4.1 Convertir vehículos a dobles

En esta etapa se analiza si existen vehículos sencillos (k_1 y k_2) que tienen las mismas rutas ($\pi_{k_1} = \pi_{k_2}$). Como se puede ver en la Figura 4.3, lo que se busca es verificar que, en el supuesto de enviar a estos vehículos como un solo vehículo doble, le fuese posible reducir costos sin violar las restricciones de las ventanas de tiempo.

Para esto se define un vecindario $N_1(S)$ que consiste en sustituir a dos vehículos sencillos que inicien en la misma planta y tengan la misma ruta (como se puede ver en el Paso 1 del Algoritmo 6), por un vehículo doble que realice las entregas de ambos vehículos y su ruta sea factible. Para ello a partir de la información de los dos vehículos sencillos se construye un vehículo doble ficticio u' cuya ruta es la misma que los vehículos sencillos (como se puede ver en el Paso 5).

$$merge(u, v) = \pi_{u'} | \pi_{u'} = \pi_u \wedge v^t_{u'} = 1$$

Posteriormente se evalúa si este nuevo vehículo proporcionaría alguna mejora en la función objetivo y además si este nuevo vehículo posee una ruta factible (como puede verse en el Paso 6). La función de mérito que se utiliza para evaluar el movimiento la definimos como la suma de los costos asociados a los dos vehículos sencillos (k_1 y k_2) menos el costo asociado al uso del vehículo doble, siendo este costo el beneficio potencial de realizar dicho cambio.

$$f\left(\pi_u = \langle \ell_1, ..., \ell_n \rangle\right) = \sum_{i=1}^{n-1} \left(\left(1 - v_k^{\text{type}}\right) c^s_{[\ell_i \ell_{i+1}]} + v^t_k c^d_{[\ell_i \ell_{i+1}]} \right)$$

La estrategia usada en esta búsqueda local es la estrategia de primera mejora con la premisa de que se trata de rutas iguales y no es posible un beneficio mayor uniendo otros dos vehículos con la misma ruta.

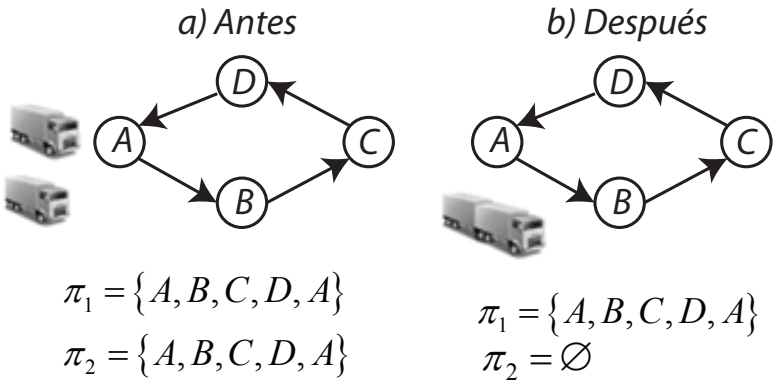


Figura 4.3: Conversión de dos vehículos sencillos a uno doble.

Algoritmo 6 convertToDoubles(K)

$N_1 \leftarrow \{(k_1, k_2) \in K \times K : \pi_{k_1} = \pi_{k_2} \wedge v^t_{k_1} = v^t_{k_2} = 0\}$
while $|N_1| > 0$ **do**
  $(u, v) \leftarrow$ se escoge un elemento de N_1
  $N_1 \leftarrow N_1 \setminus \{(u, v)\}$
  $\pi_{u'} \leftarrow merge\,(u, v)$
  if $\pi_{u'}$ es una ruta factible y $f(\pi_{u'}) < f(\pi_u) + f(\pi_v)$ **then**
    $h(u) \leftarrow h(u) \cup h(v)$
    $K \leftarrow K \setminus \{v\}$
    $N_1 \leftarrow N_1 \setminus \{(k_1, k_2) : k_1 \in \{u, v\} \vee k_2 \in \{u, v\}\}$
  end if
end while
return

4.4.2 Transferir entregas

En esta etapa se analiza si existe algún vehículo (k_1) que le sea posible de realizar sus entregas y después realizar las entregas de algún otro vehículo (k_2) que

comparta la misma planta inicial, para que se le transfiera las entregas de este vehículo y así ahorrarse el costo de utilizar dos vehículos siendo que se es posible realizar las entregas en un solo vehículo como se puede visualizar en la Figura 4.4. Para esto se define un vecindario $N_2(S)$ que consiste en todas las soluciones alcanzables a partir de la solución S al sustituir dos vehículos del mismo tipo de vehículo (k_1 y k_2) que compartan la misma planta (como se puede ver en el Paso 1 del Algoritmo 7), por un vehículo (u') del mismo tipo de vehículo que realice las entregas de k_1 y k_2 y que la ruta resultante no viole las restricciones de las ventanas de tiempo. Para ello primeramente se selecciona un par de vehículos con estas características (como puede verse en el Paso 3).

A partir de la información se construye un vehículo ficticio u' que realiza primero las entregas del vehículo u y después las entregas del vehículo v, para la construcción del vehículo u' se analiza si es posible al término de las entregas de u realizar inmediatamente las entregas de v sin necesidad de pasar al nodo origen. Para ello es necesario que tanto el vehículo u no tenga que entregar producto en la planta de origen ni el vehículo v tenga que recoger producto al inicio de su recorrido (como puede verse en el Paso 5). De otro modo se construye un vehículo que realiza las entregas de u y de v pasando por el nodo origen (como se puede ver en el Paso 8).

Posteriormente se evalúa si este nuevo vehículo proporcionaría alguna mejora en la función objetivo y además si este nuevo vehículo posee una ruta factible (como puede verse en el Paso 10). La función de mérito que se utiliza para evaluar el movimiento la definimos como la suma de los costos asociados a los dos vehículos originales menos el costo asociado al uso del vehículo resultante, siendo este costo el beneficio potencial de realizar dicho cambio. La estrategia usada en esta búsqueda local es la estrategia de primera mejora.

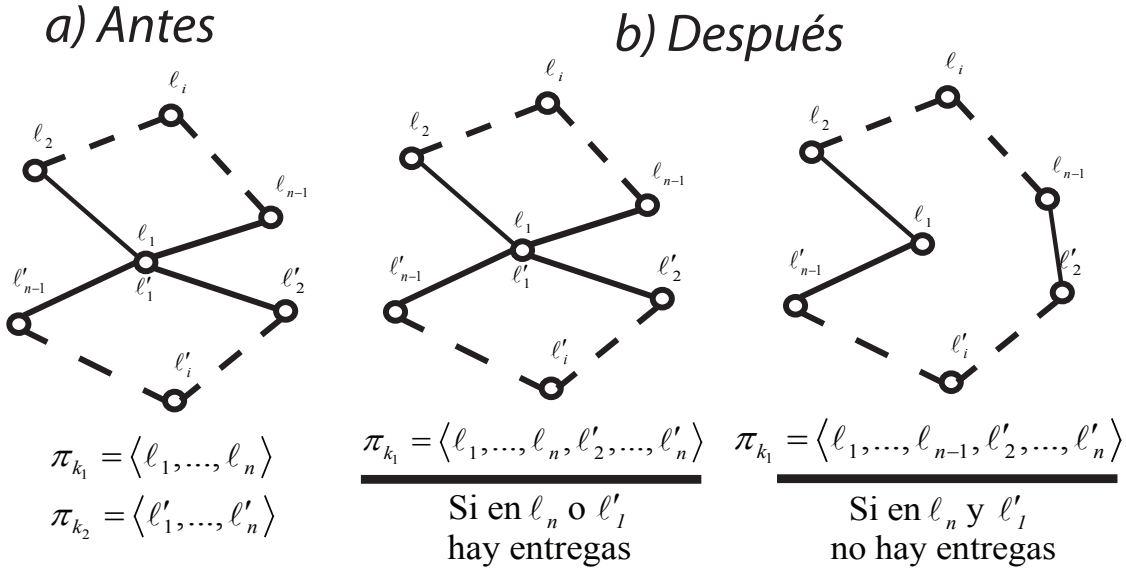


Figura 4.4: Transferir entregas de un vehículo a otro.

Algoritmo 7 transferShipment()

$N_2 = \left\{(k_1, k_2) \in K \times K : \sigma(k_1) = \sigma(k_2) \wedge v^t_{k_1} = v^t_{k_2}\right\}$
while $|N_2| > 0$ **do**
  $(u, v) \leftarrow$ escoger un elemento de N_2; $N_2 \leftarrow N_2 \backslash (u, v)$
  $\pi_u = \langle \ell_1, \ldots, \ell_n \rangle$; $\pi_v = \langle \ell'_1, \ldots, \ell'_n \rangle$
  if $\left(\sum_{p \in P} g^-_{\ell_n u p} = 0 \vee \sum_{p \in P} g^+_{\ell'_1 v p} = 0 \right)$ **then**
    $\pi_{u'} = \langle \ell_1, \ldots, \ell_n, \ell'_2, \ldots, l'_n \rangle$
  else
    $\pi_{u'} = \langle \ell_1, \ldots, \ell_{n-1}, \ell'_2, \ldots, l'_n \rangle$
  end if
  if $\pi_{u'}$ es una ruta factible y $f(\pi_{u'}) < f(\pi_u) + f(\pi_v)$ **then**
    $\pi_u \leftarrow \pi_{u'}$
    $K \leftarrow K \backslash \{v\}$
    $N_2 \leftarrow N_2 \backslash \{(k_1, k_2) : k_1 = v \vee k_2 = v\}$
  end if
end while
return

4.4.3 Reparar andenes

En esta etapa lo que se busca es cumplir con el criterio de que no lleguen más vehículos que los andenes que se tienen disponibles en los centros de distribución y las plantas como se puede ver en la Figura 4.5. Para esto se analiza en cada hora h los nodos que tienen más arribos que los andenes disponibles, para determinar que vehículos tienen que esperarse para que todos puedan ser atendidos en los centros de distribución. Para ello se establece la función $J(h,i)$ que define a todos los vehículos que se encuentran en la hora h en el nodo i definida como:

$$J(h,i) = \{k \in K : \lfloor s_{ik}/60 \rfloor = h\}$$

Con esto es posible determinar un conjunto de nodos $\Psi(h)$ cuya cantidad de vehículos sobrepasen la cantidad de andenes existentes tal que :

$$\Psi(h) = \{i \in V : f_{ih} < |J(h,i)|\}$$

Para determinar el vehículo que se le tendrá que asignar un tiempo de espera se utiliza la función $\Upsilon(i,k)$ que determina el máximo tiempo que se le puede asignar a un vehículo sin que este viole alguna restricción de ventanas de tiempo. En la Figura 4.6 se muestra el máximo tiempo que se le puede asignar a un vehículo para no violar las restricciones de las ventanas de tiempo.

$$\Upsilon(i,k) = \min_{u \in [i,..,\ell_n]} \{b_u - s_{uk}\}$$

De tal forma que al vehículo que se le asigne un tiempo de espera será aquel vehículo que se encuentre más holgado de tiempo.

Este procedimiento de reparar la factibilidad del uso de los andenes tiene lugar al final de la fase de construcción y mejora por el motivo de que en cada una de las etapas anteriores se modifican las posibles rutas de los vehículos y esto provoca que cada movimiento realizado altere la cantidad de llegadas a los nodos y no tenga mucho sentido tomar en cuentra esta factibilidad para algo que posiblemente vaya a cambiar a través de alguno de estos movimientos.

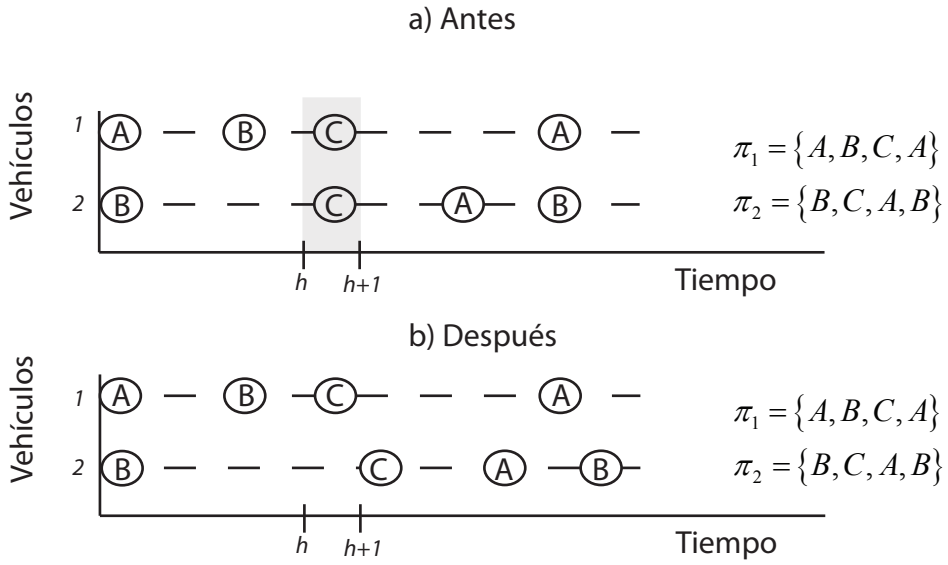


Figura 4.5: Reparar factibilidad por uso de andenes.

