





**Universidad Autónoma de Ciudad Juárez**

**Instituto de Ingeniería y Tecnología**

**Departamento de Ingeniería Eléctrica y Computación**

**Maestría en Cómputo Aplicado**

***“Búsqueda Adaptativa de Vecindad Amplia para la optimización de rutas en base al uso de bahías en el Problema Extendido del Camión y su Remolque”***

Tesis para obtener el grado de Maestro en Cómputo Aplicado

**Jared Arturo Olmos Villagrán**

“Becado por el Consejo Nacional de Ciencia y Tecnología”

**Bajo la dirección del**

**Dr. Rogelio Florencia Juárez**

**Codirector de tesis:**

**Dr. Francisco López Ramos**

Ciudad Juárez, Chihuahua enero de 2021

## **Declaración de originalidad**

Cd. Juárez, Chih., a enero del 2021

Por medio de la presente, yo, Jared Arturo Olmos Villagrán como autor de la presente tesis, declaro que es por el producto de mi trabajo original que se realizó la redacción del presente documento. No infringe derechos de terceras personas tales como derechos de autor, patentes entre otros. Por lo que me declaro como autor y titular de los derechos del mismo.

Así mismo, declaro que se referencia explícitamente cualquier cita o resumen de otros autores, con sus publicaciones correspondientes, a lo largo del documento. Acepto cualquier reclamación, si se presenta, en cuanto a derechos de autor de parte de terceros, tomando como propia la responsabilidad que esto conlleva.



Ing. Jared Arturo Olmos Villagrán

## **Agradecimientos**

Agradezco a mis asesores de tesis, Dr. Rogelio Florencia Juárez y Dr. Francisco López Ramos, por su paciencia, conocimientos y consejos brindados a lo largo de la maestría. A mis profesores, por sus valiosas clases y tiempo por compartir su conocimiento. A la Universidad Autónoma de Ciudad Juárez por las facilidades e instalaciones otorgadas para el desarrollo del documento de tesis. Al Consejo Nacional de Ciencia y Tecnología (CONACyT) por el apoyo para que este trabajo sea realidad.

## **Dedicatoria**

Dedico este trabajo a Dios, a mis padres y a mi familia por todo el apoyo recibido antes y durante el proceso de la maestría y el desarrollo de la tesis.

A Dios, por darme la oportunidad de tener salud y los dones para desarrollarlos.

A mi padre, por demostrarme lo que es vivir con objetivos y alegría.

A mi madre, por darme la vida y la motivación para seguir.

A mi hermano, por apoyarme y ser ejemplo a seguir.

A mis abuelos, por cuidarme como segundos padres y brindarme ese amor de abuelo.

A mis tíos, por apoyarme en todo de distintas maneras.

A mis primos, por las alegrías y buenos momentos que pasamos.

**Tabla de contenido**

<b>CAPÍTULO I</b> .....	<b>14</b>
<b>1.1. Introducción</b> .....	<b>15</b>
<b>1.2. Descripción del problema</b> .....	<b>15</b>
<b>1.3. Objetivos</b> .....	<b>16</b>
<b>1.3.1. General</b> .....	<b>16</b>
<b>1.3.2. Particulares</b> .....	<b>16</b>
<b>1.4. Justificación</b> .....	<b>16</b>
<b>1.5. Alcances y Limitaciones</b> .....	<b>17</b>
<b>1.5.1. Alcances</b> .....	<b>17</b>
<b>1.5.2. Limitaciones</b> .....	<b>17</b>
<b>1.6. Impacto</b> .....	<b>18</b>
<b>CAPÍTULO II</b> .....	<b>19</b>
<b>2.1. Problema de ruteo de vehículos</b> .....	<b>20</b>
<b>2.2. Problema de ruteo del camión y su remolque</b> .....	<b>20</b>
<b>2.2.1. Problema de ruteo del camión y su remolque con ventanas de tiempo</b> .....	<b>21</b>
<b>2.2.2. Problema de ruteo del camión y su remolque con capacidad</b> .....	<b>21</b>
<b>2.3. Complejidad Computacional</b> .....	<b>22</b>
<b>2.4. Algoritmos metaheurísticos</b> .....	<b>23</b>
<b>2.5. Búsqueda Adaptativa de Vecindad Amplia</b> .....	<b>23</b>
<b>CAPÍTULO III</b> .....	<b>25</b>
<b>3.1. Revisión de la literatura</b> .....	<b>26</b>
<b>3.2. Contribuciones</b> .....	<b>28</b>
<b>CAPÍTULO IV</b> .....	<b>31</b>

<b>4.1 Metodología.....</b>	<b>32</b>
<b>4.2. Desarrollo del prototipo.....</b>	<b>32</b>
<b>4.2.1. Modelo de optimización para AMPL .....</b>	<b>33</b>
<b>4.2.2. Representación de la solución.....</b>	<b>39</b>
<b>4.2.3. Desarrollo de algoritmo Búsqueda Adaptativa de Vecindad Amplia.....</b>	<b>41</b>
<b>4.2.4. Operadores de destrucción e inserción.....</b>	<b>43</b>
<b>4.3. Instancias de prueba .....</b>	<b>53</b>
<b>4.3.1. Identificación del área de estudio.....</b>	<b>54</b>
<b>4.3.2. Identificación de proveedores y de bahías de estacionamiento .....</b>	<b>55</b>
<b>4.3.3. Identificación de clientes.....</b>	<b>56</b>
<b>4.3.4. Cálculo de distancias y estimación de tiempos de servicio.....</b>	<b>57</b>
<b>4.3.5. Cálculo de costo de uso de bahías .....</b>	<b>61</b>
<b>4.3.6. Descripción de pedidos.....</b>	<b>62</b>
<b>CAPÍTULO V .....</b>	<b>64</b>
<b>5.1. Validación Experimental .....</b>	<b>65</b>
<b>5.2. Identificación de escenarios.....</b>	<b>65</b>
<b>5.3. Comparativa de rutas actuales con rutas resultantes .....</b>	<b>66</b>
<b>5.4. Evaluación de la solución propuesta de acuerdo a escenarios.....</b>	<b>69</b>
<b>5.5. Análisis de Resultados.....</b>	<b>70</b>
<b>CAPÍTULO VI .....</b>	<b>75</b>
<b>6.1. Conclusión.....</b>	<b>76</b>
<b>6.2. Trabajo futuro .....</b>	<b>77</b>

## Índice de tablas

<b>Tabla 1. Comparativa del presente trabajo con trabajos de la literatura .....</b>	<b>29</b>
<b>Tabla 2. Elementos de entrada del ALNS .....</b>	<b>40</b>
<b>Tabla 3. Pesos asignados al par de operadores de destrucción-reparación .....</b>	<b>51</b>
<b>Tabla 4. Análisis de porcentaje de cobertura .....</b>	<b>57</b>
<b>Tabla 5. Distancias en metros de bahías.....</b>	<b>58</b>
<b>Tabla 6. Tiempo de desplazamiento en minutos de bahías.....</b>	<b>59</b>
<b>Tabla 7. Muestra de distancia en metros desde bahías a clientes .....</b>	<b>60</b>
<b>Tabla 8. Muestra de tiempo de desplazamiento en minutos desde bahías hasta clientes</b>	<b>60</b>
<b>Tabla 9. Costos de utilización de bahías.....</b>	<b>61</b>
<b>Tabla 10. Pedidos de Empresa 1 .....</b>	<b>62</b>
<b>Tabla 11. Lista de escenarios.....</b>	<b>65</b>
<b>Tabla 12. Ruta óptima y ganancia .....</b>	<b>69</b>
<b>Tabla 13. Rutas y ganancias de E1, día lunes y vehículo 2 .....</b>	<b>69</b>
<b>Tabla 14. Lista de escenarios.....</b>	<b>70</b>
<b>Tabla 15. Porcentaje de mejora de escenarios .....</b>	<b>70</b>
<b>Tabla 16. Rutas con mejora por reducción de bahías .....</b>	<b>71</b>
<b>Tabla 17. CSV para análisis de rutas óptimas .....</b>	<b>71</b>

## Índice de figuras

<b>Figura 1. Esquema general de la solución recalcando la aportación de la investigación .....</b>	<b>33</b>
<b>Figura 2. Interacción AMPL-presolvedor-resolvedor .....</b>	<b>35</b>
<b>Figura 3. Sistema de archivos de AMPL.....</b>	<b>36</b>
<b>Figura 4. Archivo “.run” .....</b>	<b>36</b>
<b>Figura 5. Archivo “.mod” .....</b>	<b>37</b>
<b>Figura 6. Archivo “.dat” .....</b>	<b>38</b>
<b>Figura 7. Archivo solución “.log”.....</b>	<b>39</b>
<b>Figura 8. Grafo representando un caso ilustrativo del ETTRP .....</b>	<b>44</b>
<b>Figura 9. Ejemplo de prueba ALNS con solución inicial, ganancia, pesos de selección y operador de destrucción seleccionado .....</b>	<b>45</b>
<b>Figura 10. Ejemplo de prueba ALNS con clientes eliminados utilizando el par peor-voraz .....</b>	<b>46</b>
<b>Figura 11. Ejemplo de prueba ALNS con solución parcial, fitness y actualización de pesos de selección .....</b>	<b>47</b>
<b>Figura 12. Ejemplo de prueba ALNS con Solución final y ganancia.....</b>	<b>48</b>
<b>Figura 13. Ejemplo operador de eliminación aleatoria.....</b>	<b>49</b>
<b>Figura 14. Ejemplo operador de eliminación del peor.....</b>	<b>50</b>
<b>Figura 15. Ejemplo operador de eliminación por clúster .....</b>	<b>51</b>
<b>Figura 16. Ejemplo operador de eliminación relacional .....</b>	<b>52</b>
<b>Figura 17. Ejemplo operador de inserción voraz .....</b>	<b>52</b>
<b>Figura 18. Ejemplo operador de inserción voraz .....</b>	<b>53</b>
<b>Figura 19. Delimitación del área de estudio .....</b>	<b>54</b>
<b>Figura 20. Ubicación de bahías de estacionamiento .....</b>	<b>55</b>
<b>Figura 21. Ubicación de clientes y su clasificación .....</b>	<b>56</b>

<b>Figura 22. Análisis de porcentaje de cobertura .....</b>	<b>57</b>
<b>Figura 23. Ruta actual de día lunes de Empresa 1 del vehículo 2 .....</b>	<b>68</b>
<b>Figura 24. Ruta actual transformada para ALNS .....</b>	<b>68</b>
<b>Figura 25. Boxplot de porcentaje de mejora por empresa .....</b>	<b>72</b>
<b>Figura 26. Gráfico de dispersión de mejora según empresa .....</b>	<b>73</b>
<b>Figura 27. Gráfico de dispersión de mejora según escenario .....</b>	<b>74</b>

**Índice de algoritmos**

<b>Algoritmo 1. Metaheurística de ALNS .....</b>	<b>42</b>
<b>Algoritmo 2. Eliminación aleatoria.....</b>	<b>43</b>
<b>Algoritmo 3. Pesos asignados al par de operadores de destrucción-reparación.....</b>	<b>44</b>
<b>Algoritmo 4. Eliminación de clústeres .....</b>	<b>45</b>
<b>Algoritmo 5. Eliminación Relacional .....</b>	<b>47</b>
<b>Algoritmo 6. Inserción voraz .....</b>	<b>47</b>
<b>Algoritmo 7. Inserción <i>K</i>-Arrepentimientos .....</b>	<b>48</b>
<b>Algoritmo 8. Método constructivo de asignación bahía-cliente con planificación de vehículos .....</b>	<b>67</b>

## Resumen

En las empresas de distribución de mercancías, generalmente, se presentan problemas de logística debido al gran número de factores que se involucran en el proceso y a la informalidad con que se aborda. Uno de estos problemas es el que se describe en el presente documento, el cual se acentúa con la manera empírica de generar rutas de reparto para distribuir mercancías. Esto último genera tres complicaciones, elevado costo en la distribución, incumplimiento o empalmes del uso de bahías (esto es, lugares de estacionamiento) entre las distintas empresas y la insatisfacción con la totalidad de pedidos de los clientes.

Este problema se asocia con lo que en la literatura se conoce como Problema Extendido del Camión y su Remolque (ETTRP, por sus siglas en inglés). Se denomina extendido ya que el problema de esta investigación aborda las restricciones de capacidad, demanda, horarios de descarga-entrega y el uso de distintas bahías o estacionamiento en el contexto de cuatro empresas en el centro histórico de una ciudad en México. Esta investigación presenta una propuesta de solución al ETTRP basada en la hibridación de un generador de rutas inicial, un optimizador de programación binaria (AMPL) y un algoritmo metaheurístico de búsqueda adaptativa de vecindad amplia (ALNS) para la optimización de la ganancia de las empresas.

El uso de esta hibridación no se ha realizado dentro de la literatura consultada, por lo que se aporta un método innovador y funcional para abordar este tipo de problemas. La experimentación realizada da evidencia de la viabilidad de utilizar este enfoque en la búsqueda de rutas que optimizan la ganancia en el proceso de distribución de mercancías.

## Abstract

Usually in companies that distribute products, there are logistic problems due to the large number of variables that are involved in the process and the informality with which is approached. One of these problems, it is the one that is described in this document, this is, the empirical way that companies generates delivery routes to distribute the products. This last thing generates three complications, a high distribution cost, schedule splices on parking bays by trucks and trailers and failure in the fulfillment of customer orders.

This problem is associated with a concept known in the literature as the Extended Truck and Trailer Routing Problem (ETTRP). It is defined as extended due to the extensive constraints that are involved, such as, capacity, demand, time scheduling and the use of distinct parking bays, in the context of four companies in the historical center of a Mexican city. This research describes a proposal solution to the ETTRP based on a hybridization of an initial route generator, a binary programming optimizer (AMPL) and an Adaptive Large Neighborhood Search metaheuristic algorithm (ALNS) seeking in optimizing the profit of these companies.

The use of this hybrid approach has not been used in the consulted literature, so it represents an innovative and functional approach to solve this kind of problems. The experiments carried out prove the viability of the solution in searching routes that optimize the profit in the product distribution process.

## Capítulo I

### Planteamiento del problema

---

En este capítulo se introduce el problema de investigación, el problema de ruteo del camión y su remolque, describiendo el problema de las cuatro empresas en el contexto seleccionado del centro histórico de la ciudad. Así mismo, se describen los objetivos de la investigación, la justificación, los alcances y limitaciones, así como el impacto del mismo.

## 1.1. Introducción

Generalmente, en las empresas de distribución de mercancías existe una problemática de logística referente a la repartición, dentro del proceso de la cadena de suministros, debido a la informalidad del mismo. Esta problemática podría generar pérdidas económicas si no se satisface la demanda de todos los clientes o bien, se satisface, pero con sobrecosto.

Este problema se asocia con lo que en la literatura se conoce como el Problema Extendido del Camión y su Remolque (ETTRP, por sus siglas en inglés). En este problema se pretende encontrar una ruta que minimice la distancia que recorren los vehículos para visitar a un conjunto de clientes. Los vehículos de este problema se componen de un camión y de remolques que pueden acoplarse y desacoplarse a dichos camiones. Por el motivo anterior, existe una clasificación de clientes, aquellos que pueden ser atendidos por un camión con su remolque (*cliente vehículo*), y aquellos que solo son atendidos por un camión con remolque desacoplado (*cliente camión*). Con las restricciones anteriores, se presentan dos tipos de rutas, las *rutas completas de vehículo* y las *rutas puras de vehículo*. Las primeras se utilizan para ambos tipos de clientes, mientras que las segundas se utilizan únicamente en *clientes camión* [1]. La resolución de este problema representa una mejora en las ganancias adoptadas por las empresas debido a que se busca minimizar la distancia recorrida y, por ende, minimizar el coste de logística [2]. Debido al carácter de mejorar la ganancia en las empresas de este contexto de investigación, este trabajo aborda el desarrollo de la segunda etapa (optimizador de programación binaria y desarrollo de algoritmo metaheurístico) de una solución propuesta compuesta por una hibridación de un generador de rutas inicial, un optimizador de programación binaria (AMPL) y un algoritmo metaheurístico de búsqueda adaptativa de vecindad amplia (ALNS) para resolver el ETTRP y maximizar las ganancias de las empresas.

## 1.2. Descripción de la problemática

En el contexto de esta investigación existen empresas que realizan diversas entregas a la semana a un total de 48 clientes, compartiendo el uso de espacios de estacionamiento de carga y descarga aledaños a los clientes, conocidos como bahías. El orden de entrega desde cada bahía no es a priori conocido por las empresas y sus operarios. Para distribuir estos productos, se cuentan con 15 bahías disponibles para el arribo de los camiones y el posterior inicio del proceso de reparto, que se describe a continuación. En primer lugar, las rutas que siguen los camiones para realizar el reparto de mercancías las establecen cada uno de los choferes a partir de su experiencia o conocimiento empírico. Además, el camión llega a una bahía y el

personal de la empresa reparte los productos, mediante carretillas de carga a los clientes cercanos a la bahía. Por lo tanto, se derivan problemas tales como:

1. Planeación inadecuada de las rutas para la entrega de la mercancía de los clientes impactando en la ganancia de las empresas.
2. Incumplimiento del uso de bahías lo cual genera gastos de recursos debido a empalmes entre camiones de diferentes empresas.
3. Incumplimiento del horario de entrega de la mercancía al cliente debido a: 1) el tiempo de espera elevado para poder descargar la mercancía en las bahías, y 2) al método empírico de generación de rutas lo cual podría ocasionar pérdidas en ventas.