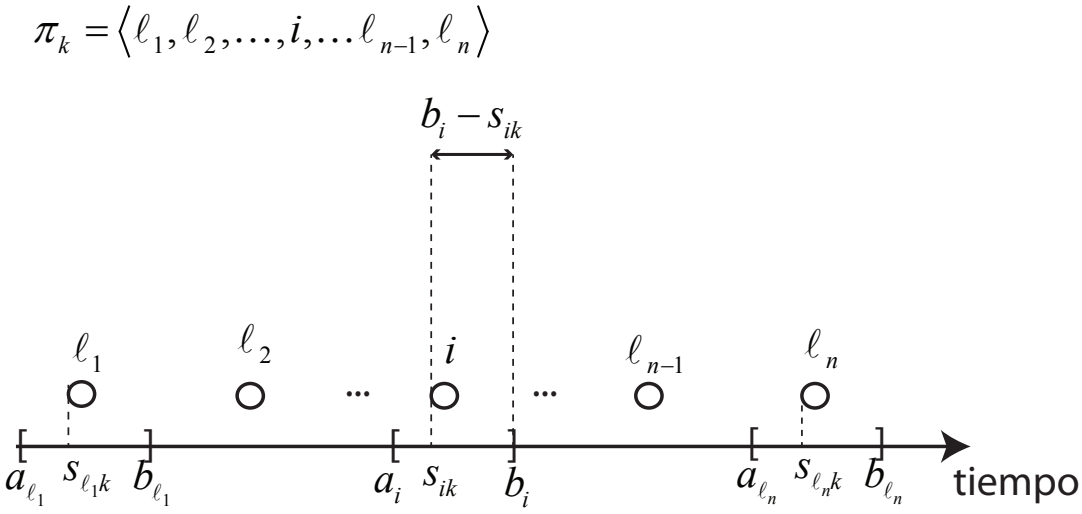


Figura 4.6: Criterio para reparar factibilidad por uso de andenes.

Algoritmo 8 repairDocks()

Input: $\pi_k = \langle \ell_1, \ldots, \ell_n \rangle$

for $h = 0$ to $|H|$ **do**
  while $(\Psi(h) \neq \emptyset)$ **do**
    $i \leftarrow$ elegir un elemento de $\Psi(h)$ de manera aleatoria
    while $(|J(h,i)| > f_{ih})$ **do**
      $k \leftarrow \arg\max_k \{\Upsilon(i,k)\}$
      for all $(u \in [i, \ldots, \ell_n])$ **do**
        $s_{uk} = s_{uk} + 60 - (s_{ik} - 60 \cdot \lfloor s_{ik}/60 \rfloor)$
      end for
    end while
  end while
end for

return

Capítulo 5

Evaluación Empirica

La evaluación de una metodología de solución para un determinado problema implica el desarrollo de una evaluación empírica con la finalidad de poder garantizar el desempeño satisfactorio del método desarrollado. Para ello es necesario realizar un estudio computacional y poder determinar si el algoritmo propuesto cumple con las expectativas planteadas inicialmente.

En este estudio, se propuso un algoritmo heurístico, lo cual implica que no es posible garantizar soluciones óptimas por lo que es necesario que las soluciones sean evaluadas de forma estadística o comparativa. Para esto, por lo general se establece una serie de experimentos controlados utilizando un conjunto de datos de prueba.

En este capítulo se realiza un estudio computacional para la evaluación del comportamiento de la metodología propuesta que fue desarrollada en esta tesis. Inicialmente se plantean los objetivos que se pretenden lograr a partir del estudio computacional. Posteriormente se explica la naturaleza de las instancias que fueron utilizadas para la realización de las pruebas así como los aspectos que fueron considerados en la evaluación. Finalmente se explican los distintos experimentos realizados, así como los resultados obtenidos.

5.1 Desarrollo experimental

En esta sección se exponen detalladamente los alcances del estudio computacional así como las especificaciones y condiciones en las que fueron realizadas. Esto es de mucha importancia para poder realizar una comparación en el desempeño con otras metodologías que aborden el mismo problema.

5.1.1 Objetivos

Los objetivos que se pretenden alcanzar con este estudio computacional son los siguientes:

1. Ajustar adecuadamente los valores para el parámetro $\Delta\beta$ (que tiene la función de determinar el grado de importancia que le da a los viajes sencillos y dobles, especificado en la Sección 4.2) y el parámetro α (que ajusta el tamaño de la lista de candidatos en la fase constructiva, especificado en la Sección 4.3.2). Esto permitirá alcanzar soluciones de mayor calidad en la metodología propuesta.

2. Determinar la eficiencia del algoritmo desarrollado determinando los tiempos de ejecución, el índice de factibilidad así como el impacto que tienen los métodos de mejora propuestos.

3. Detectar las debilidades y fortalezas del algoritmo analizando su comportamiento bajo diversas condiciones.

4. Resolver un caso de estudio de la industria comparando los resultados entre el procedimiento de la empresa y la metodología propuesta en este estudio.

5.1.2 Condiciones experimentales

Las condiciones experimentales pueden afectar directamente en los resultados de un experimento computacional. Estas condiciones deben de ser documentadas con

la finalidad de que este experimento sea posible compararlo con otras metodologías. Dichas condiciones pueden deberse a factores relacionados con la naturaleza de los datos utilizados para llevar a cabo el experimento.

Otras condiciones que afectan directamente a los resultados son condiciones relacionadas con el ambiente computacional y comprenden principalmente las herramientas que se utilizaron para llevar a cabo la metodología de solución. Para este estudio todos los algoritmos fueron implementados en el lenguaje C# en el ambiente de Visual Studio 2010. Todos los experimentos fueron llevados a cabo en un procesador Intel Core i7 bajo el sistema operativo Windows 7. Cabe destacar que también fueron utilizadas las bibliotecas de optimización para .NET de CPLEX 11.2. CPLEX [33] es una de las implementaciones mas avanzadas a nivel mundial para resolver eficientemente problemas de programación lineal y programación lineal entera mixta. En la metodología CPLEX fue utilizado para resolver el problema de asignación expuesto en la Sección 4.2.

5.1.3 Generación de instancias del problema

La fase de experimentación se llevó a cabo mediante la ejecución del método aplicado a un número específico de instancias para evaluar el desempeño del algoritmo en diversas condiciones, como usualmente se hace. Una instancia de un problema es un conjunto de datos de entrada que satisface dicho problema, es decir, un caso particular del problema. Para que nuestros resultados puedan tener una interpretación realista es muy importante que las instancias del problema que se van a resolver representen lo mas fielmente posible a un problema real. Para ello, la empresa que motivó el presente estudio colaboró en proveer información importante sobre las características de los datos reales y con esto permitió identificar las características más importantes del problema y así poder definir el conjunto de instancias de prueba.

Para este experimento computacional se desarrolló una herramienta para ge-

nerar las instancias. Este programa recibe como entrada el número de tráileres que tiene la instancia, el número de centros de distribución y de plantas, así como los diferentes tipos de productos entre otras características que es posible ajustar a la herramienta. El generador de instancias a partir de esta información construye las instancias del problema y las almacena en archivos de texto con un formato adecuado que sirven como entrada para el algoritmo propuesto anteriormente.

Para la generación de las instancias que se consideraron los siguientes aspectos específicos:

Ubicación de los centros de distribución y de las plantas

Para localizar los nodos de la instancia se generaron los n nodos aleatoriamente en un plano cartesiano. Las coordenadas x y y que definen la ubicación de los nodos fueron generados con una distribución uniforme dentro de un plano de $[500 \times 500]$ unidades. Una vez teniendo la ubicación de los nodos, se prosiguió a seleccionar un subconjunto 5 nodos para ser considerados como plantas de abastecimiento. Este subconjunto de nodos se busca que estén ubicados lo mas dispersos posible. Existen diversos métodos de dispersión para seleccionar dichos nodos, la manera la cual fueron seleccionados fue a través de un método de clusterización. Para ello fue implementado el algoritmo de K-Means [43], que es un método de clusterización que tiene el objetivo de subdividir un conjunto de n nodos (observaciones) en k conjuntos tal que se minimice un criterio de dispersión, el cual usualmente es la suma de las inter-distancias entre los nodos pertenecientes a dicho conjunto. El algoritmo de K-Means internamente para su ejecución considera para cada conjunto un nodo representativo llamado centro. Para la selección de las plantas de abastecimiento, fueron utilizados dichos nodos representativos.

Estructura del grafo

Con la finalidad de representar fielmente a la realidad se generaron las aristas de la instancia de tal manera que estos formaran un grafo planar, es decir que las aristas que lo conforman no se cruzaran entre si. Además para que sea posible realizar las entregas se requiere que exista al menos una ruta para poder viajar de cualquier nodo i a cualquier nodo j, esto último conocido en la literatura como grafo conexo. Para la generación de las instancias se decidió crear tipos de grafos conocidos como grafos Gabriel [21] ya que estos tipos de grafos cumplen con los requerimientos anteriormente señalados.

Ubicación de los tráileres

Inicialmente los tráileres están ubicados en las plantas, para ello se establecieron aleatoriamente los tráileres en los nodos que fueron asignados como plantas con una distribución uniforme discreta con un rango de 0 a 5, el número de plantas. De igual forma la cantidad de tarimas en la parte superior y en la parte inferior fueron establecidas cada una de ellas con una distribución uniforme discreta con límite inferior y superior de 8 y 20 respectivamente. Así mismo la probabilidad de que un tráiler tenga entrepiso es del 50 %.

Distribución de los productos

Para la distribución de los productos en un número específico de iteraciones se seleccionó un producto al azar con una distribución uniforme discreta entre la cantidad total de productos de la instancia. También la cantidad de cajas a transportar de dicho producto fue establecido con una distribución normal con media $\mu = 50$ y varianza $\sigma^2 = 10$. Con esta información se seleccionó una planta y un centro de distribución con una distribución uniforme y se les asignaron la cantidad de cajas del producto como capacidad y como demanda respectivamente.

Cantidad de andenes

Para establecer la cantidad de andenes que pose una planta o un centro de distribución, se fijo la cantidad de andenes de cada nodo con una distribución uniforme discreta con limite inferior y superior de 4 y 8, respectivamente, en donde la cantidad de andenes que tiene un nodo es la misma para cualquier hora del día.

5.1.4 Características y parámetros de las instancias

Con la finalidad de garantizar una adecuada evaluación del algoritmo, es necesario que el algoritmo sea ejecutado bajo una gran cantidad de instancias con diferentes características, con el fin de detectar posibles cambios en el comportamiento del algoritmo y lograr así un conocimiento del mismo. La generación de instancias fue dada bajo las siguientes características que definen a cada instancia:

a) Instancias chicas: 10 nodos, 20 tráileres, 10 tipos de productos.

b) Instancias medianas: 15 nodos, 50 tráileres, 40 tipos de productos.

c) Instancias grandes: 25 nodos, 150 tráileres, 300 tipos de productos.

Además para cada tipo de instancias fueron generadas 20 réplicas diferentes dando un total de 60 instancias, las cuales fueron evaluadas en la experimentación del algoritmo.

5.1.5 Aspectos a evaluar del algoritmo

Las características que se pretenden evaluar con el estudio computacional comprenden los siguientes aspectos:

1. *Tiempo de ejecución.* Se busca determinar el tiempo necesario para resolver las instancias y que tanto impacto tiene la cantidad de nodos, vehículos y

productos en el tiempo de resolución. Este aspecto es de relevancia evaluarlo ya que es primordial para cualquier algoritmo obtener las soluciones en un tiempo razonable.

2. *Número de iteraciones*. Se busca determinar la cantidad de veces que es necesario construir las soluciones para que el algoritmo converja a la mejor solución alcanzada en un tiempo razonable sin comprometer la calidad de las soluciones.

3. *Mejoría en la búsqueda local*. Es necesario evaluar que tanto impacto tiene la implementación de la fase de mejora del algoritmo y determinar la mejoría que ofrece esta fase, este aspecto es importante de evaluar para poder determinar si el esfuerzo computacional que implica realizar la fase de mejora provee una mejoría en las soluciones.

4. *Factibilidad de las soluciones*. Se busca determinar la cantidad de soluciones infactibles que genera el algoritmo considerando si tiene algún impacto la cantidad de nodos, vehículos y productos. Este aspecto sirve para determinar si el algoritmo proporciona soluciones de buena calidad.

Para cada uno de los aspectos antes mencionados se implementaron contadores en el algoritmo cuyo propósito fue el de proporcionar estas medidas de desempeño.

5.1.6 Diseño de experimentos

Para la realización de la experimentación se realizaron básicamente las siguientes etapas:

- *Etapa 1*: Experimentación para establecer el valor adecuado de los parámetros utilizados en el algoritmo.

- *Etapa 2*: Experimentación para evaluar la calidad de las soluciones.

- *Etapa 3*: Experimentación para evaluar el comportamiento del algoritmo bajo diversas combinaciones de los factores del problema de carga y descarga de productos.
- *Etapa 4*: Resolución del caso de estudio de la industria.

5.2 Análisis de resultados

5.2.1 Etapa 1: Calibración de parámetros de la heurística

Debido a que el algoritmo propuesto tiene parámetros que son necesarios ajustar, se realizó una experimentación para determinar los valores de estos parámetros de tal forma que se obtuvieran mejores soluciones.

Ajuste de parámetro α

Para la calibración del parámetro α se propone realizar un diseño factorial con dos factores. Para las instancias chicas se llevó a cabo la experimentación con los siguientes factores: el primer factor (el parámetro α) se fijó su dominio en rango de valores de $\alpha = 0.00$ hasta $\alpha = 0.80$ en intervalos de 0.01. Mientras que para las instancias medianas se fijó su dominio en un rango de $\alpha = 0.00$ hasta $\alpha = 0.50$ en intervalos de 0.01. En cuanto al segundo factor (el parámetro β) se fijó su dominio en el rango de $\beta = 0.00$ hasta $\beta = 1.00$ en intervalos de 0.25 para las instancias chicas y medianas, dando un total de 13,200 ejecuciones. Por otro lado, para la evaluación de las instancias grandes el dominio del primer factor se fijó en el rango de valores de $\alpha = 0.00$ hasta $\alpha = 0.20$ en intervalos de 0.01. En cuanto al segundo factor se fijó su dominio en el rango de $\beta = 0.00$ hasta $\beta = 1.00$ en intervalos de 0.25 para las instancias grandes, dando un total de 2,100 ejecuciones. Para esta experimentación se fijó el parámetro $maxIterations$ en 1000 iteraciones.

En esta evaluación se analizaron tres aspectos importantes para determinar el

valor adecuado que ofrezca las mejores soluciones. Para el desarrollo de los análisis se utiliza la siguiente nomenclatura:

Z	Conjunto de instancias.
A	Conjunto de parametros α analizados.
$\text{obj}(\alpha, z)$	Mejor solución encontranda en la instancia z utilizando el parametro α.
X_α	Valor reportado en el análisis para el parámetro α.

Análisis 1. Se propone contabilizar las veces en que el mejor valor de la función objetivo fue encontrado bajo el parámetro α

$$X_\alpha = \sum_{z \in Z} f_1(\alpha, z) \qquad f_1(\alpha, z) = \left\{ \begin{array}{ll} 1 & \alpha = \arg\min_{a \in \text{A}} \{\text{obj}(a, z)\} \\ 0 & \text{otro caso} \end{array} \right\}$$

Con este análisis se espera detectar que valor de α ofrece las mejores soluciones. Para ello se busca detectar que valor de α es el que arrojó las mejores soluciones utilizando el valor de α que haya tenido mayor número de mejores soluciones.

En la Figura 5.1 se muestran los resultados del análisis donde se indica la cantidad de veces en que la solución fue obtenida en el parámetro α con los diferentes valores de β. Como se puede ver, para las instancias chicas los mejores valores fueron obtenidos con un valor cercano a $\alpha = 0.70$ mientras que el mejor valor para las instancias medianas y grandes se ubican en un valor cercano a $\alpha = 0.22$ y $\alpha = 0.03$ respectivamente. Esta diferencia posiblemente se deba al tamaño de la lista de candidatos ya que, para un mismo valor de α existen un menor número de movimientos posibles en las instancias chicas que en las instancias grandes. Por otro lado, se puede observar que el parámetro β no impacta significativamente en el comportamiento del algoritmo y las diferencias obtenidas posiblemente pueden deberse a la aleatoriedad del método de solución.

Con base a los resultados obtenidos se puede concluir que el valor de α está en función al tamaño de la instancia y se sugiere dar valores pequeños para instancias

grandes y extender este valor para instancias pequeñas.

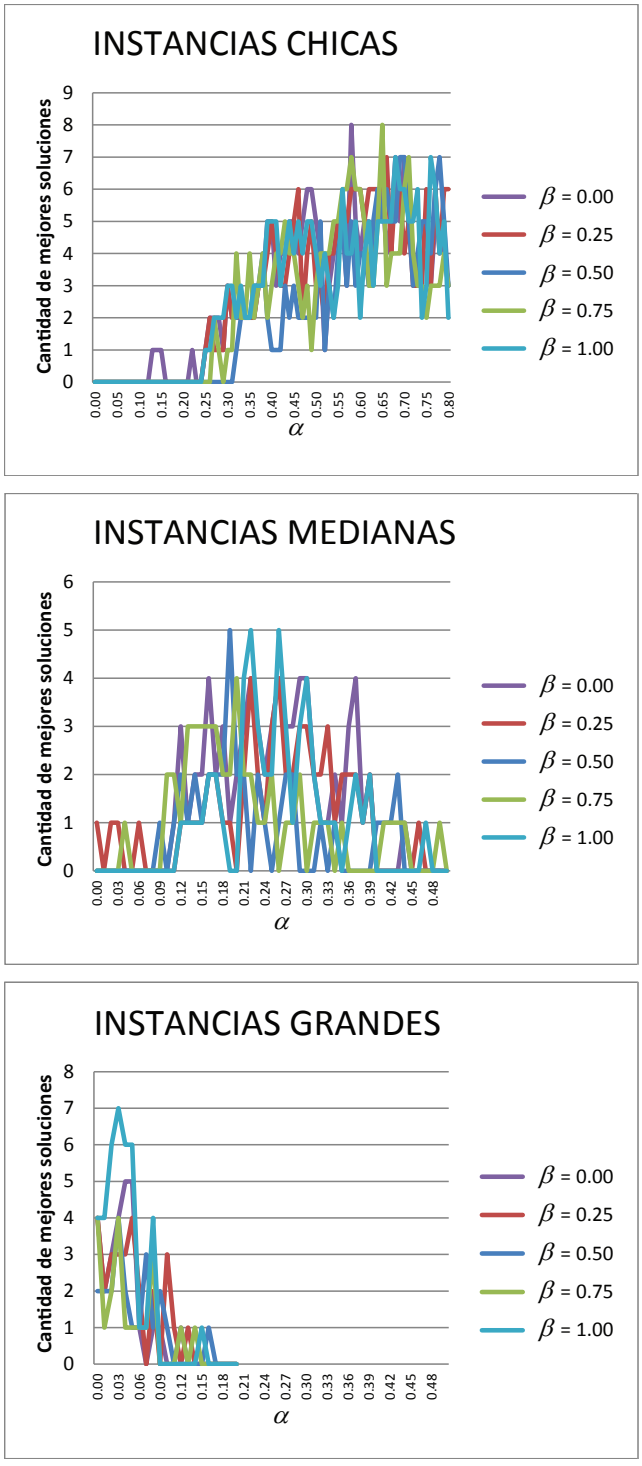


Figura 5.1: Desempeño del algoritmo en cuanto al número de éxitos como función de α y β.

Análisis 2. Se propone determinar que tan alejado está la solución con respecto al mejor valor obtenido.

$$X_\alpha = \frac{1}{|Z|} \sum_{z \in Z} f_2(\alpha, z) \qquad f_2(\alpha, z) = \frac{\text{obj}(\alpha,z) - \min\limits_{a \in A}\{\text{obj}(a,z)\}}{\min\limits_{a \in A}\{\text{obj}(a,z)\}}$$

Con este análisis se espera observar el impacto que tienen los diferentes valores de α en el algoritmo. En este análisis se busca determinar intervalo de desviación relativa (IDR) con respecto a la mejor solución encontrada. Para este análisis el mejor valor para el parámetro α es aquel que en promedio fue más cercano a 0 lo cual significa que en promedio los valores reportados fueron lo más cercano al mejor valor encontrado.

Como se puede observar en la Figura 5.2, el comportamiento del algoritmo, en el caso de las instancias chicas, es mejor cuando se exploran las soluciones con el parámetro α ajustado a un valor cercano al 0.70. Para las instancias medianas y grandes se obtienen mejores soluciones cuando el valor de α se ubica en un valor cercano a $\alpha = 0.20$ y $\alpha = 0.03$, respectivamente.

Estos resultados son muy similares a los obtenidos en el Análisis 1, que nos indican que el parámetro α está en función al tamaño de la instancia, de modo que para una instancia pequeña se obtienen mejores resultados si el valor del parámetro α toma valores cercanos al 0.70 y si se trata de una instancia grande se obtienen mejores resultados si el valor del parámetro α toma valores cercanos al 0.03.

Con base a los resultados obtenidos se puede concluir que el valor de α está en función al tamaño de la instancia y se sugiere dar valores pequeños para instancias grandes y extender este valor para instancias pequeñas.

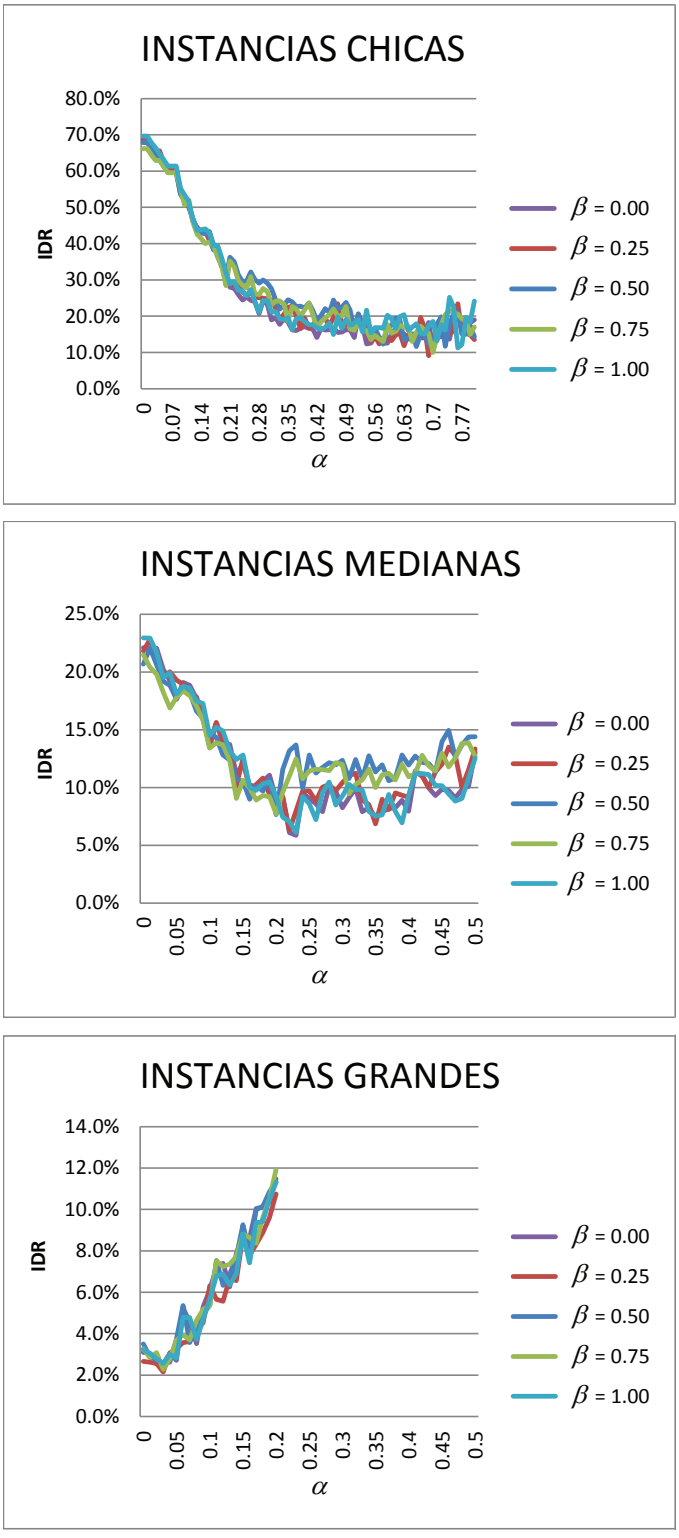


Figura 5.2: Desempeño del algoritmo en cuanto a calidad de solución como función de α y β.

Análisis 3. Se propone determinar el porcentaje de mejora de la solución con respecto a la heurística puramente voraz ($\alpha = 0$).

$$X_\alpha = \frac{1}{|Z|} \sum_{z \in Z} f_3(\alpha, z) \qquad f_3(\alpha, z) = \frac{\text{obj}_{(0,z)} - \text{obj}_{(\alpha,z)}}{\text{obj}_{(0,z)}}$$

Con este análisis se busca determinar el porcentaje de mejora que se tiene al variar el parámetro α con respecto a la solución obtenida sin que haya aleatoriedad en la lista de candidatos, es decir que en la fase constructiva se tenga un procedimiento voraz y se tome siempre el mejor movimiento ($\alpha = 0$). El criterio para determinar cual es mejor valor para el parámetro α es el de detectar cual parámetro ofrece en promedio una mejor solución con respecto a la solución antes señalada.

Un comportamiento esperado para estos resultados es que a medida que se aumente el parámetro α se obtengan mejores soluciones por permitir diversificación en el proceso de construcción. Sin embargo, si se extiende mucho este valor se corre el riesgo de hacer inserciones de muy poca calidad y reportar resultados pobres en la solución.

Como se puede observar en la Figura 5.3, las mejores soluciones se ubican cuando el parámetro α de ajusta en un valor cercano a $\alpha = 0.70$, $\alpha = 0.20$ y $\alpha = 0.03$ para las instancias chicas, medianas y grandes, respectivamente. Esto es consistente con los análisis anteriores

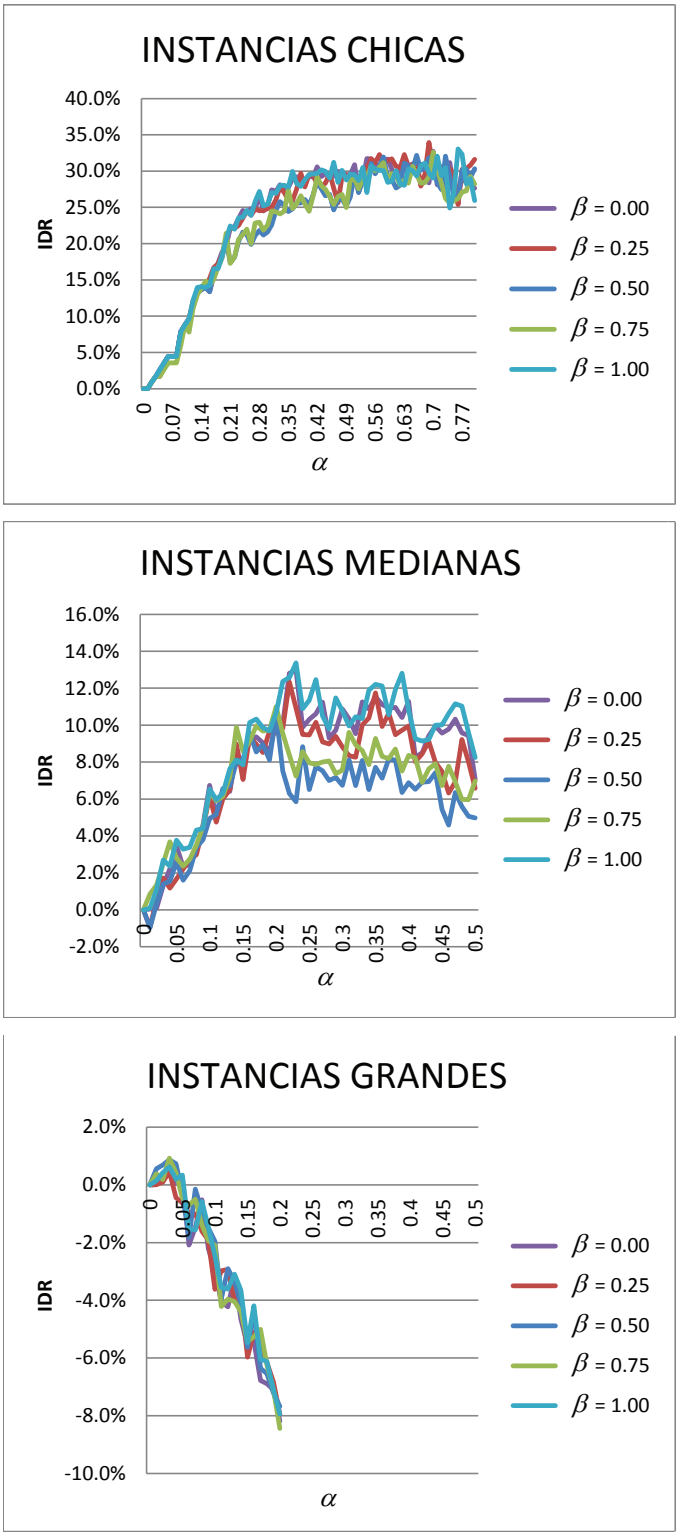


Figura 5.3: Desempeño del algoritmo en cuanto a desviación de la solución enteramente miope ($\alpha = 0$) como función de α y β.