### **1.3. Objetivos**

A continuación, se presentan los objetivos de la tesis (general y particulares).

#### **1.3.1. General**

Optimizar la generación de rutas de reparto en base a las bahías de estacionamiento, utilizando una metaheurística basada en una búsqueda adaptativa de vecindad amplia para incrementar la ganancia.

#### **1.3.2. Particulares**

- Modelar matemáticamente el proceso de distribución de productos, considerando ventanas de tiempo, capacidad de los vehículos y demanda de los clientes.
- Desarrollar una metaheurística basada en una búsqueda adaptativa de vecindad amplia para la generación de las rutas de reparto de productos.
- Evaluar el desempeño de la metaheurística propuesta.

### **1.4. Justificación**

El proceso de entrega de mercancías en una empresa proveedora representa el 20% total de sus gastos [2]. Por lo anterior, es importante optimizar y mejorar dicho proceso para reducir el gasto y, con esto, mejorar la ganancia que produce la empresa. Al contar con una gran cantidad de clientes a atender y restricciones de capacidad de vehículos, demanda de mercancía y ventanas temporales por parte de los clientes, la implementación de una metaheurística es recomendable para encontrar una buena solución en tiempo razonable.

## 1.5. Alcances y Limitaciones

A continuación, se describen los alcances y limitaciones del trabajo de investigación.

### 1.5.1. Alcances

Este sistema se desarrolló para el dominio de empresas proveedoras de mercancías que entregan producto en el centro histórico de una ciudad en México, las cuales comparten bahías de estacionamiento donde descargan sus mercancías.

Entre los alcances definidos para el proyecto destacan los siguientes:

- Se modelan restricciones tales como: 1) capacidad de los vehículos, 2) ventanas de tiempo de las bahías y los clientes y 3) satisfacción de la demanda de los clientes.
- Se contempla una demanda promedio de 12 cajas de mercancía y de 1 a 3 vehículos para la entrega de la misma.
- Se contemplan costos en las bahías de estacionamientos, por lo que la configuración de los mismo repercute en la optimización de la ganancia de las rutas.

### 1.5.2. Limitaciones

Entre las limitaciones que se presentaron durante el desarrollo de este trabajo de investigación se mencionan las siguientes:

- La obtención de los permisos de las empresas para trabajar con los datos reales, por lo que se generaron instancias sintéticas que siguen las propiedades de los datos reales.
- Se contempla la situación en donde el uso de bahías de estacionamiento que se comparten con varios clientes.
- No se contemplan clientes que requieren productos que por sus dimensiones no pueden ser transportados por una carretilla de carga.
- Las rutas a optimizar se toman de un generador de rutas iniciales basado en un algoritmo de  $k$ -caminos mínimos.

## 1.6. Impacto

- **Social:** Con el desarrollo e implementación de la ALNS se minimiza el tiempo y distancia que recorran los camiones de las empresas en la ciudad. Al disminuir estos valores habrá un impacto social positivo debido a que el tráfico por donde transitan estos camiones disminuirá.
- **Económico:** El impacto económico se presenta directamente en las empresas que utilicen esta propuesta. Debido a que el propósito principal es minimizar costos relacionados con la distribución de mercancías, esto permite lograr una disminución de recursos destinados a este proceso. Así mismo, se buscan rutas que prioricen la ganancia de las empresas al atender a clientes que representen un aumento en las mismas.
- **Tecnológico o académico:** Al ser un problema muy restrictivo, esto es considerar condiciones de ventanas temporales, capacidad y múltiples horarios de descarga y entrega de mercancías, no existen trabajos relacionados que contemplen todos los elementos que se abordan en el problema. Por esto, la elaboración del presente trabajo contribuyó a un desarrollo tecnológico que permite abordar un problema que, hasta nuestro entendimiento, nunca antes ha sido resuelto y sirve como base para trabajos futuros o trabajos con restricciones similares.

## Capítulo II

### Marco teórico

---

En este capítulo se describen las bases teóricas del proyecto de investigación. En primer lugar, se describe el Problema de Ruteo de Vehículos. A continuación, se detalla un problema más específico, el Problema de Ruteo del Camión y su Remolque, así como los derivados del mismo, el Problema de Ruteo del Camión y su Remolque con ventanas de tiempo y con capacidad que en su conjunto se le conoce como Problema Extendido del Camión y su Remolque. Por último, se describe el significado de complejidad computacional como fundamento para la resolución de este tipo de problemas combinatorios con enfoque metaheurístico.

## 2.1. Problema de ruteo de vehículos

El problema de ruteo de vehículos (VRP, por sus siglas en inglés) es un problema de carácter combinatorio NP-duro [3]. El problema consiste en visitar a un conjunto de clientes partiendo y terminando en un depósito, dado un vehículo.

El objetivo es visitar a cada cliente satisfaciendo su demanda y minimizando el costo asociado a la entrega de productos con la condición de salir y de regresar al nodo depósito [4]. Formalmente se define de la manera siguiente: sea  $G = (V, A)$  un grafo donde  $V = \{1, \dots, n\}$  es un conjunto de vértices y  $A = \{(i, j) : i, j \in V\}$  un conjunto de aristas. El vértice 0 representa el depósito inicial y final, mientras que el resto de los vértices representan a los clientes. Cada arista tiene asociado una distancia no negativa llamada costo  $C_{i,j}$  que representa la distancia o el tiempo que tarda el vehículo en trasladarse del vértice  $i$  al vértice  $j$  [5]. Existen diversas variantes de este problema, siendo una de estas el problema de ruteo del camión y su remolque.

## 2.2 Problema de ruteo del camión y su remolque

El problema de ruteo del camión y su remolque (TTRP, por sus siglas en inglés) es una integración de los problemas clásicos del VRP y del problema de localización de instalaciones (FLP, por sus siglas en inglés), cuyo objetivo es encontrar el arreglo más efectivo en concordancia con requerimientos puntuales tales como puntos de carga/descarga, posición, tamaño entre otros [6]. En el TTRP se utilizan vehículos compuestos por un camión y un tráiler o remolque con la capacidad de desacoplarse del camión. Esta variante representa de mejor manera el mundo real en comparación al VRP. Formalmente se define de la manera siguiente: sea  $V = \{0, 1, 2, \dots, n\}$  un conjunto de nodos clientes donde el 0 representa el depósito y  $E$  el conjunto de aristas o arcos entre dichos nodos en  $V$ , entonces  $G = \{V, E\}$  es un grafo completo en donde cada punto  $i \in V \setminus \{0\}$  tienen una demanda positiva  $q_i$  y un índice igual a 1 representando que un cliente es llamado de *tipo camión* (aquel que puede ser visitado solo por un camión sin remolque) o un 0 representando que el cliente es llamado de *tipo vehículo* (aquel que puede ser visitado por un camión acoplado a su remolque, llamado vehículo completo, o solo por el camión). Cada arco  $E$  tiene asociado un costo  $C_{i,j}$  no negativo y simétrico, definido por la distancia Euclidiana entre el nodo  $i$  y el nodo  $j$ . En el TTRP, un conjunto de rutas  $M = \{m_i\} \cup \{m_k - m_i\}$  son construidas, las cuales consisten en  $m_i$  rutas *completas de vehículo* o rutas *puras de vehículo* y un  $m_k - m_i$  rutas *puras de camión*, con el fin de que la distancia total recorrida por la flota sobre los tipos de rutas sea minimizada satisfaciendo todas las restricciones [1].

### 2.2.1 Problema de ruteo del camión y su remolque con ventanas de tiempo

El problema de ruteo del camión y su remolque con ventanas de tiempo (TTRPTW, por sus siglas en inglés) es una extensión del TTRP. En el TTRPTW se añade la restricción de ventanas temporales en los clientes, las cuales limitan el tiempo en que se puede entregar mercancía, y así, satisfacer la demanda. Formalmente se define de la manera siguiente: Sea un  $G = (V, A)$  grafo no dirigido donde  $V = \{0, 1, 2, \dots, n\}$  es un conjunto de vértices y  $A = \{(i, j): i, j \in V\}$  es el conjunto de aristas. El vértice 0 representa el depósito desde el cual se parte y al que se debe de llegar al final, mientras que los vértices que restan representan a los clientes. Cada vértice  $i$  tiene asociado una demanda no negativa  $d_i$ , una ventana de tiempo de servicio  $ET_i, LT_i$  un tiempo de servicio  $ST_i$  y un indicador del tipo de cliente  $CT_i$  (esto para representar si puede ser atendido por un camión sin remolque o por un vehículo completo, es decir, el camión junto con el remolque).  $ET_i$  y  $LT_i$  representan el tiempo más temprano y el tiempo más tarde que el servicio al cliente  $i$  puede realizarse, respectivamente.  $ST_i$  representa el tiempo de servicio utilizado para atender al cliente  $i$ .  $CT_i$  con valor 1 indica que el cliente es de tipo *cliente camión* (es decir, que solo un camión sin su remolque tiene acceso), mientras que un 0 indica que es de tipo *cliente vehículo* (es decir, que un vehículo completo puede acceder a él). Así mismo cada arco  $(i, j)$  tiene asociado un costo  $c_{i,j}$  que representa el costo, ya sea tiempo o distancia, necesaria para recorrerlo [7].

### 2.2.2 Problema de ruteo del camión y su remolque con capacidad

El problema de ruteo del camión y su remolque presenta otra extensión en la cual se aborda la restricción de capacidad. Este problema se conoce en la literatura como el problema de ruteo del camión y su remolque con capacidad (CTTRP, por sus siglas en inglés). Formalmente se define de la manera siguiente: dada una flota  $f_k$  de camiones y  $f_l$  remolques ( $f_k \geq f_l$ ) que atienden a un conjunto de clientes de un depósito central, se tienen unas capacidades fijas  $Q_k$  y  $Q_j$  respectivamente en la flota de camiones y remolques. Dado lo anterior se tiene que un vehículo completo tiene una capacidad total igual a  $Q_k + Q_j$ . Con la restricción anterior, y dado un grafo no dirigido  $G = (V, A)$  donde  $V = \{0, 1, 2, \dots, n\}$  representa un conjunto de vértices y  $A = \{(i, j): i, j \in V\}$  el conjunto de aristas, se tiene al vértice 0 que representa el depósito central, mientras que los vértices que restan representan a los clientes. Cada vértice  $i$  tiene asociado una demanda no negativa  $d_i$ , un tiempo de servicio  $ST_i$  y un indicador del tipo de cliente  $CT_i$  (representando si puede ser atendido por un camión sin remolque o un camión con remolque) con valor 1 cuando representa al cliente es de tipo *cliente camión* (solo un camión sin su remolque tiene acceso), mientras que un 0 representa que

es de tipo *cliente vehículo* (un vehículo completo puede acceder a él). Así mismo, cada arco  $(i, j)$  tiene asociado un costo  $c_{i,j}$  que representa el valor, ya sea tiempo o distancia, que se necesita para recorrerlo [1].

### 2.3. Complejidad Computacional

La resolución de problemas combinatorios lleva consigo la noción de un concepto denominado *complejidad computacional*. Este concepto hace referencia a la eficiencia que tiene un algoritmo, es decir, el procedimiento paso a paso necesario para resolver un problema. Michael R. Garey define un problema como una cuestión general a ser contestada que, generalmente, posee varios parámetros cuyos valores no están especificados [8]. Dicho problema puede definirse al dar: una descripción general de sus parámetros y un enunciado de las propiedades que debe tener la solución. Aunado a la definición anterior, el autor menciona que una instancia de un problema se obtiene al especificar particularmente los valores de todos los parámetros del problema, teniendo una longitud de entrada diferente por cada tipo de esquema de codificación que se utilice para representar una instancia dada.

Una función de complejidad de tiempo de un algoritmo representa los requisitos temporales, para cada longitud de entrada dada, expresando la cantidad máxima de tiempo que necesita para resolver una instancia del problema de tamaño particular. Por lo anterior, la función de complejidad de tiempo brinda una idea del tiempo necesario para resolver un problema dada una instancia y brinda información del tipo de problema en función al tiempo de resolución necesario. Existen problemas que pueden ser resueltos con algoritmos que tardan un tiempo polinomial y son de carácter determinista (P) y aquellos cuya solución se obtiene por un algoritmo cuya ejecución es de tipo no determinista en un tiempo polinomial (NP). Los problemas combinatorios generalmente entran en esta última categoría debido a su función de complejidad de tiempo de carácter exponencial, esto quiere decir que entre más grande el tamaño de la longitud de la entrada de la instancia el tiempo crece de manera exponencial.

La clase de complejidad NP-completa es aquella que representa un conjunto de todos los problemas  $x$  en NP para los cuales es posible reducir cualquier otro problema tipo NP de  $y$  a  $x$  en un tiempo polinomial. El problema de satisfacibilidad booleana es un ejemplo de un problema NP-completo. La NP-dureza de un problema indica que un problema es al menos tan difícil como un problema NP-completo, formalmente se puede decir que un problema  $x$  es NP-duro si existe un problema  $y$  NP-completo tal que  $y$  puede ser reducido a  $x$  en tiempo polinomial. El problema TTRPTW así como el CTTRP son de carácter NP-duro [8].

## 2.4. Algoritmos metaheurísticos

Los algoritmos metaheurísticos pueden considerarse como una manera eficiente de encontrar soluciones aceptables a problemas complejos por el principio de prueba y error. Son utilizados en problemas cuya complejidad hace imposible la búsqueda de cada posible solución o combinación con el objetivo de encontrar soluciones factibles en una escala de tiempo aceptables (denominados, NP-Complejos) [9]. Se diferencian de las heurísticas, ya que estas últimas están diseñadas para resolver un problema específico, mientras que las metaheurísticas están diseñadas para resolver un grupo de problemas con características similares [10]. Los algoritmos metaheurísticos se clasifican de distintas maneras, según Birattari estas clasificaciones se dan bajo 6 criterios, lo cuales son [11]:

- Continuidad en su búsqueda
- Número de puntos
- Capacidad de memoria
- Tipo de vecindario
- Función objetivo
- Fuente de inspiración

## 2.5. Búsqueda Adaptativa de Vecindad Amplia

El algoritmo de Búsqueda Adaptativa de Vecindad Amplia (ALNS) es una metaheurística propuesta por Ropke & Pisinger [12]. Se basa en la heurística de Búsqueda de Vecindad Amplia (LNS) en la cual se realiza una búsqueda local con algunas restricciones relajadas para garantizar una mayor movilidad dentro del espacio de soluciones de un problema y, con esto, tener mayor probabilidad de encontrar un óptimo global como solución. En comparación con las heurísticas de búsqueda local simples, las cuales suelen variar muy poco en la búsqueda de soluciones en cada iteración, esta búsqueda ampliada suele proporcionar mejores resultados [12]. Las mejoras de la metaheurística de ALNS con respecto a la LNS son tres:

1. La utilización de múltiples operadores de eliminación e inserción de elementos de la solución.
2. La utilización de heurísticos simples para dichos operadores, en lugar de la utilización de ramificación y poda de la LNS.
3. La utilización de recocido simulado como mecanismo de aceptación y variación de la solución.

Formalmente, la ALNS se define de la siguiente manera: dada una solución  $S$  factible inicial, un número de clientes  $q$  a eliminar por iteración, una probabilidad de aceptación  $p$  de utilizar una solución peor, una tasa de enfriamiento de la temperatura  $c$ , un factor de reacción  $r$  y unas puntuaciones  $\sigma_1$ ,  $\sigma_2$  y  $\sigma_3$  en función a la solución nueva obtenida se procede a realizar la búsqueda en el espacio de soluciones.

## **Capítulo III**

### **Literatura científica**

---

En este capítulo se detalla la revisión literaria del problema del camión y su remolque con restricciones de ventanas de tiempo y capacidad. Se presentan diversos trabajos que abordan el problema con diferentes enfoques. Así mismo, se detallan las contribuciones a la literatura que se desprenden de este trabajo de tesis.

### 3.1. Revisión de la literatura

Algunos trabajos relacionados con el problema de ruteo del camión y su remolque con restricciones de ventanas de tiempo y capacidad (CTTRPTW, por sus siglas en inglés) son el de Taillard [13] que aborda un VRP de la vida real, es decir, considera restricciones de capacidad, abordando restricciones adicionales a las que la literatura en su época abordaba, utiliza una búsqueda tabú para su resolución. Así mismo, Scheuere [14] aborda el problema de CTTRP utilizando Búsqueda Tabú, pero utilizando datos ficticios para su análisis. Lin et al. en [15] utiliza un recocido simulado para resolver el CTTRP utilizando, igualmente, datos ficticios para su experimentación. En [16] utilizan una formulación matemática junto con una heurística de inserción para abordar el problema, utilizan una combinación de representación gráfica basado en arcos y nodos, así como en rutas.