Ajuste de parámetro $\Delta\beta$

Como se pudo observar en la experimentación del ajuste del parámetro α, no se observó impacto en los resultados al variar el parámetro β, de tal modo que, para la calibración del parámetro $\Delta\beta$, se propone realizar un diseño experimental con un solo factor (el parámetro $\Delta\beta$) donde se fijó su dominio en el rango de valores $\Delta\beta = 1/x$ donde $x \in \{1, 2, \ldots, 10\}$. Para esta experimentación se fijó el parámetro $maxIterations$ en 1000 iteraciones.

En esta evaluación se analizó un aspecto importante para determinar el valor adecuado que ofrezca las mejores soluciones. Para el desarrollo de este análisis se utiliza la siguiente nomenclatura:

Z	Conjunto de instancias.
$\text{obj}(\Delta\beta, z)$	Mejor solución encontrada en la instancia z utilizando el valor $\Delta\beta$.
$X_{\Delta\beta}$	Valor reportado en el análisis para el parámetro $\Delta\beta$.
B	Conjunto de los valores $\Delta\beta$ analizados.

Análisis 4. Se propone determinar en que valor de $\Delta\beta$ se encontraron las mejores soluciones.

$$X_{\Delta\beta} = \frac{1}{|Z|} \sum_{z \in Z} f(\Delta\beta, z) \qquad f(\Delta\beta, z) = \frac{\text{obj}(\Delta\beta, z) - \min_{b \in \text{B}}\{\text{obj}(b, z)\}}{\min_{b \in \text{B}}\{\text{obj}(b, z)\}}$$

En este análisis se espera detectar cuál es el mejor valor del parametro $\Delta\beta$ con el fin de determinar hasta que punto ya no es significativo seguir explorando diferentes valores de β y por consiguiente que no se ejecuten operaciones innecesarias. En este experimento se busca determinar en que valor de $\Delta\beta$ se ubican las mejores soluciones con el menor tiempo computacional posible.

Los resultados se muestran en la Figura 5.4, donde se puede ver que las mejores soluciones fueron obtenidas cuando se discretiza el parámetro β con un $\Delta\beta$ de $\frac{1}{8}$. Con base al algoritmo propuesto uno esperaría que entre más chico sea este parámetro

mejores soluciones se obtengan, debido que se realizan un mayor número de ejecuciones. Sin embargo, en el caso de las instancias grandes es posible que no se requiera hacer una discretización muy fina ya que no es un parámetro que impacte en la obtención de buenas soluciones, posiblemente debido a las dimensiones de las instancias.

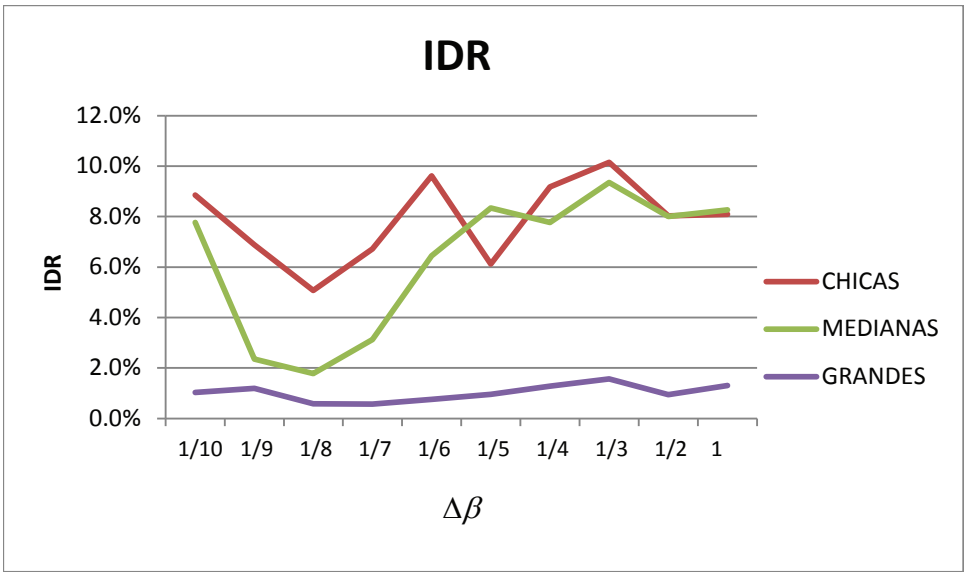


Figura 5.4: Desempeño del algoritmo en cuanto a calidad de solución como función de $\Delta\beta$.

Ajuste de parámetro *maxIterations*

Para la calibración del parámetro *maxIterations* se fijó el valor del parámetro $\alpha = 0.70$, $\alpha = 0.20$ y $\alpha = 0.03$ para las instancias chicas, medianas y grandes, respectivamente. Se fijó además $\Delta\beta = \frac{1}{8}$. El valor de *maxIterations* se fijó en 13,500 iteraciones. Se ejecutaron las instancias de la base de datos y se registró para cada instancia en cuál iteración alcanzó su mejor solución.

Análisis 5. En este aspecto se analizó en que rango de las iteraciones fue donde se presentó en promedio el mayor número de veces el mejor valor en la función objetivo.

Se puede observar en la Figura 5.5 que la mayoría de las mejores soluciones fueron obtenidas en las primeras iteraciones de la ejecución. De manera más minuciosa, el 95 % de las mejores soluciones para las instancias chicas, medianas y grandes fueron obtenidas al ejecutar menos de 4,500, 6,000 y 7,250 iteraciones, respectivamente. Este comportamiento se atribuye a que al tener una instancia de mayor tamaño existe un mayor número de combinaciones posibles, por lo que seguramente se requerirán un mayor número de iteraciones para encontrar una buena solución.

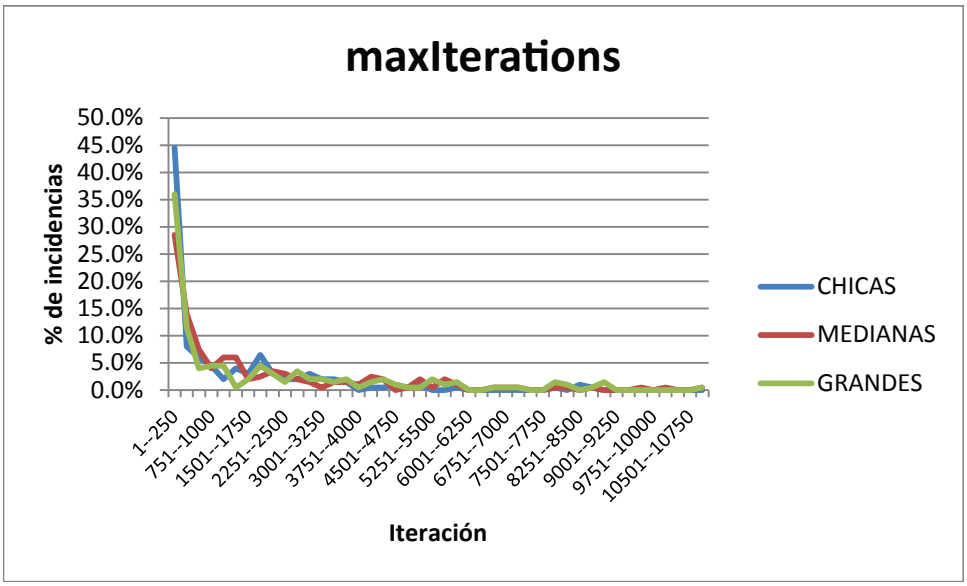


Figura 5.5: Porcentaje de veces donde se obtuvo el mejor valor de la función objetivo.

Análisis 6. Este análisis tiene como finalidad el detectar el beneficio relativo que se obtiene al ejecutar cierto número de iteraciones, ya que se parte de la premisa de que, conforme se incrementa el número de iteraciones a ejecutar se obtiene un mejor resultado, sin embargo es importante detectar hasta que punto este beneficio es verdaderamente significativo.

En este análisis se resolvió el banco de instancias que fue descrito en la sección 5.1.4 y se obtuvo el mejor valor de la función objetivo para cada instancia y este valor fue muestreado en intervalos de 500 iteraciones. Para el desarrollo de este análisis se utiliza la siguiente nomenclatura:

Z Conjunto de instancias.

$\text{obj}(i, z)$ Mejor solución encontranda en la instancia z en la i-ésima iteración.

X_i Valor reportado en el análisis para la i-ésima iteración.

$$X_i = \frac{1}{|Z|} \sum_{z \in Z} f(i, z) \qquad f(i, z) = \frac{\text{obj}_{(i,z)} - \text{obj}_{(i_{\text{máx}}, z)}}{\text{obj}_{(i_{\text{máx}}, z)}}$$

En los resultados obtenidos en esta experimentación, presentados en la Figura 5.6, se puede observar cómo a medida que se ejecuta un mayor número de iteraciones, naturalmente se obtienen mejores soluciones. Sin embargo, es posible analizar, por ejemplo, que en el caso de las instancias chicas, a partir de ejecutar aproximadamente 2500 iteraciones, el porcentaje de mejora ya no incrementa tan rápidamente. Otro punto importante que se podría resaltar es en el caso de las instancias medianas donde se ve un mejoramiento en la solución con un comportamiento progresivo hasta llegar a la ejecución número 7,500 donde en promedio ya no fue posible mejorar significativamente el valor de la función objetivo. Algo interesante que se pudo observar es el hecho que conforme se aumenta el tamaño de las instancias, en las primeras ejecuciones se obtienen soluciones cuyo IDR es menor. Por ejemplo en el caso de las instancias grandes, en las primeras ejecuciones se encuentran soluciones con un IDR inferior del 1 % es decir, que al término de todas las ejecuciones, no fue posible mejorar la solución más que un 1 %. Esto hace ver que si bien se mejoran las soluciones conforme se realizan más iteraciones, desde un principio en el caso de las instancias grandes ya se obtienen soluciones de muy buena calidad.

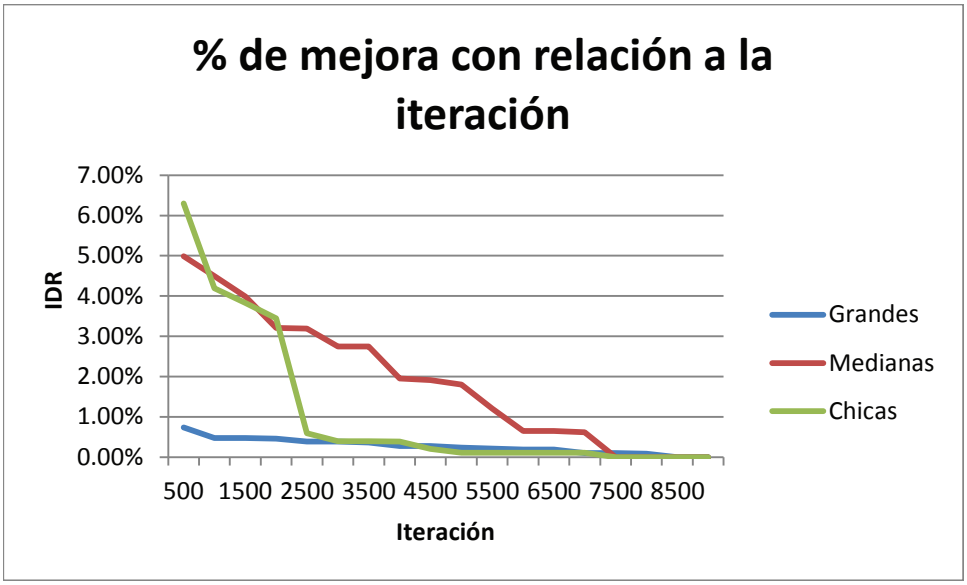


Figura 5.6: Valoración del intervalo de desviación relativa como función del número de iteraciones.