En [17] se aborda el problema del camión y su remolque con ventanas de tiempo (TTRPTW) con un algoritmo de *branch-and-price-and-cut*, minimizando la función objetivo que representa los costes de transportación. Este trabajo aborda, además, dos escenarios del mundo real, el primero, se refiere a que en el horizonte de la planeación se contempla una entrega de dos días, en donde el producto puede ser entregado uno u otro día, o incluso, ambos días, en cuyo caso haría que se recoja el doble de suministro diario para atender a dicho cliente. El segundo, aborda coste en los tiempos de transferencia de mercancía entre un remolque y su camión, para la entrega posterior de la mercancía a clientes que no pueden ser atendidos por el camión junto con su remolque. Por otro lado [18] proponen un algoritmo de tipo mate-heurístico (es decir, que combina métodos heurísticos y exactos) para resolver TTRPTW buscando minimizar el costo de la ruta utilizando un algoritmo de *branch-and-price* junto con una búsqueda de vecindario de amplia adaptabilidad. Por su parte [7] utiliza un algoritmo memético para resolver el TTRPTW con demandas estocásticas, una restricción más apegada a la realidad. La función a minimizar es la distancia total, añadiendo la consideración del costo de recurso en caso de falla y costo de recurso de clientes con ventanas de tiempo que no se visitaron. Así mismo, en [19] utilizan un enfoque híbrido basado en búsqueda local y búsqueda de vecindario local basado en estrategias de control de metaheurísticas estándar buscando minimizar la distancia total recorrida para TTRPTW y para el problema del camión y su remolque con transferencia de carga (TTRPLT, por sus siglas en inglés). Los autores concluyen que se puede personalizar un marco de referencia tipo metaheurístico para abordar variantes del TTRP. Cabe destacar que, al comparar los métodos de resolución de los artículos anteriormente mencionados, el algoritmo de Parragh y Cordeau [18] provee las mejores soluciones además de hacerlo en un tiempo de ejecución razonable para grandes instancias.

Al hablar del problema de TTRP sin ventanas de tiempo, existen diversos trabajos que lo abordan. El autor que introduce este término es Chao [1] en donde define al TTRP como una variante del VRP. El VRP busca minimizar la distancia de las rutas que se generan al tener un conjunto de clientes que demandan una cantidad de un producto con vehículos heterogéneos u homogéneos. Dentro del problema se tiene solo un depósito en donde los vehículos tienen que empezar y terminar su recorrido. Se busca minimizar la distancia recorrida por la flota, al atender la demanda de cada cliente, dando a cada vehículo una orden de visitas [1]. Para mejorar la representación de la realidad, Chao introduce la variante de que esa flota de vehículos este compuesta por camiones a los cuales se les pueda agregar un remolque. Cada camión y cada remolque tiene una capacidad determinada. Con esta restricción, dos tipos de clientes se definen, así como tres tipos de rutas. El primer tipo de cliente, al que llamaremos *cliente completo*, es aquel que puede ser visitado por un camión con su remolque. El segundo tipo, llamado *cliente camión*, es aquel que solo admite la visita del camión, esto se debe a una serie de restricciones que pueden existir tales como la incapacidad de realizar maniobras por parte del vehículo completo, es decir, el camión y su remolque, así como por restricciones de reglamentación de tránsito vehicular. Con estos tipos de clientes que se pueden encontrar, es posible tener tres tipos de rutas para atenderlos. La primera, una ruta compuesta por clientes completos y clientes camión, la cual puede ser atendida por un camión sin su remolque, es llamada una *ruta pura de camión*. La segunda, es aquella que comprende *clientes completos* solamente, y que debe ser atendida por vehículos completos llamada *ruta pura de vehículo*. La tercera es la *ruta completa de vehículo* compuesta por una ruta principal con *clientes completos* y una o más subrutas que contengan *clientes camión*. Chao concluye que su heurística propuesta puede resolver el TTRP de manera consistente, eficiente y eficaz.

### 3.2. Contribuciones

El presente trabajo incluye las siguientes aportaciones al estado del arte dadas por las restricciones particulares del dominio de aplicación.

- La incorporación de restricciones de coordinación de acceso a bahías
- Múltiples ventanas de tiempo para descarga de producto en las bahías, y para entrega a cliente.
- El trabajo con datos reales, posibilitando la evaluación del impacto en la empresa, tiendas, y en la sociedad que consume los productos.
- La formulación y resolución de nuevos modelos de optimización.
- El desarrollo de una extensión de la ALNS, adhoc al ETTRP con las características descritas.

La Tabla 1 hace una comparación de este trabajo con los trabajos de la literatura descritos en la Sección 3.1.

TABLA 1.  
**Comparativa del presente trabajo con trabajos de la literatura**

Año	Autores	Entrega al Cliente	Pedidos de Cliente	Horario de descarga/entrega	Duración de ruta	Coordinación de acceso a bahías
2013	Derigs et al.	1 solo tráiler	Un pedido	Uno de descarga Uno de entrega	<b>Limitada</b>	No
2015	Mirmohammadsadeghi & Ahmed	1 solo tráiler	Un pedido	Uno de descarga Uno de entrega	No limitada	No
2016	Grechikhin	<b>Varios tráileres</b>	<b>Varios Pedidos</b>	Uno de descarga Varios de entrega	<b>Limitada</b>	No
2017	Parragh & Cordeau	<b>Varios tráileres</b>	Un pedido	Uno de descarga Uno de entrega	No limitada	No
2018	Rothenbächer et al.	<b>Varios tráileres</b>	Un pedido	Uno de descarga Uno de entrega	<b>Limitada</b>	No
2020	Olmos	<b>Varios tráileres</b>	<b>Varios Pedidos</b>	<b>Varios de descarga Varios de entrega</b>	<b>Limitada</b>	<b>Si</b>

Como se puede observar en la Tabla 1, el problema del presente trabajo tiene 5 consideraciones, el uso de varios tráileres, la satisfacción de varios pedidos, varios horarios de descarga y entrega, una duración de ruta limitada y la coordinación de acceso a bahías. El trabajo de Mirmohammadsadeghi & Ahmed no comparten ninguna consideración con el problema que se aborda en este trabajo. Derigs et al. consideran solamente la duración de la ruta limitada. Grechikhin solo comparte las restricciones de los varios pedidos, varios tráileres y duración de ruta limitada. Rothenbächer et al. solo considera varios tráileres y duración de ruta limitada. Por último, Parragh & Cordeau comparte la restricción de varios tráileres y duración de ruta

limitada. En la literatura consultada, como se observa, no se encontró un trabajo relacionado que involucre todas las restricciones a considerar en el problema que se presenta en esta investigación.

## Capítulo IV

### Propuesta de solución

---

En este capítulo se describe la metodología empleada para desarrollar el presente trabajo, así como los pasos realizados en esta tesis. Se describe el desarrollo del prototipo, el modelo de optimización de AMPL y el desarrollo de la ALNS, así como los operadores de destrucción e inserción utilizados por esta metaheurística. También, se describen las instancias de prueba utilizadas, detallando el proceso de obtención de las mismas. Por último, se describe el contexto en donde se aplicó la metodología.

## 4.1. Metodología

Para el desarrollo del proyecto se decidió utilizar una metodología experimental debido a que Dodig-Crnkovic [20] declara que es más eficaz en problemas donde se intenta desarrollar herramientas para resolver problemas de optimización con restricciones. La metodología consiste en los siguientes pasos:

### 1. Modelación:

- Analizar y evaluar trabajos relacionados con metaheurísticas resolviendo problemas similares (ver Sección 3.1).
- Construir un modelo del proceso de distribución de las empresas haciendo la adquisición de los datos y el análisis del dominio de aplicación (ver Sección 4.2).

### 2. Comprensión de la naturaleza del problema del proceso de información:

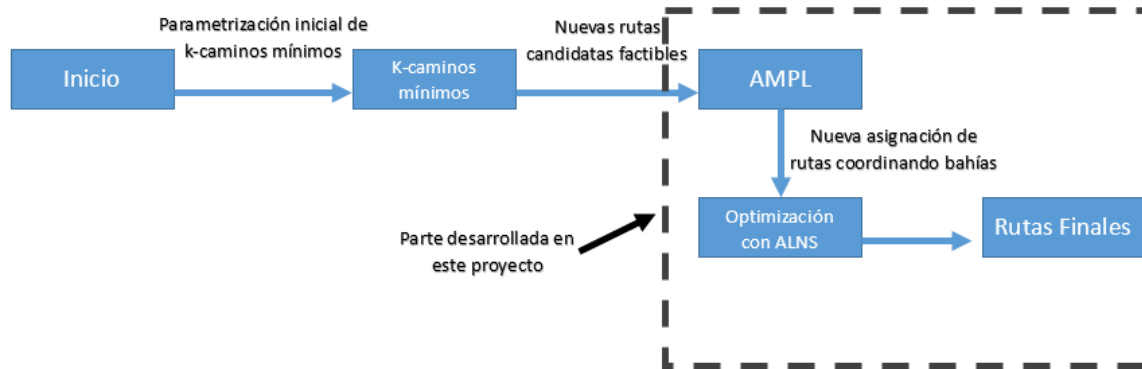
- Analizar el modelo realizado y elegir la metaheurística más adecuada de acuerdo a las restricciones, en este caso ALNS de acuerdo a Ropke & Pisinger [12](ver Sección 4.2.2).
- Desarrollar una metaheurística, en este caso ALNS, tomando en cuenta la capacidad de los vehículos y ventanas de tiempo utilizando un algoritmo de generación inicial de rutas aunado a un modelo matemático AMPL como proceso preliminar para obtener datos de entrada (ver Sección 4.2.2).
- Evaluar la solución final obtenida (ver Sección 5.4).

Al ser una metodología experimental se requiere un proceso iterativo en la implementación de la metaheurística desarrollada.

## 4.2. Desarrollo del prototipo

El prototipo desarrollado se trabajó en dos etapas. La primera etapa se utilizó para generar rutas iniciales utilizando el algoritmo de  $k$ -caminos mínimos. Dichas rutas iniciales se proyectan mediante un modelo AMPL del problema de ETTRP para obtener una configuración de asignación de los tipos de camiones a las rutas existentes utilizando un Lenguaje de Programación de Modelo Algebraico (AMPL, por sus siglas en inglés). Una vez teniendo la asignación, se toman esas rutas como solución inicial para la segunda etapa, que es la optimización de las mismas mediante el algoritmo de ALNS. La Figura 1 muestra el proceso descrito anteriormente.

FIGURA 1.

**Esquema general de la solución propuesta recalcando la aportación de la presente investigación****4.2.1. Modelo de optimización para AMPL**

Para modelar el problema de ETTRP buscando optimizar la ganancia, se desarrolló un sistema de ecuaciones, el cual es utilizado por AMPL y que se describe a continuación:

$$\max_{x,y,h} \sum_{v \in V} \sum_{r \in R^v} I_r x_r^v - \sum_{b \in B} f_b y_b \quad 4.1$$

sujeto a :

$$\sum_{r \in R^v} x_r^v \leq n^v, \forall v \in V \quad 4.2$$

$$\sum_{v \in V} \sum_{r \in R_o^v} x_r^v + h_o = 1, \forall o \in O \quad 4.3$$

$$x_r^v \leq y_b, \forall b \in B, v \in V, r \in R_b^v \quad 4.4$$

$$x_r^v \in \{0,1\}, \forall v \in V, r \in R^v \quad 4.5$$

$$y_b \in \{0,1\}, \forall b \in B \quad 4.6$$

$$h_o \in \{0,1\}, \forall o \in O \quad 4.7$$

Teniendo los conjuntos de datos siguientes:

**B** Conjunto de bahías

**O** Conjunto de ordenes a atender

**V** Conjunto de tipo de vehículos

**R** Conjunto de rutas factibles para entrega de mercancía a clientes

**R<sup>B</sup>** Conjunto de rutas que comparten alguna bahía

$R^v$  Conjunto de rutas factibles para entrega que utilizan tipo de vehículo  $v$

$R_o^v$  Conjunto de rutas factibles para entrega que utilizan vehículo  $v$  y atienden orden  $o$

$R_b^v$  Conjunto de rutas factibles para entrega que utilizan vehículo  $v$  y utilizan bahía  $b$

$V^r$  Conjunto de vehículos que pueden utilizarse en la ruta  $r$

El conjunto de parámetros siguiente:

$n^v$  Número de vehículos de tipo  $v$

$I_r$  Ganancia de la ruta  $r$

$f_b$  Costo de utilizar bahía  $b$

Y el conjunto de variables de decisión siguiente:

$x_r^v$  Indica si el vehículo  $v$  se utiliza en la ruta  $r$

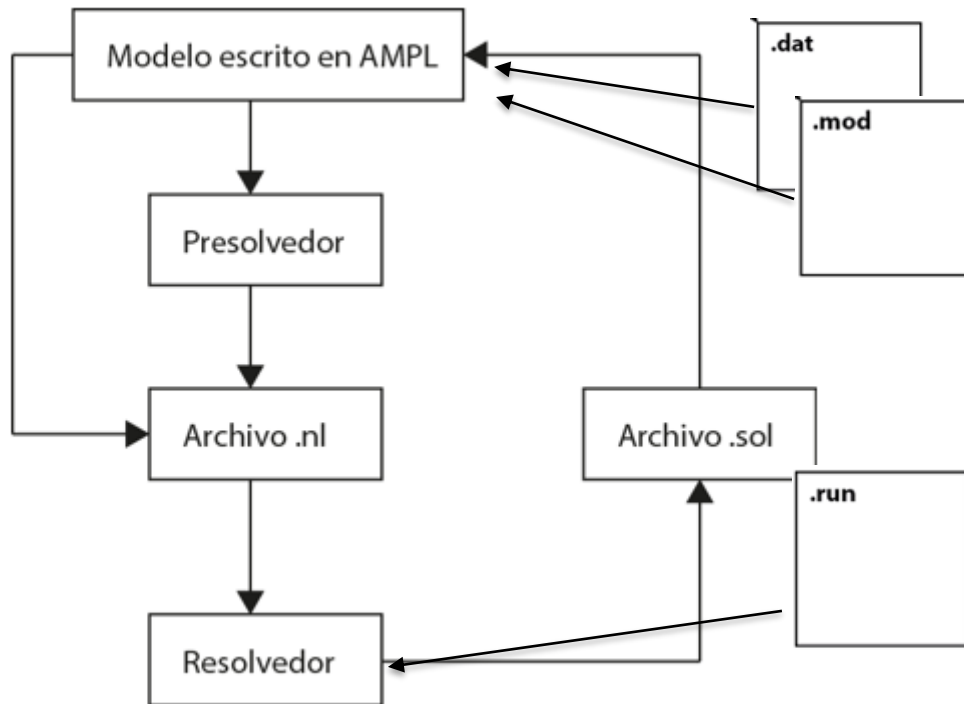
$y_b$  Indica si se utiliza la bahía  $b$

$h_o$  Indica si hay un vehículo que cumpla la orden  $o$

La función objetivo (4.1) busca maximizar la ganancia de la empresa en base a tres restricciones, esto es, el número de vehículos, ganancia de la ruta y el costo de uso de bahías. La restricción (4.2) establece un límite de vehículos en la solución a la cantidad que existen de cada tipo. La siguiente restricción (4.3) habilita la variable  $h_o$  si no existe un vehículo que satisfaga la orden  $o \in O$ . La restricción (4.4) establece la coordinación adecuada de las rutas a bahías sin superar el número de bahías existentes. El conjunto de restricciones 4.5-4.7 establece los requisitos binarios de las variables de decisión.

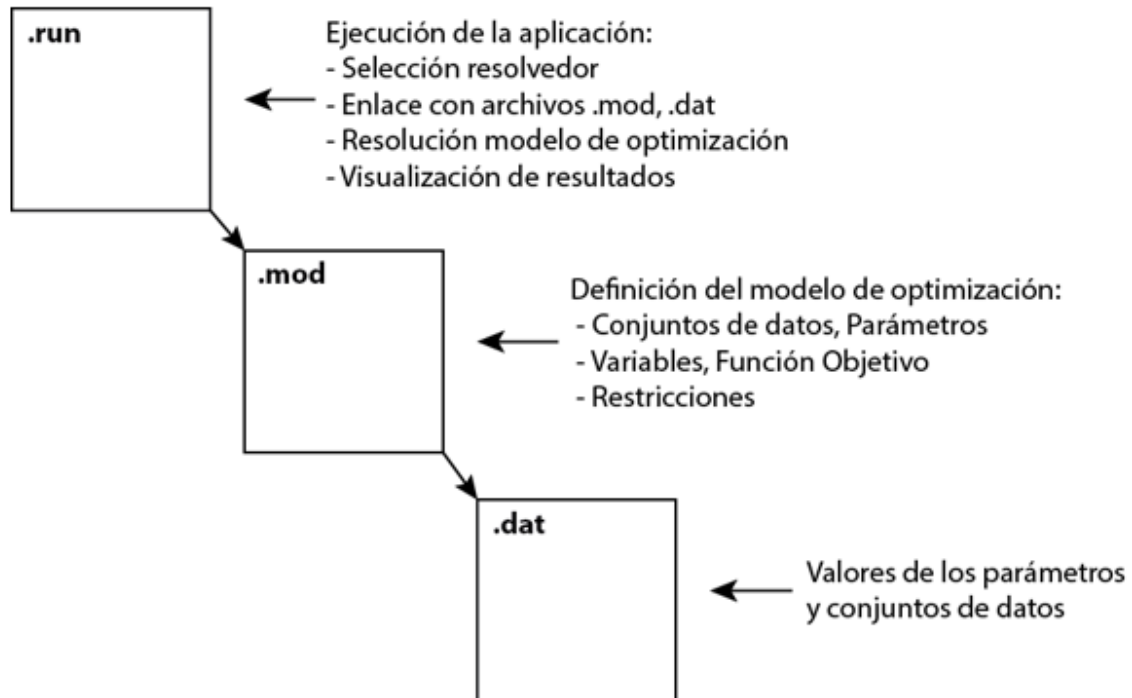
Se utilizó un lenguaje de programación algebraico para implementar el modelo de optimización descrito anteriormente. El modelo implementado pasa por un resolvidor previo (que llamaremos presolvidor) que genera un archivo con extensión “.nl” y, posteriormente, por un resolvidor el cual genera una solución representada en un archivo con extensión “.sol”. El archivo solución define la configuración de rutas, tipos de vehículos y bahías a utilizar para lograr la optimización de la función objetivo. La Figura 2 describe el proceso de la interacción AMPL-presolvidor-resolvidor.

FIGURA 2.  
**Interacción AMPL-presolvedor-resolvedor**



Existen tres pautas para la implementación en AMPL. La primera es la definición de un archivo “.run”, en la cual, se elige el resolvedor a utilizar (CPLEX en este caso), los archivos “.mod” y “.dat” y las sentencias para visualizar resultados tales como “solve”. La segunda es la declaración del archivo “.mod”, en el cual se establecen los diferentes conjuntos, parámetros y variables de decisión, así como la función objetivo y las restricciones. La tercera pauta es la definición de los valores de los datos declarados en el archivo “.mod”, los cuales se declaran en un archivo “.dat” tal como se muestra en la Figura 3.

FIGURA 3.  
Sistema de archivos de AMPL



La Figura 4 muestra el archivo “.run” a utilizar, mientras que la Figura 5 y la Figura 6 muestran los archivos “.mod” y “.dat”, respectivamente.

FIGURA 4.  
Archivo “.run”

```

1 option solver cplexamp;
2 # Asociamos AMPL con el resolvidor CPLEX.
3 model ECTRPTW.mod;
4 # Asociamos AMPL con el modelo de optimización.
5 data ECTRPTW.dat;
6 # Cargamos los datos de entrada necesarios para el modelo de optimización.
7
8 option cplex_options 'timing = 2';
9 param filename symbolic;
10
11 solve;
12 display _solve_elapsed_time, z2;
13

```

La primera línea de la Figura 4 representa el uso del resolvidor CPLEX, mientras que, en la tercera y quinta línea, se especifica el archivo .mod y el .dat, respectivamente. Las líneas 8 y 9 son configuraciones estándar para la ejecución. La línea 11 hace que se resuelva el programa, mientras que la línea 12 define las variables

a mostrar en la solución, en este caso, el tiempo de ejecución (`_solve_elapsed_time`), la función objetivo ( $z_2$ ), la variable binaria de apertura de bahía ( $y$ ) y la variable binaria que indica la asignación de un vehículo  $k$  a una ruta  $r$ .

FIGURA 5.  
Archivo “.mod”

```

1  # Declaración de conjuntos simples
2  set K; # Conjunto de tipo de vehículos
3  set C; # Conjunto de clientes
4  set B;
5
6  # Declaración de conjuntos complejos
7  set RK{K};
8  set RC{C};
9  set RB{B};
10 set RKC{k in K, c in C}:= setof{r1 in RK[k], r2 in RC[c]: r1 = r2} r1;
11 set RKB{k in K, b in B}:= setof{r1 in RK[k], r2 in RB[b]: r1 = r2} r1;
12 set R:= union {k in K} RK[k];
13
14 # Parámetros del modelo.
15 param n{K}; # Máximo de vehículos disponibles por cada tipo.
16 param f{B}; # Costo de apertura de cada bahía.
17 param ir{R}; # Ingreso por ruta
18
19 # Variables de decisión.
20 var y{b in B} binary;# Si la bahía b se abre o no
21 var x{k in K, r in RK[k]} binary;# Si el vehículo de tipo k se asigna o no a la ruta r
22 var h{C} binary;
23 # Función Objetivo.
24 maximize z2: sum{k in K, r in RK[k]} ir[r]*x[k,r] - sum{b in B} f[b]*y[b]
25
26 # Definición de las restricciones
27
28 # Restricciones de límites inferiores y superiores de frecuencias por tipo de línea
29 subject to VehKava{k in K}: sum{r in RK[k]} x[k,r] <= n[k];
30 subject to Sat1{c in C}: sum{k in K, r in RKC[k,c]} x[k,r] + h[c] = 1;
31 subject to rute2bay{k in K, b in B, r in RKB[k,b]}: x[k,r] <= y[b];

```

En la Figura 5 se muestra el archivo que sirve como modelo. De la línea 2 a la 4 se declaran los conjuntos simples en el programa, estos son: los tipos de vehículos  $K$ , los clientes  $C$  y las bahías  $B$ . Las líneas siguientes, desde la 7 hasta la 12, declaran los conjuntos complejos, estos son los que dependen de más de una variable, los cuales son: el conjunto de rutas asignadas por tipo de vehículo  $K$ ,  $RK(K)$ , el conjunto de rutas que atienden al cliente  $C$ ,  $RC(C)$ , el conjunto de rutas que utilizan la bahía  $B$ ,  $RB(B)$ , el conjunto de rutas que utilizan el vehículo tipo  $K$  y atienden al cliente  $C$ ,  $RKC(k \text{ en } K, c \text{ en } C)$ , el conjunto de rutas que utilizan el vehículo  $K$  y utilizan la bahía  $B$ ,  $RKB(k \text{ en } K, b \text{ en } B)$  y por último el conjunto de rutas  $R$ . Los parámetros a utilizar en el modelo se declaran de la línea 15 a la 17. Estos parámetros son: el máximo número de vehículos disponibles de cada tipo  $n(K)$ , el costo de apertura de cada bahía  $f(B)$  y el ingreso por cada ruta  $Ir(R)$ . En las

siguientes líneas (20 a 22) se declaran las variables de decisión que son: la apertura de una bahía,  $y(b \text{ en } B)$ , la asignación de un tipo de vehículo  $k$  a una ruta,  $x(k \text{ en } K \text{ r en } RK(k))$  y la atención a un cliente  $c$ ,  $h(c)$ . La línea 24 declara la función objetivo  $z_2$ , mientras que las últimas 3 líneas declaran las restricciones a las que está sujeto el modelo, estas son: el límite de vehículo,  $VehkAva$ , la atención a la mayor cantidad de clientes, satisfaciendo la demanda,  $Sat1$ , y el límite de bahías disponibles,  $rute2bay$ .

FIGURA 6.  
Archivo “.dat”

```

1 set K:= 1 2; #tipos de vehiculo
2 set C:= 1 2 3; #numero de clientes
3 set B:= 1 2 3; #bahias
4
5 set RK[1]:= 1 2 3; #rutas k que utiliza el vehiculo 1
6 set RK[2]:= 1 3; # rutas k que utiliza el vehiculo 2
7
8 set RC[1]:= 1; #rutas que reparte al cliente 1
9 set RC[2]:= 2; #rutas que reparte al cliente 2
10 set RC[3]:= 3; #rutas que reparte al cliente 3
11
12 set RB[1]:= 1; #rutas que utiliza la la bahia 1
13 set RB[2]:= 2; #rutas que utiliza la bahia 2
14 set RB[3]:= 3; #rutas que utiliza la bahia 3
15
16
17 param n:= #numero de vehiculos de tipo k
18 1 2
19 2 1;
20
21 param ir:= #ingreso neto dado por usar la ruta r
22 1 200
23 2 100
24 3 225;
25
26 param f:= #costo de puesta a punto de la bahia
27 1 10
28 2 20
29 3 15;

```

En la Figura 6 se muestra un ejemplo del archivo “.dat” para el programa de AMPL. En este archivo es donde se asigna el valor de los conjuntos y parámetros. De la línea 1 a la 14 se muestra el valor de los conjuntos, mientras que del 17 al 29 se muestra el valor de los parámetros.

El programa de AMPL genera una salida que indica el resultado de la función objetivo a optimizar, así como las bahías que se han utilizado y la asignación de los tipos de vehículo a cada una de las rutas seleccionadas. Dicha información se presenta en un archivo con extensión “.log” tal como se muestra en la Figura 7.

FIGURA 7.  
**Archivo solución “.log”**

```

ECTTRPTW.log - Notepad
File Edit Format View Help
Times (seconds):
Input = 0
Solve = 0
Output = 0
CPLEX 12.8.0.0: optimal integer solution; objective 480
2 MIP simplex iterations
0 branch-and-bound nodes
_solve_elapsed_time = 0.047
z2 = 480

y [*] :=
1 1
2 1
3 1
;

x :=
11 1
12 1
13 0
21 0
23 1

```

Con esta configuración de uso de bahías y asignación de tipos de vehículos a las rutas correspondientes.

#### 4.2.2. Representación de la solución

Para representar la solución generada por el ALNS se utilizaron arreglos del lenguaje JAVA. En la Tabla 2, se muestran los datos de entrada necesarios y el formato de representación para un correcto funcionamiento del algoritmo de ALNS. La primera columna indica el nombre del parámetro. La segunda columna indica el tipo de dato utilizado. La tercera columna muestra una descripción del parámetro, mientras que la cuarta columna muestra un ejemplo del parámetro en base al ejercicio de aplicación de la Figura 8. El sombreado representa que el parámetro es variable, esto es, que depende de la solución inicial. El renglón sin sombreado representa que el parámetro es fijo.

TABLA 2.  
Elementos de entrada del ALNS

Nombre	Tipo	Descripción	Ejemplo
Solución Inicial	String	Lista de rutas iniciales en la solución (en el ejemplo, se representa una solución con tres rutas)	((("d10", "b10", "c20", "b10", "b30", "c60", "b30", "b30", "c10", "b30", "d11"), ("d10", "b20", "c50", "b20", "b31", "c71", "b31", "b21", "c41", "b21", "d11"), ("d10", "b10", "c30", "b10", "d11"))
Probabilidad	int	Probabilidad de aceptación de una solución peor	0.005
Tasa de enfriamiento	int	Tasa de enfriamiento de temperatura de recocido simulado	0.99975
Factor de reacción	int	Factor de reacción para ayuda a selección de pares destrucción-reparación mejores.	0.1
Puntuación 1	int	Puntuación auxiliar 1 para evaluar la calidad de las soluciones	2
Puntuación 2	int	Puntuación auxiliar 2 para evaluar la calidad de las soluciones	1
Clientes a eliminar	int	Cantidad de clientes a eliminar de la solución inicial	2
Tiempo de Servicio	int	Tiempo de servicio: Minutos necesarios para servir a un cliente se basa en el arraylist de nodos	(0, 0, 0, 0, 0, 0, 0, 0, 20, 20, 30, 40, 40, 30, 20, 20, 40, 20)
Límite temporal inferior	int	Límite inferior de ventana de tiempo de cada elemento (expresado en minutos), se basa en el arraylist de nodos	(420, 420, 480, 960, 480, 960, 480, 960, 540, 1020, 540, 540, 1020, 1020, 540, 1020, 540, 1020)
Total de elementos	string	Lista total de elementos que se encuentran en la solución, se incluyen depósitos, bahías y clientes	("d10", "d11", "b10", "b11", "b20", "b21", "b30", "b31", "c10", "c11", "c20", "c30", "c31", "c41", "c50", "c51", "c60", "c71")
Costo bahía a bahía	int [][]	Costo en minutos basándose en distancia de bahía a bahía	[(0, 15, 15), (15, 0, 30), (15, 30, 0)]

Nombre	Tipo	Descripción	Ejemplo
Costo cliente a cliente	int [][]	Costo en minutos basándose en distancia de cliente a cliente	((0, 0, 0, 15, 15, 0, 0), (0, 0, 0, 15, 15, 15, 15).....)
Demanda	int	Cantidad de cajas que necesitan ser servidas por un cliente se basa en arreglo clien	(10, 15, 20, 10, 10, 20, 15)
Clientes	string	Clientes totales del problema, sin contar con identificador de mañana o tarde	("c1", "c2", "c3", "c4", "c5", "c6", "c7")
Atención de bahías a clientes	string	Lista de clientes atendido por cada bahía identificándolos por mañana y tarde se basa en arreglo bahías	((("c10", "c20", "c30"), ("c11", "c31"), ("c30", "c50"), ("c31", "c41", "c51"), ("c10", "c60"), ("c11", "c71"))
Total de bahías	string	Lista total de bahías en la solución con identificador de mañana o tarde	("b10", "b11", "b20", "b21", "b30", "b31")
Límite temporal superior	int	Límite superior en minutos de ventana de tiempo de cada elemento, se basa en el arraylist de nodos	(1260, 1260, 720, 1200, 720, 1200, 720, 1200, 660, 1140, 660, 660, 1140, 1140, 660, 1140, 660, 1140)
Cajas	int	Cantidad de cajas necesarias para cada elemento de arraylist nodos	(0, 0, -10, -10, -5, -5, -3, -3, 10, 10, 15, 20, 20, 15, 10, 10, 20, 15)
Capacidad de camión	int	Capacidad de camión por ruta, expresado en número de cajas, se basa en las rutas de la S	(55, 50, 60)

#### 4.2.3. Desarrollo de algoritmo Búsqueda Adaptativa de Vecindad Amplia

Se ha seleccionado ALNS debido a que Parragh et al. [18] muestra que su propuesta de ALNS provee las mejores soluciones para ETTRP en comparación a otras metaheurísticas utilizadas en la literatura. Se describe el proceso de la ALNS en el Algoritmo 1.

ALGORITMO 1.  
Metaheurística de ALNS

---

**Entrada:**  $S, q, p, r, c, \sigma_1$  y  $\sigma_2$

**Salida:**  $S_{mejor}$

---

**Inicio**

1.  $S_{mejor} = s, t = \frac{p}{\ln(0.5)}f(s), w_{dr} = 1, s_{dr} = 0, S_{dr} = 0$

2. **Repetir**

3. **Escoger** el par destrucción-reparación  $(d,r)$  cuyo peso  $w_{dr}$  sea  $\max \left\{ \frac{w_{dr}}{\sum_{d,r} w_{d,r}} \right\}$

4.  $s' =$  resultado de aplicar  $(d,r)$  con  $q = rnd(q_0, q_1)$  a  $s, S_{dr} = S_{dr} + 1$

5. **Si**  $f(s') < f(s_{mejor})$  **entonces**  $s_{mejor} = s', s = s', s_{dr} = s_{dr} + \sigma_1$

6. **Sino si**  $f(s') < f(s)$  **entonces**  $s = s', s_{dr} = s_{dr} + \sigma_2$

7. **Sino si**  $e^{-\frac{(f(s')-f(s))}{t}} \geq p$  **entonces**  $s = s', s_{dr} = s_{dr}$

8.  $t = t \cdot c, w_{dr} = w_{dr} \cdot (1 - r) + r \cdot \frac{s_{dr}}{S_{dr}}$

9. **Hasta** Criterio de Parada es alcanzado

10. **Devolver**  $S_{mejor}$

---

La línea 1 representa la inicialización de los siguientes parámetros:

1. Mejor solución  $S_{mejor}$ : Parámetro para guardar las mejores soluciones encontradas.
2. Temperatura  $t$ : Parámetro necesario para el funcionamiento del Recocido Simulado.
3. Peso  $w_{dr}$ : Parámetro utilizado para la selección de los operadores de destrucción-reparación.
4. Dos parámetros auxiliares  $s_{dr}$  y  $S_{dr}$ : utilizados para ponderar qué operadores brindan mejores resultados.

La línea 3 representa la selección de los operadores de destrucción-reparación tomando en cuenta el peso  $w_{dr}$  de cada par de operadores de destrucción-reparación. La línea 4 representa una solución en la cual ya se han eliminado e insertado una cantidad  $q$  de clientes. La línea 5 representa que se ha encontrado una solución global con un fitness mejor a la que se tenía ( $s_{mejor}$ ), llevando así a que se ajusten los parámetros de solución mejor y aquellos que llevan el rastreo de la eficiencia del par destrucción-reparación elegido ( $s_{dr}$  y  $S_{dr}$ ). La línea 6 representa que se ha encontrado una mejor solución a nivel local ( $s$ ) ajustando los parámetros correspondientes. La línea 7 representa la posibilidad de elección de una solución peor ( $p$ ), para ampliar la búsqueda de soluciones. La línea 8 representa la actualización de la temperatura del recocido simulado ( $t$ ) y de la puntuación del operador de destrucción-reparación seleccionado ( $w_{dr}$ ). La línea 9 evalúa los criterios de parada, esto es, el enfriamiento del recocido simulado o 5,000 iteraciones si no se encuentra una mejor solución [18].