5.2.2 Etapa 2: Verificar la calidad de las soluciones

Debido a la complejidad del problema y al hecho de ser un problema que no ha sido abordado en la literatura, el tener métricas cuantificables para determinar la calidad de las soluciones que arroja el algoritmo es algo que se ve difícil; sin embargo, es posible comparar los resultados obtenidos con instancias admisibles a los modelos matemáticos propuestos, que si bien no es posible alcanzar a apreciar las bondades de la heurística al tratarse de instancias muy reducidas, será posible destacar algunas cualidades de ésta. Otra manera en la que es posible verificar la calidad de las soluciones es realizar una comparación con los resultados reportados actualmente por la empresa.

Por consiguiente en esta etapa se buscará verificar la calidad de las soluciones obtenidas por el algoritmo, realizando una comparación con la resolución de los modelos matemáticos y una comparación con los resultados obtenidos por la empresa.

Comparación con métodos exactos

Para verificar la calidad de las soluciones reportadas por la heurística, se propone comparar las soluciónes con los resultados obtenidos resolviendo de manera exacta los Modelos A y C.

Para ello se resuelven diez instancias con los Modelos A y C por el método de Ramificación y Acotamiento implementado por CPLEX y se comparan con los resultados obtenidos por la heurística. Lo que se espera obtener con este análisis es poder medir el Índice de Optimalidad Relativa (IOR) con respecto a la mejor solución encontrada, lo cual a su vez es un indicador para determinar si las soluciones obtenidas son de buena calidad o no. Cabe aclarar que en este análisis es posible determinar el IOR ya que la comparación de cada instancia se lleva a cabo contra la solución óptima. Para este análisis se resolvieron diez instancias pequeñas cada una con 4 tráileres, 5 nodos y 2 tipos de productos bajo las mismas condiciones.

En la Tabla 5.1 se muestran los resultados obtenidos de la experimentación descrita anteriormente. Para la comparación de los resultados se consideraron dos aspectos. El primer aspecto que se consideró fue el valor objetivo obtenido por ambos modelos que representa el costo de ruteo y el costo fijo por el uso de vehículos. Por otro lado el segundo aspecto que se consideró fue el tiempo de ejecución (medido en segundos).

Como se puede observar en la Tabla 5.1, las soluciones encontradas con el Modelo C son de igual o mejor calidad que las del Modelo A, lo cual era de esperarse ya que, como se presentó anteriormente, el Modelo C es más flexible permitiendo formar rutas de tipo camino cerrado. No obstante, el tiempo necesario para resolver el Modelo C es mayor. De hecho, teniendo un tiempo máximo de 4 horas, no fue posible asegurar el óptimo global en algunas instancias. En cuanto a la heurística propuesta, se puede observar que a la heurística le fue posible ofrecer soluciones competitivas en un tiempo de computo razonable.

	Modelo A			Modelo C			Heurística		
id	IOR(%)	objetivo	tiempo	IOR(%)	objetivo	tiempo	IDR(%)*	objetivo	tiempo
1	0.00	3120	7.22	0.00	3120	96.33	33.21	4156	<1
2	0.00	8042	20.88	74.88	5972	14401.13	17.88	7040	<1
3	0.00	16850	42.74	90.58	15932	14400.64	34.55	21436	<1
4	0.00	9308	20.43	76.65	6424	14401.07	5.95	6806	<1
5	0.00	6794	44.13	77.92	6794	14400.67	16.57	7920	<1
6	0.00	3667	6.43	0.00	3474	323.07	0.00	3474	<1
7	0.00	3667	6.34	0.00	3474	444.62	0.00	3474	<1
8	0.00	9632	63.51	80.41	7656	600.81	0.00	7656	<1
9	0.00	2644	9.44	0.00	2644	111.66	0.00	2644	<1
10	0.00	3456	14.31	0.00	3456	7249.72	36.00	4700	<1
* Con respecto al valor encontrado por el Modelo C.									

Tabla 5.1: Comparación de la heurística y los Modelos A y C en instancias pequeñas.

En la Figura 5.7 se puede observar una comparación de los tiempos de ejecución de los distintos procedimientos evaluados en este análisis y en la Figura 5.8 la desviación relativa con respecto al óptimo. Si bien las mejores soluciones son obtenidas con el Modelo C, el tiempo de ejecución necesario para poder resolver estas instancias hacen que ésta no sea una opción viable para resolver instancias de tamaño real.

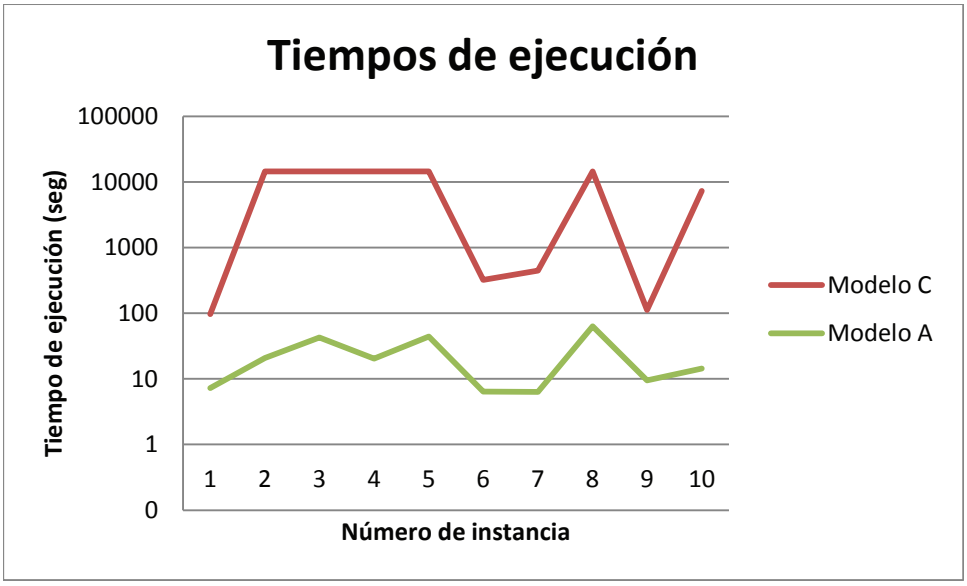


Figura 5.7: Tiempos de ejecución para resolver los Modelos A y C.

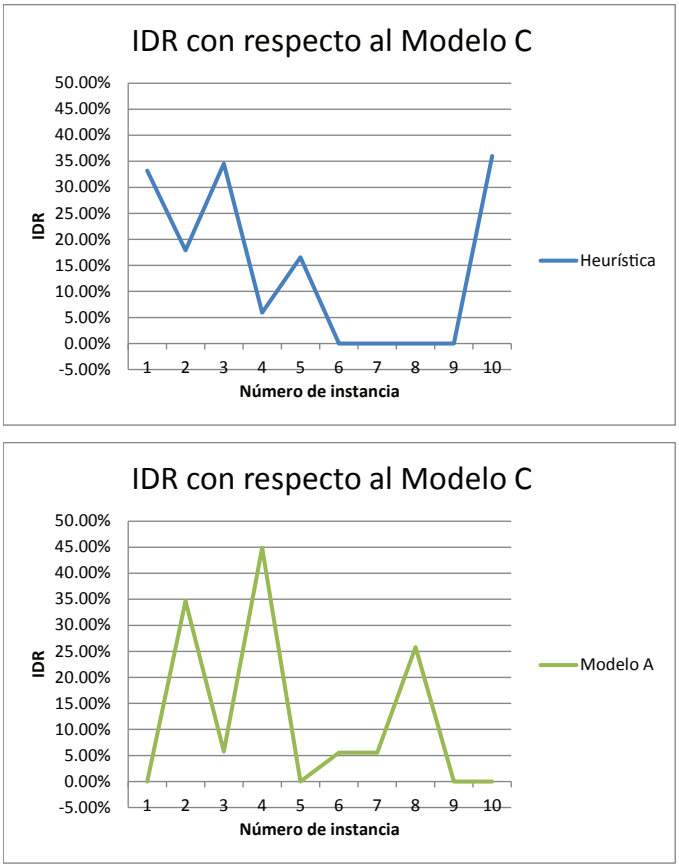


Figura 5.8: IDR del Modelo A y la Heurística con respecto al Modelo C.

Resultados de las instancias generadas

En las evaluaciones expuestas en la Sección 5.1.4 se generaron un banco de instancias, las cuales fueron utilizadas para poder evaluar el desempeño del algoritmo. A continuación en la Tabla 5.2 se presentan los resultados obtenidos con la heurística propuesta para estas instancias, divididas según el tamaño de las instancias (chicas, medianas y grandes), donde se muestra el identificador de la instancia (ID), el valor obtenido en la función objetivo (FO) representando el costo de ruteo y el costo fijo del uso de los vehículos, la cantidad de vehículos dobles (VD), la cantidad de vehículos sencillos (VS), el tiempo de ejecución expresado en segundos (Tiempo) y finalmente el costo del ruteo (Costo).

Chicas						Medianas						Grandes					
ID	FO	VD	VS	Tiempo	Costo	ID	FO	VD	VS	Tiempo	Costo	ID	FO	VD	VS	Tiempo	Costo
C001	8305	0 (0 %)	3 (100 %)	2.61	3805	M001	12355	1 (20 %)	4 (80 %)	43.052	4855	G001	69907	11 (41 %)	16 (59 %)	2626.939	29407
C002	5119	0 (0 %)	2 (100 %)	8.952	2119	M002	12447	0 (0 %)	5 (100 %)	18.851	4947	G002	68128	19 (70 %)	8 (30 %)	818.983	27628
C003	7035	0 (0 %)	3 (100 %)	2.158	2535	M003	15212	2 (33 %)	4 (67 %)	48.969	6212	G003	69941	18 (64 %)	10 (36 %)	4423.143	27941
C004	6638	0 (0 %)	3 (100 %)	2.483	2138	M004	11626	0 (0 %)	5 (100 %)	38.667	4126	G004	76647	16 (53 %)	14 (47 %)	4565.851	31647
C005	6407	0 (0 %)	3 (100 %)	6.109	1907	M005	12094	0 (0 %)	5 (100 %)	86.001	4594	G005	75340	15 (52 %)	14 (48 %)	4187.319	31840
C006	4748	0 (0 %)	2 (100 %)	9.059	1748	M006	12716	0 (0 %)	5 (100 %)	84.004	5216	G006	73921	19 (68 %)	9 (32 %)	4377.654	31921
C007	7951	0 (0 %)	3 (100 %)	4.753	3451	M007	14000	0 (0 %)	6 (100 %)	56.846	5000	G007	72118	12 (43 %)	16 (57 %)	2884.006	30118
C008	7334	0 (0 %)	3 (100 %)	2.232	2834	M008	13740	0 (0 %)	6 (100 %)	56.878	4740	G008	100899	7 (18 %)	33 (83 %)	8588.573	40899
C009	6667	0 (0 %)	3 (100 %)	1.998	2167	M009	14133	0 (0 %)	6 (100 %)	48.477	5133	G009	36315	9 (60 %)	6 (40 %)	1228.22	13815
C010	5341	0 (0 %)	2 (100 %)	10.6	2341	M010	13913	0 (0 %)	6 (100 %)	25.738	4913	G010	41895	12 (75 %)	4 (25 %)	1488.127	17895
C011	6523	0 (0 %)	3 (100 %)	11.123	2023	M011	14203	0 (0 %)	6 (100 %)	44.003	5203	G011	35248	10 (71 %)	4 (29 %)	1703.412	14248
C012	5155	0 (0 %)	2 (100 %)	6.095	2155	M012	12107	0 (0 %)	5 (100 %)	47.365	4607	G012	44300	8 (47 %)	9 (53 %)	598.368	18800
C013	6694	0 (0 %)	3 (100 %)	10.796	2194	M013	11914	0 (0 %)	5 (100 %)	41.966	4414	G013	34149	12 (86 %)	2 (14 %)	427.298	13149
C014	5323	0 (0 %)	2 (100 %)	10.935	2323	M014	10138	0 (0 %)	4 (100 %)	42.119	4138	G014	36573	10 (71 %)	4 (29 %)	443.501	15573
C015	7540	0 (0 %)	3 (100 %)	2.161	3040	M015	14037	0 (0 %)	6 (100 %)	47.397	5037	G015	41324	10 (63 %)	6 (38 %)	461.606	17324
C016	6549	0 (0 %)	3 (100 %)	6.255	2049	M016	12072	0 (0 %)	5 (100 %)	48.385	4572	G016	32309	11 (85 %)	2 (15 %)	416.283	12809
C017	7330	0 (0 %)	3 (100 %)	2.232	2830	M017	12387	1 (20 %)	4 (80 %)	94.611	4887	G017	42232	7 (44 %)	9 (56 %)	10342.01	18232
C018	7104	0 (0 %)	3 (100 %)	3.142	2604	M018	14699	0 (0 %)	6 (100 %)	49.648	5699	G018	32098	13 (100 %)	0 (0 %)	422.671	12598
C019	5448	0 (0 %)	2 (100 %)	2.032	2448	M019	12595	1 (20 %)	4 (80 %)	47.068	5095	G019	41564	12 (75 %)	4 (25 %)	457.315	17564
C020	6584	0 (0 %)	3 (100 %)	1.969	2084	M020	12257	0 (0 %)	5 (100 %)	15.545	4757	G020	43314	11 (65 %)	6 (35 %)	482.744	17814

Tabla 5.2: Resultados del banco de instancias.

5.2.3 Etapa 3: Comportamiento del algoritmo

En esta etapa se propone realizar un estudio para conocer el comportamiento del algoritmo, con el fin de poder detectar puntos fuertes y débiles del algoritmo. Debido a la gran cantidad de información que se obtiene en esta etapa fue necesario el uso de graficas y tablas para lograr explicar con mayor facilidad los resultados. Se fijó el valor del parámetro $\alpha = 0.70$, $\alpha = 0.20$ y $\alpha = 0.03$ para las instancias chicas, medianas y grandes, respectivamente. Para el parámetro $\Delta\beta$ se fijó su valor en $\Delta\beta = \frac{1}{8}$. Se fijó el parámetro $maxIterations$ en 2,500. A continuación se exponen los resultados para cada uno de los aspectos evaluados en el algoritmo.