#### 4.2.4. Operadores de destrucción e inserción

Para el desarrollo del algoritmo se utilizaron los cuatro operadores de destrucción básicos que son: eliminación aleatoria, eliminación del peor, eliminación de clústeres y eliminación relacional, y los dos de reparación propuestos inicialmente por Ropke & Pisinger [12], esto es, inserción voraz e inserción  $k$ -arrepentimientos, que sirvieron como base para el algoritmo de ALNS. Por lo anterior, la variabilidad y la calidad de la solución obtenida depende del par de operadores de destrucción-reparación a utilizar (véase línea 3, 5, 6, 7 y 8 del Algoritmo 1). El primer operador de destrucción es la eliminación aleatoria cuyo pseudocódigo se muestra en el Algoritmo 2.

#### ALGORITMO 2. Eliminación aleatoria

---

**Entrada:**  $s, q$   
**Salida:**  $s_{reducida}$

---

Hacer:

1.  $L =$  Todo cliente  $i$  en  $s$
  2. **Mientras**  $q > 0$  **hacer**
  3.      $y = \text{rand}(0, 1]$
  4.     **Escoger** cliente  $i = L(y|L)$
  5.     **Eliminar** cliente  $i$  de  $s$  y de  $L$
  6.      $q = q - 1$
  7. **Fin Mientras**
- 

En este operador de destrucción, se genera una lista de todos los clientes en la solución, tal como se muestra en la línea 1 del Algoritmo 2. De la línea 2 a la línea 7 se usa una sentencia *while* para iterar el procedimiento en base a la cantidad  $q$  de clientes a eliminar. Se genera un número aleatorio para saber qué cliente eliminamos de la lista  $L$ , se elimina y se decrementa el contador  $q$ . Siguiendo el ejemplo que se ilustra anteriormente se puede ver en la Figura 8 el resultado del procedimiento de este operador de destrucción aleatoria.

FIGURA 8.  
Ejemplo operador de eliminación aleatoria

```

Lista de clientes = [c41, c10, c50, c71, c30, c20]
nodo = 2
nodo 2 string c50
Lista de clientes = [c41, c10, c71, c30, c20]
nodo = 4
nodo 4 string c20
ClientesEliminados aleatoria normal= [c50, c20]

```

El segundo operador de destrucción es la eliminación del peor, el Algoritmo 3 muestra el pseudocódigo utilizado.

ALGORITMO 3.  
Eliminación del peor

---

**Entrada:**  $s, q, v_2 > 1$

**Salida:**  $s_{reducida}$

---

Hacer:

1. **Mientras**  $q > 0$  **hacer**
  2.      $L =$  Todo cliente  $i$  en  $s$  ordenado por  $c_{i,s} = f(s) - f_{-i}(s)$  descendiente
  3.      $y = \text{rand}(0, 1)$
  4.     **Escoger** cliente  $i = L(y^{v_2} | L|)$
  5.     **Eliminar** cliente  $i$  de  $s$  y de  $L$
  6.      $q = q - 1$
  7. **Fin Mientras**
- 

En la eliminación del peor, se genera una lista  $L$  ordenada de todos los clientes de la solución, tal como se muestra en la línea 2 del Algoritmo 3. El ordenamiento se basa en el costo,  $c_{i,s}$  de la diferencia en fitness de la solución mejor menos la solución sin el cliente  $i$  y es de carácter descendiente. Después, se genera un número aleatorio el cual se ve afectado por el parámetro  $v_2$  que determina el grado de aleatoriedad para la selección del cliente, priorizando los clientes peores de acuerdo al ordenamiento, esto se muestra en la línea 3 y 4, respectivamente. Posteriormente, se elimina el cliente y se decrementa el contador  $q$ , tal como se muestra en la línea 5 y 6. La Figura 9 muestra el ejemplo del procedimiento antes descrito.

FIGURA 9.  
Ejemplo operador de eliminación del peor

```

lista de cliente del peor al mejor = [c41, c10, c50, c71, c30, c20]
nodo a eliminar 0 string c41
lista de cliente del peor al mejor = [c71, c10, c50, c30, c20]
nodo a eliminar 0 string c71
ClientesEliminados peor normal = [c41, c71]

```

El tercer operador de destrucción es la eliminación de clústeres. El Algoritmo 4 muestra el pseudocódigo.

ALGORITMO 4.  
Eliminación de clústeres

---

**Entrada:**  $s, q, \theta_1, T, c_{ij}, qc$

**Salida:**  $S_{reducida}$

---

Hacer:

1. **Mientras**  $q > 0$  **hacer**
  2.     **Si** Es la primera vez **entonces**
  3.         **Escoger**  $r^* = \text{rand}(1, |R|)$
  4.     **Sino**
  5.         **Escoger**  $i^* = \text{rand}(i \text{ en } Clu^*)$
  6.         • **Escoger**  $r^*$  donde un  $i(r^*)$  tiene el  $R_{i^*,i(r^*)} = \theta_1|T_{i^*} - T_{i(r^*)}| + \theta_2c_{i^*,i(r^*)} + \theta_3|qc_{i^*} - qc_{i(r^*)}|$
  7.         más bajo
  8.     **Fin Si**
  9.     **Dividir**  $i$  de  $r^*$  en  $Clu1, Clu2$  por similitud de  $R_{i1i2}$
  10.     **Escoger**  $Clu^* = \text{rand}(Clu1, Clu2)$  aleatoriamente
  11.     **Eliminar** todo  $i$  de  $r$  contenido en  $Clu^*$
  12.      $q = q - \# \text{ clientes en } Clu^*$
  13. **Fin Mientras**
- 

Este operador es más complejo, ya que supone eliminar clientes que pertenecen a un grupo de clientes (clúster) basados en características similares, las cuales se representan con la medida de similitud,  $R_{ij}$ , descrita en la última viñeta del Algoritmo 4. Esta medida de similitud considera tres factores para agrupar a los clientes: el tiempo de servicio para atención a clientes, el costo de transporte, basado en tiempo, entre cliente  $i$  y  $j$ , y la cantidad de cajas solicitadas por los clientes  $i$  y  $j$ . Para lograr esto, en la primera iteración, se escoge una ruta al azar y se dividen los clientes de acuerdo a esta medida de similitud en  $Clu1$  y  $Clu2$ , tal como se muestra en la línea 3 y 8 del Algoritmo 4, respectivamente. Una vez divididos, se escoge de manera

aleatoria uno de esos clústeres, representado como  $Clu^*$ , y se eliminan los clientes del clúster seleccionado, teniendo en cuenta el valor  $q$  del total a clientes a eliminar, esto se muestra en las líneas 9 y 10, respectivamente. Para las posteriores iteraciones, se selecciona un cliente  $i^*$  de  $Clu^*$ , y se calcula la medida de similitud con los clientes de las rutas restantes, seleccionando la ruta que tenga un cliente con la medida de similitud más baja, tal como se muestra en las líneas 5 y 6 y se realiza el mismo procedimiento de división de los clientes como en la primera iteración, decrementando la cantidad de clientes al contador  $q$ , tal como se muestra en la línea 11. La Figura 10 ilustra un ejemplo de esta eliminación con el ejercicio que sea ha estado desarrollando.

FIGURA 10.  
**Ejemplo operador de eliminación por clúster**

```
cluster1 = [c20, c30]  
ClientesEliminados cluster normal = [c20, c30]
```

Se puede observar en el Figura 10, que el cliente “c2” y el cliente “c3” se agruparon en el clúster 1 debido a las similitudes de tiempo de servicio, cercanía y demanda de cajas que presentan.

El último operador de eliminación es el relacional. El Algoritmo 5 muestra el pseudocódigo. Este operador utiliza la misma medida de similitud  $R_{ij}$  descrita en la eliminación de clúster. Sin embargo, la medida es utilizada para ordenar la lista de clientes de manera ascendente de acuerdo al valor del  $R_{ij}$ , lo cual representa que existe una mayor relación entre clientes, tal como se muestra en la línea 6 del Algoritmo 5. Una vez ordenada la lista de clientes, se determina aleatoriamente el cliente a eliminar, tal como se muestra en las líneas 6, 7, 8 y 10. Se debe resaltar que el primer cliente a eliminar se selecciona aleatoriamente, y con este cliente se van construyendo las medidas de similitud con los demás clientes, tal como se muestra en las líneas 1, 2 y 10, respectivamente. La Figura 11 ilustra un ejemplo de esta eliminación.

FIGURA 11.  
Ejemplo operador de eliminación relacional

```
seleccionar relacional voraz
ClientesEliminados relacional normal = [c20, c360]
```

ALGORITMO 5.  
Eliminación Relacional

---

**Entrada:**  $s, q, \theta_1, T, c_{ij}, qc, vI > 1$

**Salida:**  $S_{reducida}$

---

Hacer:

1. **Escoger** un cliente  $i$  de forma aleatoria de  $s$
  2.  $D = \{i\}$
  3. **Mientras**  $|D| < q$  **hacer**
  4.     **Escoger** un cliente  $d$  de forma aleatoria de  $D$
  5.      $L =$  Todo cliente  $i$  en  $s$  no contenido en  $D$
  6.     **Ordenar**  $L$  de tal modo que  $i < j$  implica que  $R_{d,i} < R_{d,j}$
  7.      $y = \text{rand}(0, 1)$
  8.      $D = D \cup \{L(y^{v_1} | L)\}$
  9.     **Fin Mientras**
  10. **Eliminar** los clientes  $i$  en  $D$  de  $s$
- 

En cuanto a los operadores de reparación, el operador de inserción voraz es el primero que se describe. El Algoritmo 6 muestra su pseudocódigo.

ALGORITMO 6.  
Inserción voraz

---

**Entrada:**  $U, R$

**Salida:**  $S_{ampliada}$

---

Hacer:

1. **Mientras**  $|U| > 0$  **hacer**
  2.     **Calcular**  $\Delta f_{i,r}$  (incremento del costo de la ruta  $r$  tras reinsertar el cliente  $i$  en la mejor posición,  $\Delta f_{i,r} = \infty$ , para toda ruta  $r$  donde no sea factible insertar el cliente  $i$ ) para todo cliente  $i \in U$  y ruta  $r \in R$
  3.     **Calcular**  $c_i = \min_{r \in R} \{\Delta f_{i,r}\}$  para todo cliente  $i \in U$
  4.     **Escoger** el cliente  $i^* = \min_{i \in U} \{c_i\}$
  5.      $U = U - \{i^*\}$
  6.     **Fin Mientras**
-

La inserción voraz reinserta los clientes en base al costo que se genera al hacer la reinsertación de los clientes en las diferentes posiciones posibles en las rutas, tal como se muestra en las líneas 2, 3 y 4 del Algoritmo 6, seleccionando con mayor prioridad el cliente que genera el menor costo en la solución, tal como se muestra en la línea 4. La Figura 12 ilustra el ejemplo de esta inserción.

FIGURA 12.  
**Ejemplo operador de inserción voraz**

```

Cliente a Insertar c390
ruta [d10, b10, c390, b10, b10, c360, b10, b20, c250, b20, b30, c270, b30, b40, c180, b40, b40, c20, b40, b50, c40, b50, d11]
ganancia 8

```

El segundo operador de reparación es el de  $k$ -arrepentimientos. El Algoritmo 7 muestra su pseudocódigo.

ALGORITMO 7.  
**Inserción  $K$ -Arrepentimientos**

---

**Entrada:**  $U, R$

**Salida:**  $S_{ampliada}$

---

Hacer:

1. **Mientras**  $|U| > 0$  **hacer**
  2.     **Calcular**  $\Delta f_{i,r}$  para todo cliente  $i \in U$  y ruta  $r \in R$
  3.     **Ordenar**  $\Delta f_{i,r}$  de forma ascendente
  4.     **Escoger** el cliente  $i^* = \min_{i \in U} \{\sum_{j=1}^k (\Delta f_{i,k} - \Delta f_{i,1})\}$
  5.      $U = U - \{i^*\}$
  6. **Fin Mientras**
- 

En este operador se prioriza insertar al cliente cuya diferencia de costo de inserción entre las distintas rutas de la solución sea la menor, tal como se muestra en las líneas 2 y 3 del Algoritmo 7, respectivamente, es decir, se selecciona al cliente que genere menor diferencia de costos en el marco general de las distintas rutas de la solución, tal como se muestra en la línea 4. Al final se elimina el cliente seleccionado de la lista de clientes a insertar, tal como se muestra en la línea 5. La Figura 13 ilustra el ejemplo de esta inserción.

FIGURA 13.  
Ejemplo operador de inserción voraz

```

Clientes acomodados k arrepentimientos
[c180, c360]

Cliente a Insertar c180
ruta [d10, b40, c180, b40, b10, c360, b10, b10, c220, b10, b10, c390, b10, b30, c270, b30, b40, c20, b40, b50, c40, b50, d11]
ganancia 11

Cliente a Insertar c360
ruta [d10, b10, c360, b10, b10, c220, b10, b10, c390, b10, b20, c250, b20, b30, c270, b30, b40, c20, b40, b50, c40, b50, d11]
ganancia 8

```

El algoritmo de ALNS necesita seis datos de entrada para funcionar correctamente. El primero, es la solución inicial,  $S_i$ , que se pretende optimizar. Dicha solución inicial, se compone por las rutas generadas por el preprocesamiento del algoritmo de  $k$ -caminos mínimos y AMPL descrito en la Sección 4.2.1. El segundo, es el número de clientes para eliminar y reinsertar representado por  $q$ . El tercero, una probabilidad  $p$  de aceptación de una solución peor. El cuarto, una tasa de enfriamiento de la temperatura  $c$  ya que se utiliza recocido simulado. El quinto, un factor de reacción  $r$  que servirá para la actualización de los pesos para seleccionar los mejores pares de destrucción-reparación. El sexto, un par de puntuaciones,  $\sigma_1$  y  $\sigma_2$ , que servirán para medir la calidad de las soluciones. El algoritmo de ALNS toma como solución inicial la proporcionada por el optimizador de AMPL, mientras que los otros cinco datos son asignados de acuerdo con lo establecido por la literatura de la ALNS, los cuales se definen a continuación [18] [12]:

- a.  $q = \text{Aleatorio}(q_0, q_1)$
- b.  $p = 0.005$
- c.  $c = 0.99975$
- d.  $r = 0.1$
- e.  $\sigma_1 = 2$  y  $\sigma_2 = 1$

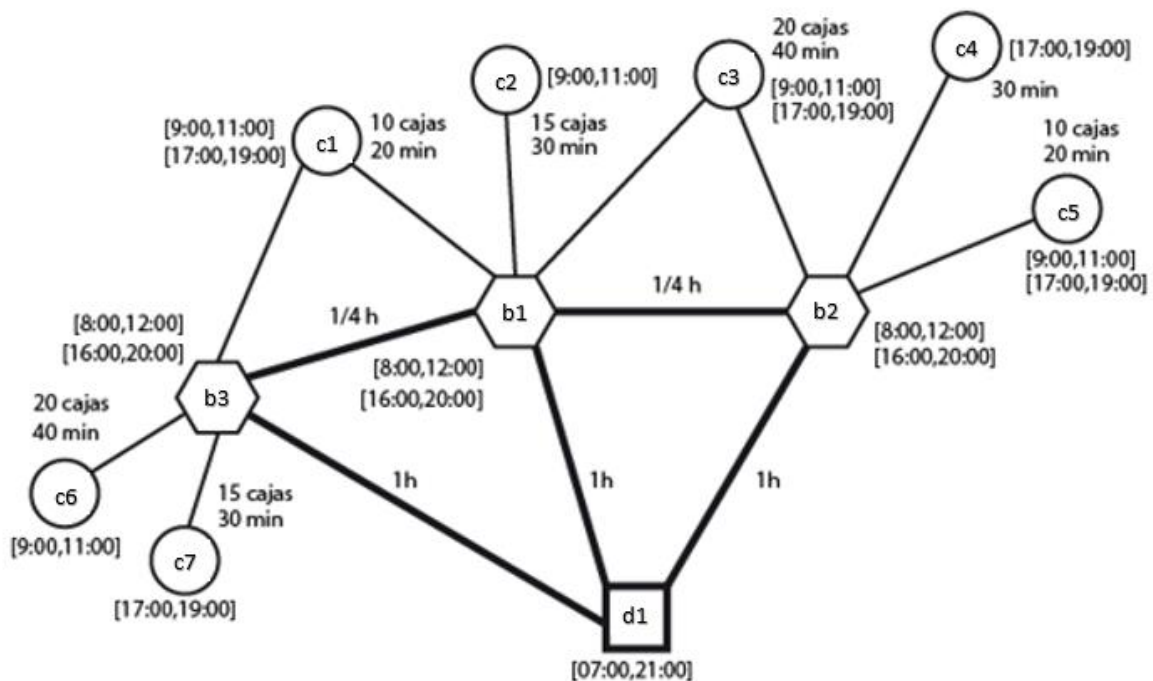
El proceso del algoritmo de ALNS se describe en los siguientes puntos, en donde se tomó como ejemplo el problema descrito en la Figura 14. Los clientes se representan con círculos y la literal “c” seguida de un indicador numérico. Una leyenda a lado de cada identificador de los clientes expresa el número de cajas solicitadas y el tiempo de servicio necesario para surtir las. Las bahías se representan con hexágonos con una literal “b” y un indicador numérico. El depósito se representa con un cuadrado con su respectiva literal “d” e indicador numérico. Los arcos remarcados en negro son conexiones entre depósitos y bahías de estacionamiento, con un indicativo del tiempo de desplazamiento del camión representado en horas entre las dos. Los arcos normales representan la conexión existente entre bahías y clientes, con un indicativo de

tiempo de desplazamiento a pie expresado en horas. Los números entre corchetes representan la ventana horaria de cada nodo por mañana o tarde si existe lo cual en la Tabla 2 se representa con un “0” y “1” al final del identificador de cliente respectivamente.

Los datos con los que se abordó este problema de ejemplo son los siguientes:

1. Número de camiones: En este caso se tienen cuatro camiones con capacidad heterogénea.
2. Capacidad de camiones: Para el camión tipo 1 la capacidad es de 55 cajas, para el camión de tipo 2 la capacidad es de 50 cajas y para el camión de tipo 3 la capacidad es de 60 cajas.
3. Costo de apertura de bahías: Existen tres bahías en el ejercicio de aplicación. La bahía “b1” tiene un costo de uso de -10 cajas, mientras que las bahías “b2” y “b3” tienen un costo de -5 y -3 cajas respectivamente. El costo se representa en base al precio por caja que se tiene. Es decir, el costo de -10 representa un gasto correspondiente a 10 cajas.

FIGURA 14.  
Grafo representando un caso ilustrativo del ETTRP



Una vez teniendo los datos de entrada necesarios, explicados en la Sección 4.2.2, el proceso es el siguiente:

1. Se evalúa el fitness de la solución inicial, el cual se toma como solución mejor global y local, y se selecciona al par de operadores de destrucción-reparación que tenga mayor peso  $w_{dr}$ . La selección se lleva a cabo de manera aleatoria cuando el peso es el mismo en todos los pares. A continuación,

la Figura 9 muestra un ejemplo con una solución inicial compuesta por tres rutas, aquellas dentro del recuadro azul de la Figura 15 y que atiende a siete clientes. La Tabla 3 muestra la correspondencia de los pesos con el par de operadores.

TABLA 3.  
Pesos asignados al par de operadores de destrucción-reparación

Variable	Par de operadores
W0	Aleatoria-Voraz
W1	Peor-Voraz
W2	Clúster-Voraz
W3	Relacional-Voraz
W4	Aleatoria-K Arrepentimientos
W5	Peor-K Arrepentimientos
W6	Clúster-K Arrepentimientos
W7	Relacional-K Arrepentimientos

FIGURA 15.  
Ejemplo de prueba ALNS con solución inicial, ganancia, pesos de selección y operador de destrucción seleccionado

```
ganancia de solución inicial
[[[d10, b10, c20, b10, b30, c60, b30, b30, c10, b30, d11],
[d10, b20, c50, b20, b31, c71, b31, b21, c41, b21, d11],
[d10, b10, c30, b10, d11]] = 69
W0 = 0.125 W1 = 0.125 W2 = 0.125 W3 = 0.125 W4 = 0.125 W5 = 0.125 W6 = 0.125 W7 = 0.125
seleccionar peor voraz
```

- Una vez seleccionado el par de destrucción-reparación, se utiliza para generar la nueva solución, en este ejemplo, se seleccionó la eliminación del peor con inserción voraz representado por el W1, por lo que el algoritmo realiza todo el procedimiento de este operador de destrucción. La Figura 16 muestra los clientes elegidos para ser eliminados de acuerdo al operador de destrucción del peor, procedimiento detallado en la Sección 4.2.4.

FIGURA 16.

**Ejemplo de prueba ALNS con clientes eliminados utilizando el par peor-voraz**

```
clienteseliminado = [c30, c41]
```

3. Posteriormente, la nueva solución se compara con la mejor solución, tanto global como local, y, si logra mejorar el fitness, se establece como mejor solución global y local, actualizando los parámetros para recompensar al par seleccionado tal como se muestra en la línea 8 del Algoritmo 1, este caso se presenta en el ejemplo de la Figura 17. Al encontrar una mejor solución, el peso de este par se ve recompensado para que tenga mayor probabilidad de ser seleccionado en futuras iteraciones. Si no se mejora la solución el par utilizado se ve castigado, tal como se muestra en la misma línea 8 del Algoritmo 1, lo cual le genera menores probabilidades para ser seleccionado en las siguientes iteraciones. Sin embargo, la probabilidad  $p$  permite seleccionar una solución peor, para así evitar caer en óptimos locales.

FIGURA 17.

**Ejemplo de prueba ALNS con solución parcial, fitness y actualización de pesos de selección**

```
W0 = 0.10989010989010989 W1 = 0.23076923076923078 W2 = 0.10989010989010989 W3 = 0.10989010989010989  
W4 = 0.10989010989010989 W5 = 0.10989010989010989 W6 = 0.10989010989010989 W7 = 0.10989010989010989
```

4. Una vez terminado el proceso de destrucción-inserción, se actualiza la temperatura, tal como se muestra en el Algoritmo 1, y continúan las iteraciones del programa de ALNS, teniendo como condición de parada 5,000 iteraciones, recalando que cada 100 iteraciones los parámetros se reinician para evitar caer en soluciones óptimas locales. En la Figura 18 que describe el ejemplo se puede ver que se encontró una mejor solución a la inicial que genera una ganancia del valor de 79 cajas.

FIGURA 18.  
**Ejemplo de prueba ALNS con solución final y ganancia**

```
Smejor despues de 100 iteraciones =  
[[d10, b20, c30, b20, b20, c50, b20, b30, c60, b30, b30, c10, b30, d11],  
[d10, b10, c20, b10, b31, c71, b31, d11],  
[d10, b21, c41, b21, d11]] con ganancia = 79
```

Como se muestra en la Figura 18 la estructura de datos de la solución final es de un *ArrayList*, mostrando el valor de la ganancia. El *ArrayList* muestra una sucesión de nodos, los cuales pueden ser de tres tipos: depósito, bahía o cliente. Cada nodo contiene una literal que representa el tipo de nodo, *d* representa depósito, *b* representa bahía y *c* representa cliente. Seguido de la literal, se encuentran dos números, el primero sirve como identificador de cada nodo y el segundo representa la ventana horaria del mismo, teniendo un “0” para la ventana horaria del nodo por la mañana y un “1” para la ventana horaria por la tarde. La Figura 18 representa la estructura descrita anteriormente.

### 4.3. Instancias de prueba

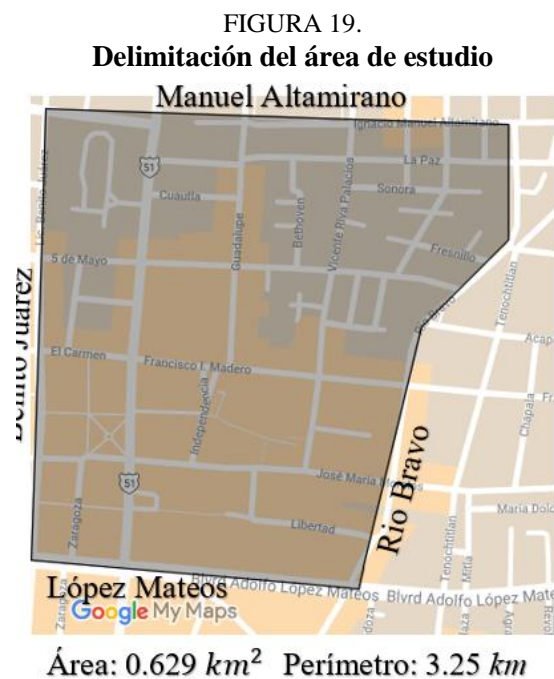
Para la adquisición de datos se llevó a cabo una estancia de investigación en el Tecnológico Nacional de México campus Celaya, en donde se realizaron diferentes actividades. El proceso de generación de rutas, optimización por AMPL y optimización por ALNS descrita en la Sección 4.2 se basó en datos obtenidos del centro histórico de la ciudad de Celaya en donde cuatro compañías dedicadas a la producción y distribución de mercaderías necesitan generar rutas de reparto para satisfacer las necesidades de sus clientes. Gracias a la naturaleza del problema y, para una correcta generación de instancias de prueba, fue necesaria la recolección de información de proveedores y clientes, tomando mayor atención a sus ventanas horarias, demanda de producto, tiempos de servicio y localización. Se realizaron dos actividades en el transcurso de la estancia de investigación, las cuales se describen a continuación:

1. Determinación del área de estudio, tomando en cuenta las características de la extensión del TTRP (ETTRP) a trabajar (Sección 4.3.1).
  - 1.1. Identificación de proveedores y de bahías de estacionamiento para la descarga de mercancías (Sección 4.3.2).
  - 1.2. Identificación de clientes a quienes se atienden a través de las bahías de estacionamiento (Sección 4.3.3).

2. Cálculo de distancias y estimación de tiempos de servicio a clientes desde las bahías. Para el cálculo de distancias se utilizó Google Maps debido a que toma en cuenta el sentido de las calles, lo cual es una consideración importante al analizar el recorrido de los camiones de reparto. Para la estimación de tiempos de servicio, se procedió tomando en cuenta la demanda de cajas del cliente y el recorrido que debe tomar el repartidor desde la bahía hasta la ubicación del cliente, tomando el promedio de las posibles bahías desde las cuales se puede atender. Se utilizó el estándar promedio de velocidad de 5 km/h de la caminata del repartidor, así como un total de 10 segundos de servicio por caja surtida a los clientes de acuerdo a la experiencia de repartidores. Así mismo, el número de viajes de ida y vuelta que debe realizar el repartidor se estimó tomando en cuenta 5 cajas por cada viaje de acuerdo a lo observado en la estancia de investigación.

#### 4.3.1. Identificación del área de estudio

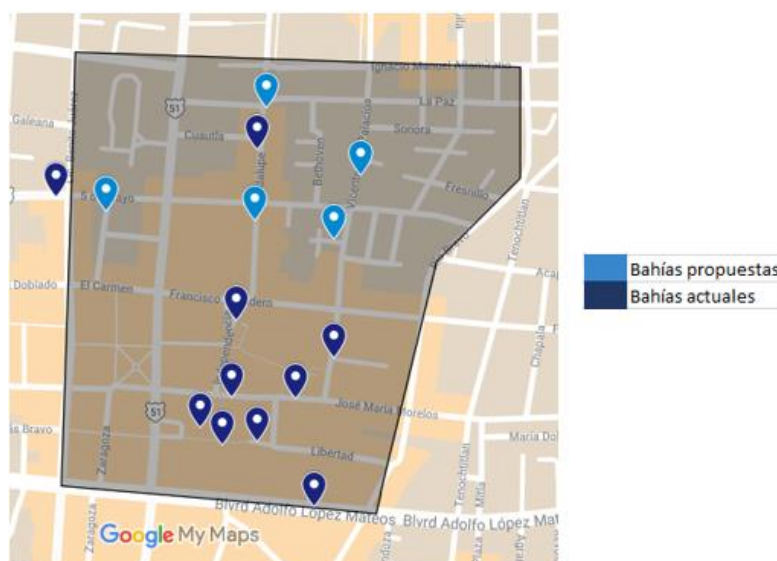
El primer paso fue delimitar el área de estudio en el centro histórico de Celaya tomando en cuenta el ancho de las calles, ya que es un factor que genera las condiciones necesarias para el problema de ETTRP a trabajar, debido a la incapacidad que tienen los camiones de entrega para estacionarse frente a los clientes para realizar la entrega de mercancías. La Figura 19 muestra los límites del área de estudio con una extensión superficial de  $0.629 \text{ km}^2$  y un perímetro de 3.25 km.



#### 4.3.2. Identificación de proveedores y de bahías de estacionamiento

Una vez delimitada el área de estudio, se identificaron las bahías de estacionamiento utilizadas por las empresas para hacer la descarga de sus mercancías. Se entiende como bahía de estacionamiento a aquellos lugares en donde los camiones de reparto se estacionan para repartir a sus clientes. Se tomaron en cuenta todas las bahías dentro del área de estudio y algunas bahías que se encontraban fuera de los límites del área de estudio, pero a una cercanía menor a 200 m. de los mismos, las cuales son utilizadas actualmente por las empresas dando un total de 15 bahías de estacionamiento. Posteriormente, con el análisis de clientes terminado, se identificaron las localizaciones de posibles ubicaciones para bahías y se propusieron un total de cinco bahías adicionales para una distribución más eficiente debido al tráfico vehicular y sentido de la vía, así como en la cercanía a los clientes a atender. Se consideraron restricciones tales como el sentido y anchura de las calles, así como a la cercanía con los clientes. La Figura 20 muestra la ubicación de las 10 bahías utilizadas actualmente en color azul marino, mientras que las bahías propuestas se resaltan de color azul cielo. Estas cinco bahías propuestas se eligieron en base a tres criterios: el primero es que las calles al menos deberían tener dos carriles. El segundo es la longitud del espacio del estacionamiento para que el camión pudiera estar estacionado y el tercero es la de tomar en cuenta la distancia de seguridad entre el lugar donde se estaciona el camión y la esquina de la calle. Estas cinco bahías son las que cumplieron con estos criterios dentro del área delimitada. Sin las bahías propuestas, existía un retraso de tiempo en la entrega de los pedidos de los clientes, por lo que no se cubría la totalidad de dichos pedidos.

FIGURA 20.  
Ubicación de bahías de estacionamiento

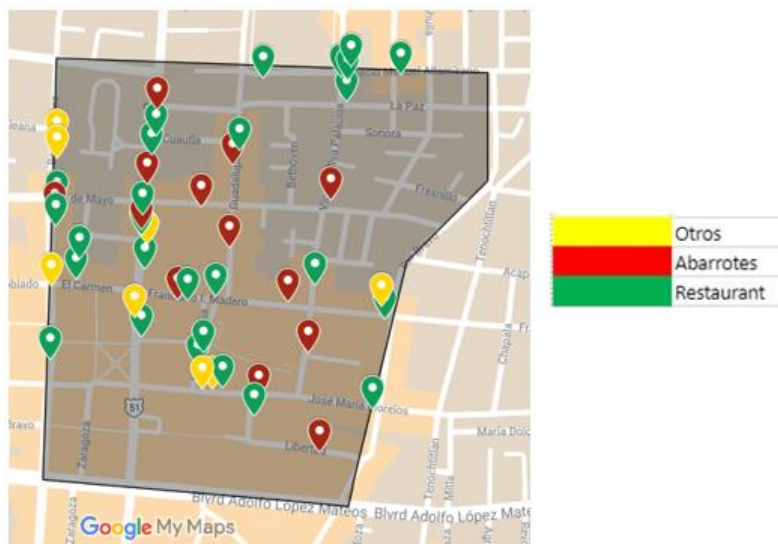


### 4.3.3. Identificación de clientes

En esta etapa del proceso, se identificaron todos los clientes que solicitan los productos de las cuatro empresas seleccionadas dando un total de 48. Se utilizaron estas cuatro empresas debido a que son las que reparten los productos con más movimiento de los clientes identificados.

Al clasificar a los clientes se identificaron tres categorías: restaurantes, tienda de abarrotes y otros (farmacias, papelerías, dulcerías, etc.). La Figura 21 muestra la ubicación de los clientes identificados, clasificados por colores. El color rojo representa a tiendas de abarrotes, el color verde representa a restaurantes y el color amarillo a la categoría de otros. Se realizó un análisis del porcentaje de clientes en cada categoría teniendo como presencia un 56.3% de restaurantes, 27.1% de tiendas de abarrotes y un 16.7% de otros comercios.

FIGURA 21.  
Ubicación de clientes y su clasificación



Con esta información se analizó el porcentaje de cobertura de las bahías actuales y propuestas a distancias de 50 y 100 m, tomando éstas como ideales para un servicio eficiente, esto es, que tarde menos de 10 minutos (tomando en cuenta un promedio de 13 cajas por pedido por cliente). La Tabla 4 muestra que el incremento de cobertura con las bahías propuestas a 50 m y a 100 m es de un 4% y de un 7% respectivamente. La Figura 22 (a) y la Figura 22 (b) representan una cobertura de 50 m con las bahías actuales y con las propuestas, respectivamente. La Figura 22 (c) y la Figura 22 (d) representan una cobertura de 100 m con bahías actuales y propuestas, respectivamente.

TABLA 4.  
Análisis de porcentaje de cobertura

Situación	Total Clientes	Clientes Cubiertos	Porcentaje
<b>50 m actuales</b>	48	14	29%
<b>50 m propuestas</b>	48	16	33%
<b>100 m actuales</b>	48	16	33%
<b>100 m propuestas</b>	48	19	40%

FIGURA 22.  
Análisis de porcentaje de cobertura.