Infactibilidad

Este experimento busca analizar el grado de factibilidad de las soluciones obtenidas por el algoritmo, para conocer si el proceso de construcción y mejora del algoritmo forman una heurística que obtenga buenos resultados, de tal modo que en el análisis desarrollado se muestra el porcentaje de éxito (en cuanto a encontrar soluciones factibles) por iteración y por ejecución global del algoritmo. Es decir, se muestra el porcentaje de iteraciones en que se pudo encontrar una solución factible (al final de cada iteración) y también se muestra el porcentaje de soluciones que al final de la ejecución del algoritmo fueron factibles.

La heurística propuesta construye y mejora la solución cierto número de iteraciones ($maxIterations$). Al ejecutar el algoritmo, es posible que en una iteración dada la solución obtenida sea infactible; sin embargo, el algoritmo al final de su ejecución reporta la mejor solución encontrada tomando en cuenta la factibilidad y el valor objetivo. Por ello es importante analizar la cantidad de soluciones infactibles que genera el algoritmo al cabo de todas las iteraciones ejecutadas.

En la Tabla 5.3 se muestra el porcentaje de iteraciones que fueron factibles para la experimentación de las instancias chicas, medianas y grandes. Se consideraron tres aspectos importantes analizar. El mejor porcentaje de iteraciones factibles (Máximo),

el promedio del porcentaje (Promedio) y el peor porcentaje (Mínimo). Es posible visualizar como a medida que se aumenta el tamaño de las instancias al algoritmo le cuesta más trabajo encontrar soluciones factibles. Por ejemplo, en la mayoría de las iteraciones se obtuvieron soluciones factibles para las instancias chicas (99.6 %). Sin embargo al ejecutar instancias de mayor tamaño este valor tiende a descender.

	Chicas	Medianas	Grandes
Máximo	100.0 %	100.0 %	100.0 %
Promedio	99.6 %	68.0 %	60.0 %
Mínimo	79.0 %	20.0 %	0.0 %

Tabla 5.3: Porcentaje de iteraciones factibles.

A pesar de ello, como se mencionó anteriormente, el algoritmo reporta al final de su ejecución la mejor solución obtenida. Con ello, en la Tabla 5.4 se muestra el porcentaje de soluciones factibles encontradas por el algoritmo para las instancias chicas, medianas y grandes. En ella se muestra que fue posible encontrar soluciones factibles para todas las instancias chicas y medianas, y el 95 % de las veces fue posible para las instancias grandes.

Chicas	Medianas	Grandes
100.0 %	100.0 %	95.0 %

Tabla 5.4: Porcentaje de soluciones factibles.

Tiempos de ejecución

Un aspecto importante, además de la calidad, es el tiempo de ejecución del algoritmo. Es importante que este tiempo sea razonable para garantizar a la empresa la obtención de soluciones en el momento que se requiera. Uno de los aspectos que se buscan analizar es el tiempo promedio que le toma ejecutar las iteraciones para las

instancias de diferentes tamaños. Además de analizar el mayor y el menor tiempo en el cual fue posible resolver una instancia.

En la Tabla 5.5 se muestran los tiempos de ejecución de las distintas instancias analizadas. Como se puede ver los tiempos de ejecución son razonables y en el caso de las instancias grandes que son las instancias con las dimensiones de un problema real, los tiempos de ejecución no llegan más allá de unos cuantos minutos.

	Chicas	Medianas	Grandes
Máximo	1.3	12.5	1075.8
Promedio	1.0	8.1	348.6
Mínimo	0.8	5.0	149.5

Tabla 5.5: Tiempo promedio (en segundos) al ejecutar mil iteraciones.

En este tiempo se resuelve el problema de asignación propuesto, se realiza la fase constructiva y la fase de búsqueda local, por lo que un aspecto importante a analizar es el tiempo promedio que toma en realizar cada uno de los componentes que integran al algoritmo.

En la Tabla 5.6 se muestra el tiempo promedio de ejecución que toma resolver el problema de asignación (AP), la fase constructiva (FC), la búsqueda local (BL) y otros procesos adicionales que fueron implementados (Otros). Este último incluye el tiempo de lectura para cargar la instancia, el tiempo de escritura para guardar las soluciones, además del tiempo de ejecución de un proceso adicional para validar las soluciones obtenidas, así como tiempo de ejecución empleado para reportar el progreso del algoritmo en tiempo real.

	Chicas	Medianas	Grandes
AP	0.28 (28.30 %)	1.84 (22.72 %)	12.80 (03.67 %)
FC	0.15 (14.73 %)	1.24 (15.28 %)	52.15 (14.96 %)
BL	0.16 (16.08 %)	1.27 (15.61 %)	55.15 (15.81 %)
Otros	0.41 (40.88 %)	3.75 (46.36 %)	228.5 (65.55 %)

Tabla 5.6: Porcentajes del tiempo de ejecución de los diferentes componentes.

En general el tiempo de ejecución de la fase constructiva crece conforme aumenta el tamaño de la instancia y representa cerca del 15 % del tiempo total de ejecución. Mientras que el tiempo de ejecución para realizar la fase de búsqueda local representa cerca del 16 %.

Mejoría en búsqueda local

Este aspecto busca determinar el beneficio de ejecutar la búsqueda local con relación a la función objetivo. Este aspecto es importante de analizar para determinar si existe un beneficio real al implementarlos. Para el desarrollo del siguiente análisis se hace uso de la siguiente nomenclatura:

X_{FC} Valor de la función objetivo después de la Fase Constructiva.

X_{BL_1} Valor de la función objetivo después de ejecutar el vecindario Convertir Vehículos a Dobles.

X_{BL_2} Valor de la función objetivo después de ejecutar el vecindario Transferir Entregas.

Para analizar el beneficio de implementar el vecindario Convertir Vehículos a Dobles que fue descrito en la Sección 4.4.1, se expresará este beneficio como el cambio relativo en la función objetivo con respecto a la fase constructiva definido como:

$$\frac{X_{\text{FC}} - X_{\text{BL}_1}}{X_{\text{FC}}}$$

En la Tabla 5.7 se muestra el beneficio promedio de implementar el vecindario de Convertir Vehículos a Dobles. Al analizar el beneficio obtenido, se puede visualizar que en las instancias de tamaño real el costo de las soluciones se reduce en promedio en un 10 %, lo cual es significativo porque esta cifra puede representar una fuerte cantidad de dinero. Si bien otro aspecto interesante que se pudiese resaltar es que el beneficio se incrementa conforme se incrementa el tamaño de la instancia, esto posiblemente se debe a que exista una mayor cantidad combinaciones que en las instancias chicas.

	Chicas	Medianas	Grandes
Máximo	1.2 %	4.0 %	17.2 %
Promedio	0.0 %	1.5 %	10.0 %
Mínimo	0.0 %	0.0 %	6.3 %

Tabla 5.7: Porcentaje de mejora al implementar vecindario: Convertir Vehículos a Dobles.

Para analizar el beneficio de implementar el vecindario Transferir Entregas que fue descrito en la Sección 4.4.2, se expresará este beneficio como el cambio relativo en la función objetivo con respecto a la fase constructiva definido como:

$$\frac{X_{\mathrm{BL}_1} - X_{\mathrm{BL}_2}}{X_{\mathrm{FC}}}$$

En la Tabla 5.8 se muestra el beneficio promedio de implementar el vecindario de Transferir Entregas. Al analizar el beneficio obtenido, se puede observar que a medida que se aumenta el tamaño de las instancias también aumenta el beneficio obtenido. Es de esperarse que el porcentaje de mejora de implementar este vecindario sea menor con respecto al vecindario Convertir Vehículos a Dobles, debido a que el vecindario de Transferir Entregas mejora la solución únicamente al disminuir la cantidad vehículos utilizados.

	Chicas	Medianas	Grandes
Máximo	4.5 %	4.9 %	4.4 %
Promedio	1.8 %	2.8 %	2.2 %
Mínimo	0.0 %	0.5 %	0.1 %

Tabla 5.8: Porcentaje de mejora al implementar vecindario: Transferir Entregas.

5.2.4 Etapa 4: Comparación con caso de la industria

En colaboración con la empresa que motivó el presente trabajo se realizó un análisis para determinar la manera de cómo se estaba resolviendo el problema expuesto en esta tesis y las características que fueron tomadas en cuenta para su resolución. Con base a la información analizada fue posible detectar ciertas características en la modelación que hacían inequiparables a las soluciones obtenidas por la heurística propuesta; sin embargo, era posible relajar algunas restricciones y así poder tener soluciones para su justa comparación.

De tal manera que para este análisis se desarrolló un experimento de tamaño real, el cual es resuelto con el método de la empresa y con la heurística propuesta relajando la restricción de la capacidad de las plantas, ya que en la solución presentada por la industria era posible exceder esta capacidad siempre y cuando se obtuvieran mejores soluciones.

Con base al análisis hecho anteriormente y buscando tener las mismas condiciones en ambos modelos, se realizó la experimentación para un caso particular, el cual se puede observar en la Tabla 5.9. Como puede verse, la solución obtenida por la heurística presentó un mayor uso de vehículos que la metodología actual. Esta situación se debe a que en la metodología de la empresa el objetivo de la modelación es minimizar el número de vehículos a utilizar a diferencia de la metodología propuesta donde se presenta una modelación cuyo objetivo es minimizar los costos. A pesar de ello se puede observar también que el costo total a utilizar es mucho menor

a la metodología de la empresa la cual es un resultado positivo para la heurística.

	Heurística propuesta	Metodología de la empresa
Núm. Vehículos	50	47
Núm. V. Sencillos	28	9
Núm. V. Dobles	22	38
Total de Tráileres	72	85
Costo Ruteo	$167,020.00	$186,018.00
Costo Vehículos	$75,000.00	$70,500.00
Costo Total	**$242,020.00**	**$256,518.00**

Tabla 5.9: Resultados obtenidos para la resolución del caso de la industria.

Este contraste entre el costo de la solución y el número de vehículos a utilizar puede tener su explicación en la manera de como se utilizan los vehículos en los viajes para entregar los productos. En la Figura 5.9 se muestra un caso ilustrativo que presenta las rutas obtenidas por ambas metodologías, donde claramente es posible apreciar que las rutas generadas por el algoritmo propuesto presentan una menor cantidad de viajes y visitas recurrentes a un mismo nodo.

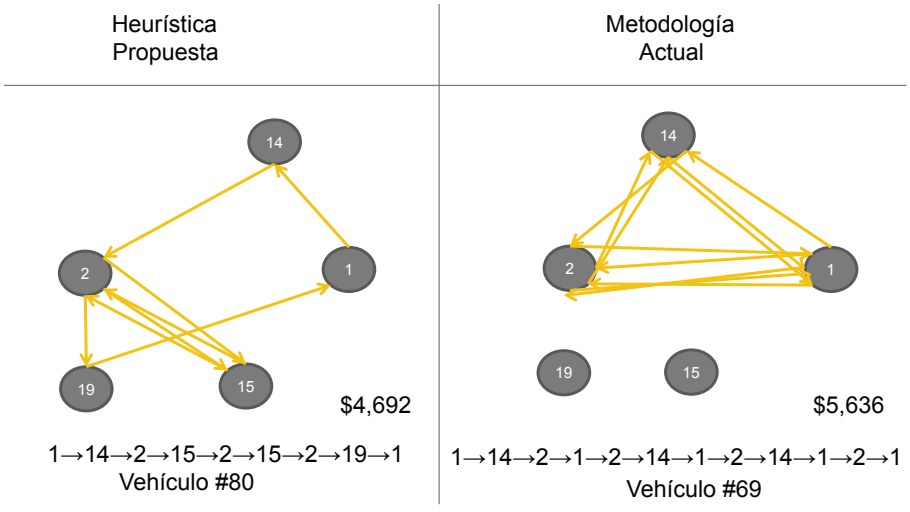


Figura 5.9: Rutas obtenidas por ambas metodologías.

Capítulo 6

Conclusiones

Después de haber realizado un estudio muy detallado sobre el problema de carga y descarga, además de formular diversos modelos matemáticos para representar fielmente al problema, así como codificar la metodología propuesta y realizar su experimentación correspondiente, es oportuno recapacitar sobre los logros obtenidos y las líneas posibles de investigación que surgen de este trabajo. En este último capítulo se expresan las conclusiones a las que se llega después del trabajo expuesto en esta tesis. Además se presentan las contribuciones y el posible trabajo futuro que se puede desarrollar a partir de este trabajo.

6.1 Conclusiones

En el presente trabajo de tesis se ha estudiado un problema de ruteo de vehículos con carga y descarga de productos que nace a partir de un problema real de una empresa embotelladora ubicada en la ciudad de Monterrey, N.L., México. Es necesario resolver este problema para solucionar los requerimientos de logística que tiene la empresa. Para este problema existen múltiples decisiones, y a su vez, cada una de estas decisiones impactan fuertemente a las otras, haciendo que la toma de decisiones sea una situación difícil de resolver.

Para solucionar este problema, se realizaron varias formulaciones matemáticas modelado como un problema lineal entero-mixto. Sin embargo, debido al tamaño y características de las instancias de tamaño real no fue posible resolverlas de forma

exacta en un tiempo razonable, lo que llevó a la implementación de una técnica heurística para encontrar soluciones a este problema.

Los objetivos planteados en la Sección 1.3 han sido completamente satisfechos. Se ha logrado un conocimiento de cada uno de los requerimientos, restricciones y características que la empresa presentó acerca de sus necesidades reales.

A partir de este conocimiento fue desarrollado el modelo matemático con el cual fue posible obtener soluciones para instancias pequeñas del problema mediante el uso del método de Ramificación y Acotamiento implementado en CPLEX. Se desarrolló una metodología tipo GRASP donde la fase de construcción y mejora fueron específicamente diseñadas para satisfacer las restricciones de la empresa. Por otro lado fueron generadas instancias a partir del conocimiento de la empresa por medio de un generador de instancias que fue otro aporte de esta tesis. Estas instancias fueron utilizadas para lograr una completa evaluación de la metodología de solución desarrollada.

El algoritmo desarrollado incorpora todas las decisiones planteadas originalmente y es capaz de proporcionar soluciones que cumplan con las restricciones del problema de carga y descarga y que son soluciones válidas para la empresa ya que toman en cuenta las características específicas del problema real. Dichas soluciones han sido obtenidas en tiempos razonables a pesar de que el tamaño de las instancias utilizadas haya sido grande. Sin embargo, el algoritmo propuesto es un algoritmo ad hoc para un problema de ruteo de vehículos con características muy específicas. Es decir que sólo podría ser utilizado como herramienta para problemas con este tipo de características. Sin embargo el objetivo de esta tesis no era desarrollar una metodología general, sino proporcionar una herramienta eficaz que permitiera trabajar el problema con todas las restricciones y consideraciones impuestas por la empresa.

6.2 Contribuciones

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

- Se logró un entendimiento detallado del problema abordado, planteándolo así como un problema de carga y descarga.
- Se realizaron varios modelos matemáticos que definen completamente al problema estudiado junto con algunas de sus variantes.
- Se demostró la complejidad computacional del problema.
- Se realizaron pruebas sobre el modelo matemático para determinar el tamaño máximo al cual es posible resolver el problema de manera óptima.
- Se diseñó y construyó una heurística basada en GRASP, la cual resuelve el problema de carga y descarga con las restricciones propias del problema estudiado, al incorporar varios procedimientos clave para tratar al problema como un problema de carga y descarga con emparejamiento.
- Se realizó un completo estudio que demostró empíricamente que la metodología de solución propuesta logra obtener soluciones que cumplen con los requisitos establecidos.
- Se logró crear una herramienta que logró obtener mejores resultados para el problema real de la industria, como resultado de aplicar los conocimientos adquiridos a lo largo de los estudios de maestría.
- Se construyó una amplia variedad de instancias para evaluar el desempeño del algoritmo.

6.3 Recomendaciones y trabajo futuro

Se proponen las siguientes líneas de investigación que podrían dar pie a futuras investigaciones a raíz del trabajo de esta tesis.

Es posible que se mejoren las soluciones al implementar diferentes métodos para la generación de pedidos.

Como se pudo observar en la fase de experimentación, la búsqueda local fue de vital importancia para obtener soluciones de buena calidad. Sin embargo los vecindarios propuestos fueron específicamente diseñados como búsquedas simples del tipo descendente y es posible que al experimentar con otros vecindarios expuestos en la literatura o vecindarios derivados de métodos más sofisticados como Búsqueda Tabú, Búsqueda por Entornos Variables, Búsqueda Dispersa, o alguna técnica especializada como Memoria Adaptativa se puedan obtener mejores soluciones.

Unos de los supuestos que se han considerado es la que el tiempo de servicio es un dato fijo que no está en función a la cantidad de producto que se carga o se descarga en un nodo. Al considerar que el tiempo se servicio esté en función de la cantidad de producto que se carga o se descarga en un nodo, se representaría más fielmente a la realidad actual de la industria.

Actualmente la manera cómo se restringe el uso de andenes es la de limitar la llegada de vehículos por hora a un centro de distribución, sin embargo es posible que existan solapamientos entre los vehículos al no considerarse el tiempo en el que un vehículo permanece en un centro de distribución. Por lo que se propone investigar nuevas formas de restringir la llegada de los vehículos a los centros de distribución para que con ello se represente más fielmente la realidad.

Bibliografía

[1] M. Ambrosini, T. Caruso, S. Foresti y G. Righini. A GRASP for the pickup and delivery problem with rear loading. Reporte Técnico 65, Information Technology Department, University of Milan, Crema, Italia, Agosto 2004.

[2] O. Arbelaitz Gallego. *Soluciones basadas en Simulated Annealing para el VRPTW*. Tesis doctoral, Universidad del País Vasco, Donostia, España, Mayo 2002.

[3] C. Archetti, M. G. Speranza y A. Hertz. A tabu search algorithm for the split delivery vehicle routing problem. *Transportation Science*, 40(1):64–73, 2006.

[4] R. Baldacci, M. Battarra y D. Vigo. Valid inequalities for the fleet size and mix vehicle routing problem with fixed costs. *Networks*, 54(4):178–189, 2009.

[5] J.F. Bard, G. Kontoravdis y G. Yu. A branch-and-cut procedure for the vehicle routing problem with time windows. *Transportation Science*, 36(2):250–269, 2002.

[6] R. Bent y P. Van Hentenryck. A two-stage hybrid algorithm for the pickup and delivery vehicle routing problem with time windows. *Computers & Operations Research*, 33(4):875–893, 2006.

[7] G. Berbeglia, J.-F. Cordeau, I. Gribkovskaia y G. Laporte. Static pickup and delivery problems: A classification scheme and survey. *TOP*, 15(1):1–31, 2007.

[8] G. Berbeglia, J.-F. Cordeau y G. Laporte. Dynamic pickup and delivery problems. *European Journal of Operational Research*, 202(1):8–15, 2010.

[9] D. J. Bertsimas. A vehicle routing problem with stochastic demand. *Operations Research*, 40(3):574–585, 1992.

[10] D. J. Bertsimas y G. Van Ryzin. A stochastic and dynamic vehicle routing problem in the euclidean plane. *Operations Research*, 39(4):601–615, 1991.

[11] R. E. Burks. *An Adaptive Tabu Search Heuristic for Location Routing Pickup and Delivery Problem with Time Windows with Theater Distribution Application*. Tesis doctoral, Air Force Institute of Technology, EUA, Febrero 2006.

[12] J. Carlsson, D. Ge, A. Subramaniam, A. Wu y Y. Ye. Solving min-max multi-depot vehicle routing problem. Reporte Técnico 4, Institute of Computational and Mathematical Engineering, Stanford University, Stanford, EUA, Mayo 2007.

[13] J. C. Créput, A. Koukam, J. Kozlak y J. Lukasik. An evolutionary approach to pickup and delivery problem with time windows. En M. Bubak, D. Van Albada, M. A. Sloot y J. Dongarra, editores, *Computational Science - ICCS 2004*, volumen 3038 de *Lecture Notes in Computer Science*, pp. 1102–1108. Springer, Berlín, Alemania, 2004.

[14] B. De Backer, V. Furnon, P. Kilby, P. Prosser y P. Shaw. Local search in constraint programming: Application to the vehicle routing problem. *Transportation Research Record*, 1520:1–15, 1997.

[15] U. Derigs, J. Gottlieb, J. Kalkoff, M. Piesche, F. Rothlauf y U. Vogel. Vehicle routing with compartments: Applications, modelling and heuristics. *OR Spectrum*, 33(4):885–914, 2011.

[16] R. Dondo, C. A. Méndez y J. Cerdá. Optimal management of logistic activities in multi-site environments. *Computers & Chemical Engineering*, 32(11):2547–2569, 2008.

[17] G. Dong, J. Tang, K. K. Lai y Y. Kong. An exact algorithm for vehicle routing and scheduling problem of free pickup and delivery service in flight ticket sales

companies based on set-partitioning model. *Journal of Intelligent Manufacturing*, 22(5):789–799, 2011.

[18] M. Dror, D. Fortin y C. Roucairol. Redistribution of self-service electric cars: A case of pickup and delivery. Reporte Técnico RR-3543, INRIA, Cedex, Francia, Noviembre 1998.

[19] T. A. Feo y M. G. C. Resende. A probabilistic heuristic for a computationally difficult set covering problem. *Operations Research Letters*, 8(2):67–71, 1989.

[20] R. Fisher. Statistical methods and scientific induction. *Journal of the Royal Statistical Society. Series B (Methodological)*, 17(1):69–78, 1955.

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

[22] Y. Gajpal y P. Abad. An ant colony system (ACS) for vehicle routing problem with simultaneous delivery and pickup. *Computers & Operations Research*, 36(12):3215–3223, 2009.

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

[24] M. Gendreau, G. Laporte y R. Séguin. Stochastic vehicle routing. *European Journal of Operational Research*, 88(1):3–12, 1996.

[25] B. L. Golden y E. A. Wasil. Computerized vehicle routing in the soft drink industry. *Operation Research*, 35(1):6–17, 1987.

[26] I. Gribkovskaia, Ø. Halskau Sr y K. N. Bugge Myklebost. Models for pick-up and deliveries from depots with lasso solutions. Working Paper 1, Molde University College, The Norwegain School of Logistics, Molde, Noruega, 2002.

[27] A. Haghani, H. Hu y Q. Tian. An optimization model for real-time emergency vehicle dispatching and routing. *Transportation Research Record: Journal of the Transportation Research Board*, 1882:176–183, 2003.

[28] H. Hernández-Pérez y J. J. Salazar-González. Heuristics for the one-commodity pickup-and-delivery traveling salesman problem. *Transportation Science*, 38(2):245–255, 2004.

[29] H. Hernández-Pérez y J. J. Salazar-González. The multi-commodity one-to-one pickup-and-delivery traveling salesman problem. *European Journal of Operational Research*, 196(3):987–995, 2009.

[30] A. Hoff, I. Gribkovskaia, G. Laporte y A. Løkketangen. Lasso solution strategies for the vehicle routing problem with pickups and deliveries. *European Journal of Operational Research*, 192(3):755–766, 2009.

[31] M. I. Hosny. *Investigating Heuristic and Meta-Heuristic Algorithms for Solving Pickup and Delivery Problems*. Tesis doctoral, Cardiff University, School of Computer Science and Informatics, Cardiff, Inglaterra, Marzo 2010.

[32] M. I. Hosny y C. L. Mumford. The single vehicle pickup and delivery problem with time windows: Intelligent operators for heuristic and metaheuristic algorithms. *Journal of Heuristics*, 16(3):417–439, 2010.

[33] IBM. *IBM ILOG CPLEX V12.1: User's Manual for CPLEX*. IBM Corporation, New York, USA, 2009.

[34] S. Jung y A. Haghani. Genetic algorithm for a pickup and delivery problem with time windows. *Transportation Research Record*, 1733:1–7, 2000.

[35] H. L. M. Kerivin, M. Lacroix y A. R. Mahjoub. Models for the single-vehicle preemptive pickup and delivery problem. *Journal of Combinatorial Optimization*, 23(2):196–223, 2012.

[36] G. F. King y T. M. Mast. Excess travel: Causes, extent and consequences. *Transportation Research Record*, 1111:126–134, 1987.

[37] A. Landrieu, Y. Mati y Z. Binder. A tabu search heuristic for the single vehicle pickup and delivery problem with time windows. *Journal of Intelligent Manufacturing*, 12(5–6):497–508, 2001.

[38] G. Laporte. The vehicle routing problem: An overview of exact and approximate algorithms. *European Journal of Operational Research*, 59(3):345–358, 1992.

[39] J. K. Lenstra y A. H. G. Rinnooy Kan. Complexity of vehicle routing and scheduling problems. *Networks*, 11(2):221–227, 2006.

[40] H. Li y A. Lim. A metaheuristic for the pickup and delivery problem with time windows. En *Proceedings of the 13th IEEE International Conference on Tools with Artificial Intelligence*, pp. 160–167, Dallas, EUA, Noviembre 2001.

[41] Q. Lu y M. Dessouky. An exact algorithm for the multiple vehicle pickup and delivery problem. *Transportation Science*, 38(4):503–514, 2004.

[42] Q. Lu y M. M. Dessouky. A new insertion-based construction heuristic for solving the pickup and delivery problem with time windows. *European Journal of Operational Research*, 175(2):672–687, 2006.

[43] J. MacQueen. Some methods for classification and analysis of multivariate observations. En L. M. Le Cam y J. Neyman, editores, *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, volumen 1, pp. 281–297. University of California Press, Berkeley, EUA, 1967.

[44] J. E. Mendoza, B. Castanier, C. Guéret, A. L. Medaglia y N. Velasco. A memetic algorithm for the multi-compartment vehicle routing problem with stochastic demands. *Computers & Operations Research*, 37(11):1886–1898, 2010.

[45] J. E. Mendoza, B. Castanier, C. Guéret, A. L. Medaglia y N. Velasco. Constructive heuristics for the multi-compartment vehicle routing problem with stochastic demands. *Transportation Science*, 45(3):346–363, 2011.

[46] S. Mitrović-Minić, R. Krishnamurti y G. Laporte. Double-horizon based heuristics for the dynamic pickup and delivery problem with time windows. *Transportation Research Part B: Methodological*, 38(8):669–685, 2004.

[47] S. Mitrović-Minić y G. Laporte. Waiting strategies for the dynamic pickup and delivery problem with time windows. *Transportation Research*, 38(7):635–655, 2004.

[48] N. Mladenović, D. Urošević, S. Hanafi y A. Ilić. A general variable neighborhood search for the one-commodity pickup-and-delivery travelling salesman problem. *European Journal of Operational Research*, 220(1):270–285, 2012.

[49] G. Nagy, N. A. Wassan y M. G. Speranza. The vehicle routing problem with divisible deliveries and pickups. Working Paper 1, University of Kent, Canterbury, Inglaterra, Mayo 2009.

[50] W. P. Nanry y J. W. Barnes. Solving the pickup and delivery problem with time windows using reactive tabu search. *Transportation Research Part B: Methodological*, 34(2):107–121, 2000.