Figura 22 (a). Análisis de cobertura de bahías actuales 50m



Figura 22 (b). Análisis de cobertura de bahías propuestas 50m



Figura 22 (c). Análisis de cobertura de bahías actuales 100m



Figura 22 (d). Análisis de cobertura de bahías propuestas 100m

#### 4.3.4. Cálculo de distancias y estimación de tiempos de servicio

Una vez identificados los proveedores, las bahías y los clientes, se calcularon las distancias, entre bahías, clientes y bahías-clientes. Se estimó el tiempo de servicio de entrega de las empresas a los clientes dada cierta bahía. Se utilizó Google Maps para calcular las distancias debido a que toma en cuenta el sentido de las calles. La Tabla 5 muestra la distancia entre bahías expresada en kilómetros, mientras que la Tabla 6

muestra el tiempo de desplazamiento entre las mismas expresado en minutos tomando en cuenta una velocidad estándar de 15 km/h de los camiones repartidores y un retraso de 3 minutos para simular tráfico y posibles contratiempos que puedan ocurrir, los cuales se determinaron por medio de lo arrojado por Google Maps. El tiempo de servicio tomó el mismo procedimiento descrito en la Sección 4.3.

**TABLA 5.**  
**Distancias en metros de bahías**

Bahías	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
<b>1</b>	0	300	450	1200	1400	700	800	400	1500	1100	1300	1200	400	180	800
<b>2</b>	450	0	140	950	1000	400	500	850	1200	850	1000	850	850	650	550
<b>3</b>	300	600	0	800	800	240	350	700	1000	700	1600	700	700	500	1200
<b>4</b>	350	650	800	0	140	650	350	750	1100	750	1700	700	750	550	1200
<b>5</b>	210	500	650	850	0	900	950	600	900	600	1500	1300	600	400	1000
<b>6</b>	950	1200	1400	600	750	0	160	1300	1600	1300	1500	500	1300	1100	1000
<b>7</b>	800	1100	1200	450	600	900	0	1200	1500	1200	1800	350	1200	1000	1300
<b>8</b>	1800	2100	2200	1700	1500	1700	1800	0	1800	1400	1000	2100	10	1900	450
<b>9</b>	650	950	1100	550	400	900	650	1000	0	1000	1600	1000	1000	800	1500
<b>10</b>	550	850	1000	450	350	850	550	1000	1700	0	1900	900	1000	750	1400
<b>11</b>	900	1200	1300	800	700	1200	900	1300	900	550	0	1300	1300	1100	1700
<b>12</b>	900	450	750	850	90	220	700	450	850	1500	1800	0	800	600	1300
<b>13</b>	1800	2100	2200	1700	1600	1700	1800	0	1800	1400	1000	2200	0	1500	500
<b>14</b>	600	120	260	1100	1200	500	600	210	1300	1000	1200	1000	230	0	600
<b>15</b>	1300	1600	1700	1200	1100	1600	1300	1700	1300	1000	500	1600	1700	1500	0

TABLA 6.  
**Tiempo de desplazamiento en minutos de bahías**

Bahías	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
<b>1</b>	0.0	4.2	4.8	7.8	8.6	5.8	6.2	4.6	9.0	7.4	8.2	7.8	4.6	3.72	6.2
<b>2</b>	4.8	0.0	3.6	6.8	7.0	4.6	5.0	6.4	7.8	6.4	7.0	6.4	6.4	5.6	5.2
<b>3</b>	4.2	5.4	0.0	6.2	6.2	3.96	4.4	5.8	7.0	5.8	9.4	5.8	5.8	5.0	7.8
<b>4</b>	4.4	5.6	6.2	0.0	3.6	5.6	4.4	6.0	7.4	6.0	9.8	5.8	6.0	5.2	7.8
<b>5</b>	3.8	5.0	5.6	6.4	0.0	6.6	6.8	5.4	6.6	5.4	9.0	8.2	5.4	4.6	7
<b>6</b>	6.8	7.8	8.6	5.4	6.0	0.0	3.6	8.2	9.4	8.2	9.0	5.0	8.2	7.4	7.0
<b>7</b>	6.2	7.4	7.8	4.8	5.4	6.6	0.0	7.8	9.0	7.8	10.0	4.4	7.8	7.0	8.2
<b>8</b>	10.2	11.4	11.8	9.8	9.0	9.8	10.2	0.0	10.2	8.6	7.0	11.4	3.1	10.6	4.8
<b>9</b>	5.6	6.8	7.4	5.2	4.6	6.6	5.6	7.0	0.0	7	9.4	7	7	6.2	9
<b>10</b>	5.2	6.4	7.0	4.8	4.4	6.4	5.2	7.0	9.8	0.0	11.0	6.6	7.0	6.0	8.6
<b>11</b>	6.6	7.8	8.2	6.2	5.8	7.8	6.6	8.2	6.6	5.2	0.0	8.2	8.2	7.4	9.8
<b>12</b>	6.6	4.8	6.0	6.4	3.3	3.9	5.8	4.8	6.4	9.0	10.0	0.0	6.2	5.4	8.2
<b>13</b>	10.2	11.4	11.8	9.8	9.4	9.8	10.2	3.0	10.2	8.6	7.0	11.8	0.0	9.0	5.0
<b>14</b>	5.4	3.5	4.0	7.4	7.8	5.0	5.4	3.8	8.2	7.0	7.8	7.0	3.9	0.0	5.4
<b>15</b>	8.2	9.4	9.8	7.8	7.4	9.4	8.2	9.8	8.2	7.0	5.0	9.4	9.8	9.0	0.0

La Tabla 7 representa una muestra de las distancias en kilómetros entre bahías y clientes, donde las filas representan a las bahías y las columnas a los clientes, calculada con base en Google Maps. La Tabla 8 representa el tiempo de desplazamiento, expresado en minutos, entre las bahías y los clientes tomando en cuenta una velocidad de caminata de 5km/h del repartidor siguiendo la misma estructura de la Tabla 7 de representación de bahías y clientes por filas y columnas, respectivamente.

TABLA 7.  
Muestra de distancia en metros desde bahías a clientes

	1	2	3	4	5	6	7	8	9	10
1	658.4	627.4	537.2	518.1	494.8	387.5	402.9	175.1	157.5	234.3
2	922.1	891	800.9	781.7	758.5	651.1	506.4	369.7	421.1	498
3	914.9	883.8	793.7	774.6	751.3	643.9	519.2	362.6	413.9	490.8
4	371	357.8	450	475.3	496.5	622.7	771.5	543.7	519.4	405.1
5	501.5	470.5	380.3	361.2	353.5	479.6	628.5	400.7	376.4	262
6	686.9	655.9	565.7	546.6	523.3	598.3	747.2	519.4	495.1	380.7
7	616.7	585.6	600	625.4	646.6	772.7	921.6	693.7	669.5	555.1
8	931.5	900.4	810.3	791.1	767.9	660.5	515.8	379.1	430.5	507.4
9	148.7	119.7	47.5	66.6	89.9	197.3	342	478.6	427.3	350.4
10	197.3	166.2	76.1	65.9	87.1	213.2	362.1	355	303.7	226.8
11	795.4	764.4	674.2	655.1	631.8	524.4	379.7	243.1	294.4	371.3
12	427.7	456.7	548.9	574.3	595.5	721.6	870.5	642.6	618.4	504
13	868.3	837.2	747.1	728	704.7	597.3	452.6	316	367.3	444.2
14	797	765.9	675.8	656.6	633.4	526	381.3	244.6	296	372.9
15	1165	1134	1044	1025	1002	894.4	749.7	613	664.4	741.2

TABLA 8.  
Muestra de tiempo de desplazamiento en minutos desde bahías hasta clientes

	1	2	3	4	5	6	7	8	9	10
1	7.9	7.5	6.4	6.2	5.9	4.7	4.8	2.1	1.9	2.8
2	11.1	10.7	9.6	9.4	9.1	7.8	6.1	4.4	5.1	6.0
3	11.0	10.6	9.5	9.3	9.0	7.7	6.2	4.4	5.0	5.9
4	4.5	4.3	5.4	5.7	6.0	7.5	9.3	6.5	6.2	4.9
5	6.0	5.6	4.6	4.3	4.2	5.8	7.5	4.8	4.5	3.1
6	8.2	7.9	6.8	6.6	6.3	7.2	9.0	6.2	5.9	4.6
7	7.4	7.0	7.2	7.5	7.8	9.3	11.1	8.3	8.0	6.7
8	11.2	10.8	9.7	9.5	9.2	7.9	6.2	4.5	5.2	6.1
9	1.8	1.4	0.6	0.8	1.1	2.4	4.1	5.7	5.1	4.2
10	2.4	2.0	0.9	0.8	1.0	2.6	4.3	4.3	3.6	2.7
11	9.5	9.2	8.1	7.9	7.6	6.3	4.6	2.9	3.5	4.5
12	5.1	5.5	6.6	6.9	7.1	8.7	10.4	7.7	7.4	6.0
13	10.4	10.0	9.0	8.7	8.5	7.2	5.4	3.8	4.4	5.3
14	9.6	9.2	8.1	7.9	7.6	6.3	4.6	2.9	3.6	4.5
15	14.0	13.6	12.5	12.3	12.0	10.7	9.0	7.4	8.0	8.9

#### 4.3.5. Cálculo de costo de uso de bahías

Debido al carácter del problema, el uso de las bahías de estacionamiento no es gratuito. Cada bahía tiene asociado un costo de uso, el cuál es variable. Este costo se calculó en base a tres parámetros y está expresado en número negativo en unidades de cajas “pérdidas” para simplicidad en los cálculos dentro del programa del ALNS. El primer parámetro es la ubicación de la bahía en cuanto al punto céntrico de la ciudad, entre más céntrica la bahía, más costosa. La segunda, la pertenencia a una avenida principal, si la bahía se encuentra en calle principal su costo es mayor. La tercera, la cantidad de tránsito vehicular de la calle en donde se encuentra la bahía, entre más transitada más costosa.

La Tabla 9 muestra el nivel de tránsito, ubicación y costo de cada bahía. El parámetro de tránsito puede representar tres valores, alto, medio o bajo. En cambio, los parámetros de calle principal y de ubicación céntrica son booleanos. El costo por un tránsito alto es de -3 unidades, mientras que el costo de un tránsito medio y uno bajo es de -2 y -1 respectivamente. El costo de la ubicación céntrica es de -1 al igual que si la bahía se encuentra en una calle principal.

TABLA 9.  
Costos de utilización de bahías

Bahia	Costo	Tránsito	Céntrico	Calle principal
1	-5	alto	x	x
2	-4	alto		x
3	-3	medio	x	
4	-3	alto		
5	-3	alto		
6	-2	medio		
7	-2	medio		
8	-3	medio	x	
9	-1	bajo		
10	-1	bajo		
11	-3	medio	x	
12	-3	alto		
13	-3	medio	x	
14	-4	medio	x	x
15	-5	alto	x	x

#### 4.3.6. Descripción de pedidos

Los pedidos de los clientes presentan las características que definen las restricciones del problema. Cada cliente identificado puede realizar más de un pedido a la misma empresa, esto debido a los tipos de productos que se solicitan. Es por eso que los pedidos se han estructurado en diversas categorías las cuales toman en cuenta la cantidad solicitada, la empresa a solicitar el pedido, y el día a entregar. En la Tabla 10 se muestra un ejemplo de pedidos de la Empresa 1. Existen seis columnas. La primera es un indicador del número de cliente. La segunda se subdivide en otras seis columnas ya que indican la existencia de una orden en cada uno de los días de la semana. La tercera indica la cantidad de cajas a entregar. La cuarta indica el tiempo de servicio requerido para repartir las cajas. La quinta indica el tipo de vehículo utilizado, mientras que la última indica la ventana horaria de cada cliente para que el pedido sea entregado.

TABLA 10.  
Pedidos de Empresa 1

Cliente	Días de entrega (0 – No entrega, 1 - Entrega)						Cajas a entregar	Tiempo de Servicio	Tipo de Vehículo	Ventana de Tiempo (min)	
	Lu	Ma	Mi	Ju	Vi	Sa				Desde	Hasta
2	1	0	1	0	1	0	5	17	4	540	660
4	1	0	0	0	0	0	2	11	4	540	660
4	0	0	0	1	0	0	2	11	2	540	660
4	0	1	0	1	0	0	3	11	2	540	660
13	0	0	0	0	0	1	3	12	4	900	1020
15	0	0	0	0	0	1	4	12	2	540	660
18	0	1	0	1	0	0	3	12	4	540	660
18	1	0	1	0	1	0	3	12	2	540	660
18	1	0	1	0	1	0	2	12	4	540	660
22	0	1	0	1	0	0	2	11	4	900	1020
22	1	0	0	0	1	0	3	11	4	900	1020
23	0	0	0	0	0	1	2	11	2	540	660
23	0	0	0	0	0	1	2	11	2	540	660
23	0	0	0	1	0	1	2	11	2	540	660
25	1	0	1	0	1	0	3	11	2	540	660
25	0	1	0	0	0	1	2	11	4	540	660
25	0	0	0	0	1	0	3	11	4	540	660
25	0	1	0	0	1	0	2	11	4	540	660
25	0	1	0	0	1	0	2	11	2	540	660
27	1	0	1	0	1	0	4	12	2	540	660

Cliente	Días de entrega (0 – No entrega, 1 - Entrega)						Cajas a entregar	Tiempo de Servicio	Tipo de Vehículo	Ventana de Tiempo (min)	
<b>36</b>	1	0	0	0	1	0	4	12	2	900	1020
<b>39</b>	0	0	0	0	0	1	2	11	2	540	660
<b>39</b>	0	1	0	0	1	0	3	11	4	540	660
<b>39</b>	1	0	1	0	0	0	3	11	4	540	660
<b>39</b>	0	1	0	1	0	0	2	11	4	540	660
<b>39</b>	0	1	0	1	0	0	3	11	2	540	660
<b>47</b>	1	0	1	0	1	0	5	17	4	540	660

## Capítulo V

### Resultados y discusiones

---

En este capítulo se describe la validación experimental de la solución propuesta al identificar escenarios en las órdenes de las empresas y realizar un comparativo entre las rutas resultantes y las rutas actuales dentro del marco de los escenarios identificados. Al final se analizan los resultados de dicha comparación para determinar que tanto mejoraron las ganancias con las rutas resultantes.

## 5.1. Validación Experimental

Para validar la solución propuesta, esto es, demostrar que existe una mejora con la implementación del generador de rutas, el optimizador binario AMPL y la metaheurística de ALNS, se realizó el siguiente procedimiento:

1. Identificación de escenarios en las órdenes reales para su clasificación de acuerdo a tres criterios (cantidad de clientes, promedio de demanda de producto de clientes y múltiples entregas).
2. Generación de rutas reales con algoritmo exacto y comparación con las rutas resultantes de la solución propuesta.
3. Evaluación de solución propuesta, generador de rutas y optimizador ALNS, en los distintos escenarios identificados.
4. Análisis de resultados.

## 5.2. Identificación de escenarios

Se analizaron los datos de las órdenes de las cuatro empresas para identificar los escenarios posibles para la experimentación. Los criterios para la identificación de estos escenarios son tres. El primero es el número de clientes, que pueden ser pocos (1-10) o muchos (11-25). El segundo es el promedio de demanda de cajas que necesitan los clientes de carácter booleano como pocos (0-15) o muchos (16-50). La tercera es la existencia de clientes que necesitan una entrega múltiple, esto es, que, debido a la necesidad de más de un tipo de producto, una misma empresa debe de hacer múltiples entregas a un mismo cliente. Se identificaron cinco escenarios distintos, los cuales se describen en la Tabla 11.

TABLA 11.  
Lista de escenarios

Núm. Escenario	Núm. Clientes	Demanda C	Múltiple Entrega
A	1-10	0-15	si
B	11-25	0-15	si
C	1-10	0-15	no
D	11-25	0-15	no
E	1-10	16-50	no

El escenario “A”, por ejemplo, describe una orden con pocos clientes (1-10), demanda baja (0-15) y con clientes que requieren múltiple entrega. En cambio, el escenario “B” describe una orden con muchos clientes (11-25), demanda baja (0-15) y clientes que requieren múltiple entrega.