[51] M. Nowak, Ö. Ergun y C. C. White. Pickup and delivery with split loads. *Transportation Science*, 42(1):32–43, 2008.

[52] J. A. Pacheco. Heurístico para los problemas de rutas con carga y descarga en sistemas LIFO. *Quaderns d'Estadística, Sistemes, Informatica i Investigació Operativa*, 21(1–2):153–175, 1997.

[53] J. A. Pacheco Bonrostro y C. R. Delgado Serna. Problemas de rutas con ventanas de tiempo y carga y descarga simultánea: Diseño de filtros en algoritmos de intercambio (caso de un vehículo). *Estudios de Economía Aplicada*, 7:79–100, 1997.

[54] G. Pankratz. A grouping genetic algorithm for the pickup and delivery problem with time windows. *OR Spectrum*, 27(1):21–41, 2005.

[55] C. Papadimitriou y M. Sideri. On the Floyd-Warshall algorithm for logic programs. *The Journal of Logic Programming*, 41(1):129–137, 1999.

[56] C. H. Papadimitriou. *Computational Complexity*. Addison-Wesley, Reading, EUA, 1993.

[57] S. N. Parragh, K. F. Doerner y R. F. Hartl. A survey on pickup and delivery problems, Part II: Transportation between pickup and delivery locations. *Journal für Betriebswirtschaft*, 58(2):81–117, 2008.

[58] D. Pisinger y S. Ropke. A general heuristic for vehicle routing problems. *Computers & Operations Research*, 34(8):2403–2435, 2007.

[59] C. Prins. A simple and effective evolutionary algorithm for the vehicle routing problem. *Computers & Operations Research*, 31(12):1985–2002, 2004.

[60] T. K. Ralphs, L. Kopman, W. R. Pulleyblank y L. E. Trotter. On the capacitated vehicle routing problem. *Mathematical Programming*, 94(2–3):343–359, 2003.

[61] J. Renaud y F. F. Boctor. A sweep-based algorithm for the fleet size and mix vehicle routing problem. *European Journal of Operational Research*, 140(3):618–628, 2002.

[62] S. Ropke y J.-F. Cordeau. Branch and cut and price for the pickup and delivery problem with time windows. *Transportation Science*, 43(3):267–286, 2009.

[63] S. Ropke, J.-F. Cordeau y G. Laporte. Models and branch-and-cut algorithms for pickup and delivery problems with time windows. *Networks*, 49(4):258–272, 2007.

[64] S. Ropke y D. Pisinger. An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows. *Transportation Science*, 40(4):455–472, 2006.

[65] S. Ropke y D. Pisinger. A unified heuristic for a large class of vehicle routing problems with backhauls. *European Journal of Operational Research*, 171(3):750–775, 2006.

[66] M. W. P. Savelsbergh y M. Sol. The general pickup and delivery problem. *Transportation Science*, 29(1):17–29, 1995.

[67] D. Sáez, C. E. Cortés y A. Núñez. Hybrid adaptive predictive control for the multi-vehicle dynamic pick-up and delivery problem based on genetic algorithms and fuzzy clustering. *Computers & Operations Research*, 35(11):3412–3438, 2008.

[68] M. M. Solomon. Algorithms for the vehicle routing and scheduling problems with time window constraints. *Operations Research*, 35(2):254–265, 1987.

[69] F.J. Srour, T. Máhr, M.M. de Weerdt y R.A. Zuidwijk. MIPLIB truckload PDPTW instances derived from a real-world drayage case. Reporte Técnico 2010-10-04, Erasmus Research Institute of Management (ERIM), Erasmus University Rotterdam, Holanda, Octubre 2010.

[70] Stat-Ease, Inc. *Design Expert 8.0 Manual*. Stat-Ease, Inc., Minnesota, EUA, 2010.

[71] F. A. Tang Montané y R. Diéguez Galvão. A tabu search algorithm for the vehicle routing problem with simultaneous pick-up and delivery service. *Computers & Operations Research*, 33(3):595–619, 2006.

[72] S. R. Thangiah, A. Fergany y S. Awan. Real-time split-delivery pickup and delivery time window problems with transfers. *Central European Journal of Operations Research*, 15(4):329–349, 2007.

[73] P. Toth y D. Vigo. The granular tabu search and its application to the vehicle-routing problem. *INFORMS Journal on Computing*, 15(4):333–346, 2003.

[74] K.-P. Urban. A guided simulated annealing search for solving the pick-up and delivery problem with time windows and capacity constraints. *International Journal of Logistics: Research and Applications*, 9(4):369–381, 2006.

[75] N. A. Wassan y I. H. Osman. Tabu search variants for the mix fleet vehicle routing problem. *Journal of the Operational Research Society*, 53(7):768–782, 2002.

[76] M. Wen, J. Larsen, J. Clausen, J.-F. Cordeau y G. Laporte. Vehicle routing with cross-docking. *Journal of the Operational Research Society*, 60(12):1708–1718, 2007.

[77] J. Yang, P. Jaillet y H. Mahmassani. Real-time multivehicle truckload pickup and delivery problems. *Transportation Science*, 38(2):135–148, 2004.

Apéndice A

Análisis de Complejidad

En esta sección se utilizan algunos resultados de la teoría de Complejidad Computacional, la cual para un mayor entendimiento pueden consultarse en [56]. Se mostrará que el Modelo A, presentado en la Sección 3.3 el cual resuelve un problema de carga y descarga de productos con un conjunto de restricciones propias al problema estudiado es un problema NP-completo.

Para probar que un problema de optimización X es NP-duro, se necesita probar que su versión decisión es NP-completo. Los tres pasos básicos para probar que el problema decisión X es NP-completo son los siguientes:

1. Probar que el problema $X \in NP$.
2. Exponer que un problema X^0 NP-completo puede ser reducido polinomialmente a X (denotado como $X^0 \propto X$).
3. Probar la equivalencia de las soluciones, la respuesta de una instancia al problema X es «SI», si solo si la respuesta a la instancia relacionada al problema X^0 es «SI».

Definición de la versión decisión

Para poder demostrar que el Modelo A pertenece a los problemas NP, primeramente se presentará el modelo en su forma decisión:

ENTRADAS: Los parámetros $c_t^{\text{above}}, c_t^{\text{below}}, \tau_t^l, \tau_t^u, \sigma(t), \sigma(k), n_{ip}^+, n_{ip}^-, a_i, b_i, o_p, c_{ij}^s, c_{ij}^d, s_{ij}^s, s_{ij}^d, f_{ih}$; $i, j \in V, t \in T = T^e \cup T^w \supseteq T_i, p \in P = P_n \cup P_r$ y los escalares $S, M_T, M_L, M_C, C, Y \geq 0$

PREGUNTA: Existe algún valor de las variables $w_{kt}, x_{ijk}, y_{ikh}, z_k, v_k^{\text{above}}, v_k^{\text{below}}, v_k^{\text{type}}, g_{ikp}^-, g_{ikp}^+, q_{ikp}, l_k, u_k, s_{ik}, c_{ik}, OC_k, OT_k$ del Modelo A $i, j \in V, t \in T = T^e \cup T^w \supseteq T_i, k \in K \supseteq K_i, p \in P = P_n \cup P_r$ que satisface las restricciones (3.2)-(3.40) tal que:

$$\sum_k OC_k + \sum_k Cz_k \leq Y?$$

Probar que $X \in NP$

Suponiendo que se tiene una solución para el Modelo A, determinar si la solución es factible para el modelo se requiere evaluar cada restricción. Como se tiene un número polinomial de restricciones y variables se requiere realizar solamente un número polinomial de pasos para determinar si es factible o no. Por consiguiente el problema X pertenece al grupo de problemas tipo NP.

Reducción polinomial

En esta sección se demuestra la complejidad computacional del Modelo A.

Teorema A.1

$$TSPTW \propto X$$

Demostración. Consideremos una instancia particular de nuestro problema X donde solamente existe un único tipo producto retornable ($P_N = \emptyset$ y $|P| = 1$). Además

existen más andenes que vehículos disponibles ($f_{ih} \geq |K|\,; i \in V, h \in H$). De igual modo todos lo vehículos inician en un único lugar, que sin pérdida de generalidad para esta demostración se asignara arbitrariamente al nodo 0 ($K = K_0$ y $T = T_0$). Además en esta instancia particular los clientes requieren únicamente una unidad de producto ($n^-_{i0} = 1; i \in V \setminus \{0\}$) y el nodo planta además posee la capacidad de satisfacer a todos los clientes ($n^+_{00} \geq |V| - 1$). Finalmente para la instancia particular los vehículos no tienen restricción de capacidad ($v_k^{\text{below}} = +\infty; k \in K$) ni restricción de ventanas de tiempo ($\tau^l_t = 0, \tau^u_t = +\infty; t \in T$).

Al tener una instancia con dichas características las restricciones relacionadas con la capacidad de los andenes (3.17)-(3.19) y (3.32) siempre serán cumplidas, debido a que existe una mayor cantidad de andenes que vehículos disponibles. Además al no tener productos no retornables, las restricciones para evitar aplastar este tipo de productos (3.31) siempre serán satisfechas.

Las restricciones relacionadas a las ventanas de tiempo de los vehículos (3.15) y (3.16); las restricciones relacionadas a no exceder la capacidad de los vehículos (3.22)-(3.26), (3.29), (3.30), (3.34) y (3.35); junto con las restricciones relacionadas a la configuración de los vehículos (3.2)-(3.8), (3.10) y (3.33) siempre serán satisfechas, ya que los vehículos no tienen restricción de capacidad ni tampoco cuentan con restricciones de ventanas de tiempo.

Como la instancia posee solamente un tipo de producto sería posible omitir escribir el subíndice p obviando que se trata del único elemento del conjunto P. Es decir la variable q_{ikp} puede escribirse como q_{ik} así:

q_{ikp} Carga del producto p que tiene el vehículo k al terminar de visitar al nodo i.
q_{ik} Carga que tiene el vehículo k al terminar de visitar al nodo i.

Entonces la formulacíon para este caso particular quedaría de la siguiente manera:

$$\text{Minimizar} \quad \sum_{k \in K} OC_k + \sum_{k \in K} Cz_k \tag{A.1}$$

Sujeto a:

$$x_{ijk} \leq z_k \qquad i,j \in V, k \in K \qquad \text{(A.2)}$$

$$\sum_{i:(i,j)\in E} x_{ijk} - \sum_{i:(j,i)\in E} x_{jik} = 0 \qquad j \in V, k \in K \qquad \text{(A.3)}$$

$$\sum_{i:(i,j)\in E} x_{ijk} \leq 1 \qquad j \in V, k \in K \qquad \text{(A.4)}$$

$$a_i \sum_{j:(j,i)\in E} x_{jik} \leq s_{ik} \leq b_i \sum_{j:(j,i)\in E} x_{jik} \qquad i \in V, k \in K \qquad \text{(A.5)}$$

$$a_{\sigma(k)} z_k \leq OT_k \leq b_{\sigma(k)} z_k \qquad i \in V, k \in K \qquad \text{(A.6)}$$

$$s_{ik} + S + s^s_{ij} \qquad (i,j) \in E | j \neq \sigma(k), \qquad \text{(A.7)}$$
$$-M_T\,(1 - x_{ijk}) \leq s_{jk} \qquad k \in K$$

$$s_{ik} + S + s^s_{i\sigma(k)}$$
$$-M_T\left(1 - x_{i\sigma(k)k}\right) \leq OT_k \qquad k \in K, i \in V \qquad \text{(A.8)}$$

$$c_{ik} + c^s_{ij} \qquad k \in K, i \in V,$$
$$-M_C\,(1 - x_{ijk}) \leq c_{jk} \qquad j \in V \setminus \{\sigma(k)\} \qquad \text{(A.9)}$$

$$c_{ik} + c^s_{i\sigma(k)}$$
$$-M_C\left(1 - x_{i\sigma(k)k}\right) \leq OC_k \qquad k \in K, i \in V \qquad \text{(A.10)}$$

$$x_{ijk}, z_k \in \{0,1\} \qquad i,j \in V, t \in T, k \in K, \qquad \text{(A.11)}$$

$$s_{ik}, c_{ik}, OC_k, OT_k \geq 0 \qquad i \in V, k \in K \qquad \text{(A.12)}$$

Esta es precisamente la formulación del VRPTW, con lo cual se concluye que VRPTW se reduce polinomialmente al problema del Modelo A. Como el VRPTW es NP-completo [39], entonces el problema del Modelo A su vez es NP-completo. De lo anterior surge el siguiente corolario:

Corolario 1 *El problema de optimización del Modelo A es NP-duro.*

Dado que finalmente el problema se pudo reducir a un problema de VRPTW, el problema es al menos tan difícil que un VRPTW, y como el VRPTW es un problema NP-duro el modelo también forma parte de los problemas NP-duro.

Ficha autobiográfica

Andrés Castrillón Escobar

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

Universidad Autónoma de Nuevo León

Facultad de Ingeniería Mecánica y Eléctrica

Tesis:

Una metaheurística para un problema de carga y descarga en la repartición de bebidas embotelladas

Nació el 2 de diciembre de 1987 en Monterrey, N.L. En los primeros años de estudio no fue considerado como un alumno destacado, sin embargo el esfuerzo y las ganas de ser mejor han hecho de él una persona trabajadora y destacada en sus áreas de interés. Realizando sus estudios de preparatoria desarrolló un gusto por las matemáticas y la computación inclinándose específicamente en las áreas de desarrollo de software. Realizó sus estudios de Licenciatura en la Universidad Autónoma de Nuevo León, graduándose de Lic. en Ciencias Computacionales en el año del 2010 por parte de la Facultad de Ciencias Físico-Matemáticas. Continúo sus estudios de Maestría en Ciencias en Ingeniería de Sistemas por parte de la misma Universidad en la Facultad de Ingeniería Mecánica y Eléctrica.