### 5.3. Comparativa de rutas actuales con rutas resultantes

Una vez identificados los escenarios, se evaluaron las rutas resultantes con las rutas que actualmente utilizan las empresas. Para poder determinar las rutas actuales, se utilizó un método constructivo de asignación bahía-cliente con planificación de vehículos desarrollado en Matlab para generar rutas desarrollado por nuestro codirector, emulando a como lo hacen los choferes con su experiencia, que representan las rutas tal como las empresas las generan. Estas rutas reales son las que se utilizaron para la evaluación con las rutas resultantes. El pseudocódigo de este método constructivo de asignación se describe en el Algoritmo 8 con la notación matemática siguiente:

Conjunto de datos:

$V_c$  Lista de tipos de vehículo por compañía  $c \in C$

$B$  Lista de Bahías

$B_{s(o)}$  Lista de bahías desde las cuales se puede atender a cliente  $s(o)$

$O_c$  Lista de ordenes de la compañía  $c \in C$

$O_c^k$  Lista de ordenes de la compañía  $c \in C$

que pueden ser atendidos por el vehículo  $k$

$B_c^k$  Lista de bahías utilizadas relacionados con la compañía  $c \in C$  por vehículo  $k$

$S_c^k$  Lista de tiendas relacionadas con la compañía  $c \in C$  por vehículo  $k$

Conjunto de parámetros:

$d_{max}$  Distancia maxima de bahía a tienda

$n_c^v$  Número de vehículos de tipo  $v \in V_c$  por compañía  $c \in C$

$q_c^v$  Total de la capacidad del vehículo tipo  $v \in V_c$  relacionado con la compañía  $c \in C$

$t_{b,s(o)}$  Tiempo de viaje desde la bahía  $b \in B$  a la tienda  $s(o)$

$\Delta t_o$  Tiempo de margen para atender la orden  $o \in O_c$

Variables:

$k$  Número de vehículos utilizados

$un_c^v$  Número de vehículos utilizados de tipo  $v \in V_c$  por compañía  $c \in C$

$\Delta q_c^v$  Capacidad restante del vehículo tipo  $v \in V_c$  relacionado con la compañía  $c \in C$

$T_c^k$  Tiempo de terminación actual del vehículo  $k$  por compañía  $c \in C$

$T_b$  Tiempo disponible actual para estacionarse en la bahía  $b \in B$

$T_b^k$  Tiempo de arribo mas temprano actual en la bahía  $b \in B$  por vehículo  $k$

$T_b^o$  Tiempo para acompletar mas temprano en la

bahía  $b \in B_o$  para entregar el producto en la tienda  $s(o)$

#### ALGORITMO 8.

#### Método constructivo de asignación bahía-cliente con planificación de vehículos

---

<b>Entrada:</b>	$B, d_{max}, V_c, O_c, q_c, n_c,$
<b>Salida:</b>	$\Delta q_c, T_c, S_c, B_c$

---

**Inicio**

1. Ordenar la lista  $O_c$  por incremento de los tiempos más tempranos de llegada  $l_o$ , si existen dos o mas elementos con  $l_o$  igual, entonces ordenar por márgenes de tiempo  $\Delta t_o = u_o - l_o - st_o$ .
2. Ordenar los tipos de vehículos  $V_c$  por capacidad  $q_c^v$  en decremento.
3. Inicializar  $v = 1, un_c^v = 0, \forall v \in V_c, k = 0, T_b = 0 \forall b \in B$
4. **Repetir**
5.     **Si**  $un_c^v = n_c^v$  **entonces**
6.         Incrementar identificador de tipo de vehículo  $v = v + 1$
7.     **Fin si**
8.     Incrementar número de vehículos utilizados  $k = k + 1$
9.     Incrementar el número de vehículos utilizados de tipo  $v, un_c^v = un_c^v + 1$
10.    Inicializar la capacidad restante del vehículo  $k: T_b^k = \max\{l_o, T_b\} \forall b \in B$
11.    Inicializar la lista de tiendas atendidas:  $S_c^k = \{\}$ , y la lista de bahías utilizadas:  $B_c^k = \{\}$
12.    Hacer una copia de la lista de las órdenes no atendidas:  $O_c^k = O_c$
13.    **Repetir**
14.     Seleccionar la orden superior de la lista de órdenes  $O_c^k$
15.     **Si**  $\Delta q_c^k \leq g_o$  **entonces**
16.         Ordenar la sublista de bahías  $B_{s(o)} = \{b \in B: d_{b,s(o)} \leq d_{max}\}$  por tiempo de terminación del tiempo más temprano en incremento:  $T_b^o = \max\{T_b^k, l_o\} + st_b^{s(o)}$
17.         Seleccionar la bahía superior de la sublista de bahías  $B_{s(o)}$
18.         **Si**  $T_b^k \leq u_o$  **entonces**
19.             Actualizar la lista de tiendas atendidas:  $S_c^k = S_c^k \cup \{s(o)\}$ , y la lista de bahías utilizadas  $B_c^k = B_c^k \cup \{b\}$
20.             Actualizar la capacidad restante del vehículo:  $\Delta q_c^k = \Delta q_c^k - g_o$
21.             Actualizar el tiempo de terminación del vehículo y de la bahía utilizada:  $T_c^k = T_b = T_b^o + st_b^{s(o)}$

---

- 
22. Actualizar los tiempos de arribo siguientes más próximos en las bahías para el vehículo  $k$ :  $T_{b'}^k = \max\{T_{b'}^k, T_c^k + t_b, t_{b'}\}, \forall b' \in B$
  23. Borrar la orden superior  $o$  de la lista  $O_c$
  24. **Fin Si**
  25. **Fin Si**
  26. Borrar la orden superior  $o$  de la lista  $O_c^k$
  27. **Hasta que**  $O_c^k = 0$
  28. **Hasta que**  $O_c = 0$  o  $\sum_{v' \in V_c} un_{c'}^{v'} < \sum_{v' \in V_c} n_{c'}^{v'}$
- 

Las rutas resultantes se generaron con la integración del generador de rutas y de la ALNS. La ejecución de la optimización se realizó en una computadora con procesador Intel Core I3 de séptima generación con 8gb de RAM. El programa fue desarrollado en Netbeans 8.2 con una versión de Java 1.8.0.

La Figura 23 ilustra un ejemplo de ruta actual de la orden del día lunes, del vehículo dos, de la Empresa 1. En donde en el recuadro punteado se representan las tiendas atendidas por la ruta, mientras que el recuadro con líneas y el de línea continua representan las bahías y la ventana temporal de entrega, respectivamente. Estas rutas son generadas por cada tipo de vehículo, por cada día y por cada empresa.

FIGURA 23.  
Ruta actual de día lunes de Empresa 1 del vehículo 2

lunes	
Route '1'	
Store '18 Abarrotes 4' ->	attended from bay 'doce' between 09:00 and 09:12 (Delayed Time = 0 min, Dead time = 0 min)
Store '27 Abarrotes 7' ->	attended from bay 'seis' between 09:13 and 09:25 (Delayed Time = 13 min, Dead time = 1 min)
Store '25 Oxxo 1' ->	attended from bay 'seis' between 09:25 and 09:36 (Delayed Time = 25 min, Dead time = 0 min)
Store '36 Abarrotes 9' ->	attended from bay 'uno' between 15:00 and 15:12 (Delayed Time = 0 min, Dead time = 324 min)

Una vez generadas estas rutas que representan la situación actual, se realiza un procedimiento de transformación para adaptarla a un formato que pueda leer el programa de ALNS y proporcionar la ganancia de la ruta. La Figura 24 representa la ruta ya transformada.

FIGURA 24.  
Ruta actual transformada para ALNS

[[d10, b120, c70, b120, b60, c200, b60, b60, c150, b60, b11, c211, b11, d11]] = 4
---

Una vez transformadas y evaluadas, se almacenan los resultados para una posterior comparación con las soluciones propuestas. Un fragmento de las rutas resultantes de la solución propuesta y su ganancia se ilustran en la Tabla 12.

TABLA 12.  
**Ruta óptima y ganancia**

Vehículo	Lunes	
	Ruta K-Caminos	Ruta ALNS
<b>2</b>	[[d10, b90, c70, b90, b40, c200, b40, b10, c150, b10, b141, c211, b141, d11]] = 1	[[d10, b120, c70, b120, b60, c200, b60, b60, c150, b60, b31, c211, b31, d11]] con ganancia = 6

Una vez generadas todas las rutas resultantes de la solución propuesta, se realizó una tabla comparativa para evaluar cada ruta resultante con la ruta actual. Un fragmento de esta tabla comparativa que representa la ganancia de las tres rutas de la Empresa 1 utilizando el vehículo 2 del día lunes se ilustra en la Tabla 13. La primera columna ilustra el tipo de vehículo utilizado, la segunda columna se divide en cuatro, de las cuales, las primeras tres representan la ganancia de la ruta en cuestión mientras que la última representa el porcentaje de mejora de la ruta resultante con respecto a la ruta actual. En el Apéndice A bajo la Tabla A1 se ilustra la totalidad de las 21 rutas distintas.

TABLA 13.  
**Rutas y ganancias de E1, día lunes y vehículo 2**

Tipo de Veh.	Lunes			
	real	k	ALNS	% mejora
<b>2</b>	4	1	6	50,00

#### 5.4. Evaluación de la solución propuesta de acuerdo a escenarios

Una vez identificados los escenarios, las rutas actuales y las rutas resultantes, se llevó a cabo la evaluación de acuerdo a los escenarios identificados en la Sección 5.2. En primer lugar, se identificaron la cantidad de órdenes en cada escenario descrito en la Tabla 14.

TABLA 14.  
**Lista de escenarios**

Num. Escenario	Cantidad de órdenes
<b>A</b>	5
<b>B</b>	1
<b>C</b>	6
<b>D</b>	4
<b>E</b>	5

Una vez clasificadas las órdenes, se procedió a evaluar las rutas resultantes de acuerdo a la clasificación por escenarios. La Tabla 15 muestra el porcentaje de mejora que obtuvieron las rutas resultantes con respecto a las actuales en cada uno de los escenarios.

TABLA 15.  
**Porcentaje de mejora de escenarios**

Escenario	Cantidad de órdenes	Promedio % mejora
<b>A</b>	5	31.67
<b>B</b>	1	100
<b>C</b>	6	15.14
<b>D</b>	4	12.41
<b>E</b>	5	2.41

## 5.5. Análisis de resultados

Por último, se analizaron los resultados de la optimización de las rutas. En primer lugar, se analizó el porcentaje de rutas que mejoraron debido a la reducción de bahías en la ruta resultante. Debido a que el espacio de búsqueda de soluciones del ALNS se limita solo a las combinaciones de los elementos de la solución inicial de entrada, esto es, a la configuración de los clientes presentes en la solución, existen, en este contexto, dos maneras de maximizar la ganancia de las rutas. La primera es la de reducir el número de bahías utilizadas, disminuyendo así, el costo total de la ruta ya que solo se restaría el costo de las bahías que se utilizaron en la ruta. La segunda es la de buscar bahías menos costosas que satisfagan los requisitos de temporalidad. La Tabla 16 muestra la información sobre este análisis. Se puede observar que la reducción de bahías tuvo lugar en 13 de las 21 observaciones. Se puede consultar a detalle este análisis en el Apéndice A en la Tabla A2.

TABLA 16.  
Rutas con mejora por reducción de bahías

	Total de rutas	Rutas que mejoran por reducción de bahías	% Rutas con mejora por reducción
<b>Empresa 1</b>	6	3	50
<b>Empresa 2</b>	4	4	100
<b>Empresa 3</b>	5	2	40
<b>Empresa 4</b>	6	4	66.67

Para un segundo análisis general se migraron los datos de las órdenes de las empresas a un archivo CSV para ser procesado en el lenguaje R para generar gráficos que faciliten la interpretación de las mejoras de las rutas resultantes, el cual se ilustra en la Tabla 17.

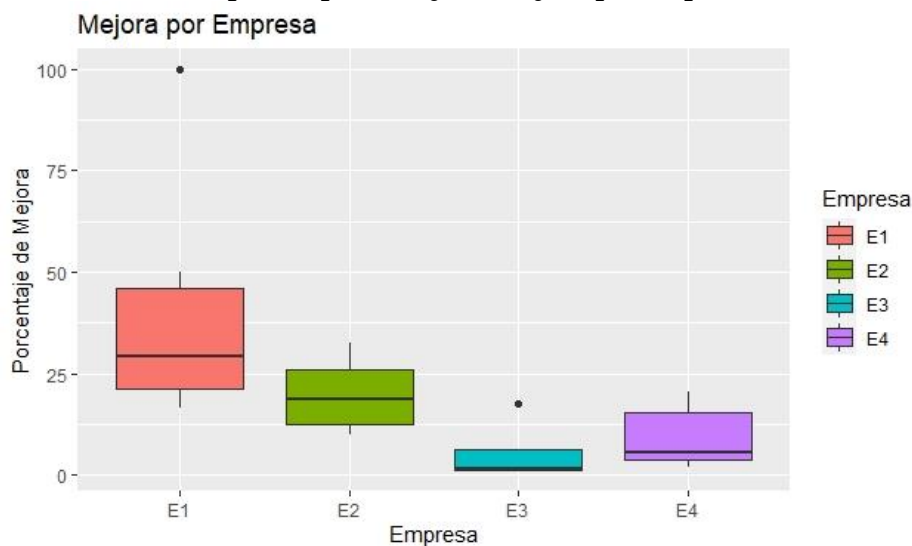
TABLA 17.  
CSV para análisis de rutas resultantes

Núm.	Empresa	Día	Prom. demanda	Múltiple Entrega	Clientes	% Mejora	Reducción de bahías	Escenario	Vehículos
1	E1	lunes	3.4	Si	10	50.00	No	A	2
2	E1	martes	2.44	Si	9	25.00	No	A	2
3	E1	miércoles	3.57	Si	7	20.00	Si	A	2
4	E1	jueves	2.42	Si	7	100.00	Si	A	2
5	E1	viernes	3.25	Si	12	33.33	Si	B	2
6	E1	sábado	2.42	Si	7	16.67	No	A	2
7	E2	martes	7.18	No	22	32.60	Si	D	2
8	E2	miércoles	5.75	No	8	10.00	Si	C	2
9	E2	viernes	7.19	No	21	13.34	Si	D	2
10	E2	sábado	5.85	No	7	23.78	Si	C	2
11	E3	lunes	13.5	No	4	17.50	Si	C	2
12	E3	miércoles	17.25	No	4	6.14	Si	E	2
13	E3	jueves	30.5	No	2	1.75	No	E	2
14	E3	viernes	50.00	No	2	1.07	No	E	2
15	E3	sábado	36.00	No	2	1.07	No	E	2
16	E4	lunes	12.73	No	15	3.66	No	D	3
17	E4	martes	10.00	No	7	6.45	Si	C	3
18	E4	miércoles	14.44	No	9	18.49	Si	C	3
19	E4	jueves	12.81	No	16	4.43	Si	D	3
20	E4	viernes	15.33	No	6	1.93	Si	E	3

Núm.	Empresa	Día	Prom. demanda	Múltiple Entrega	Clientes	% Mejora	Reducción de bahías	Escenario	Vehículos
21	E4	sábado	8.8	No	5	20.63	No	C	3
<b>Total</b>	X	X	12.61	X	X	19.42	X	X	X

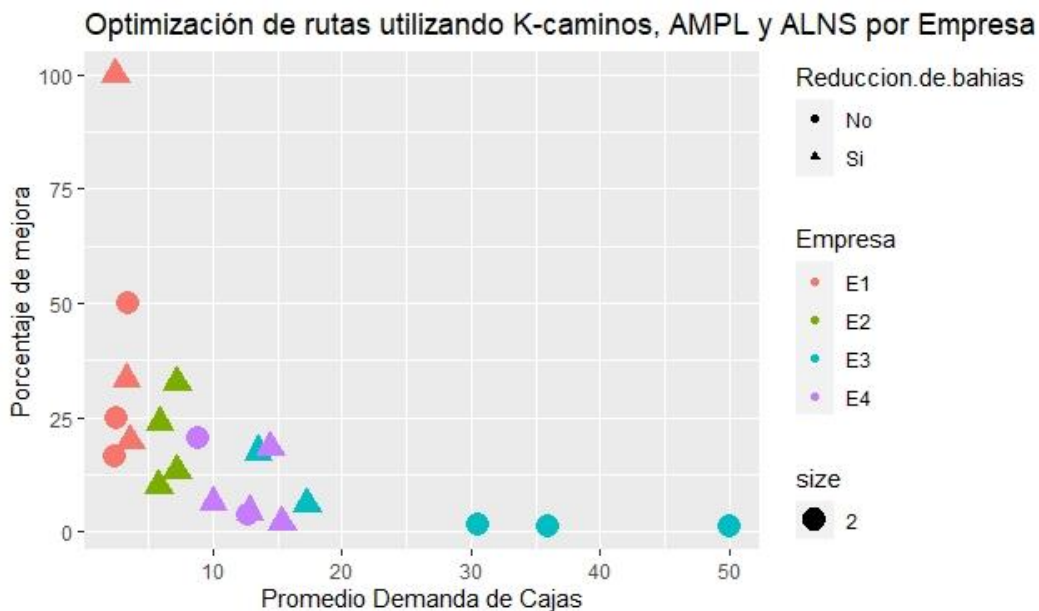
La primera columna es un identificador del número de orden. La segunda columna indica de que empresa es la orden. La tercera columna indica el día de la orden. La cuarta indica un promedio de la demanda de los clientes que componen la orden. La columna de “Múltiple Entrega” indica si hay clientes que requieren que más de una entrega de la misma empresa. En la columna siguiente, la sexta, se indica la cantidad de clientes que conforman la orden. En la séptima columna se indica el porcentaje de mejora que tiene la ruta resultante de acuerdo a la actual, esto se tomó de la Tabla A1 que se muestra en el apéndice A- En la octava columna se indica si existe una reducción de bahías para su optimización mientras que las últimas dos representan el escenario al que pertenece la orden, así como el número de vehículos que pueden atender la orden, respectivamente. Con esto, se buscó generar gráficos que muestren de mejor manera cómo se optimizaron las rutas. En la Figura 25 se visualiza un boxplot del porcentaje de mejora que existió en cada empresa. Donde se puede observar que la empresa que mejoró más es la Empresa 1, en donde el 50% de sus órdenes tuvieron una mejora entre el 28% y el 50%, con un *outlier* o valor excepcional de mejora de un 100%. La empresa con una menor mejora fue la Empresa 3.

FIGURA 25.

**Boxplot de porcentaje de mejora por empresa**

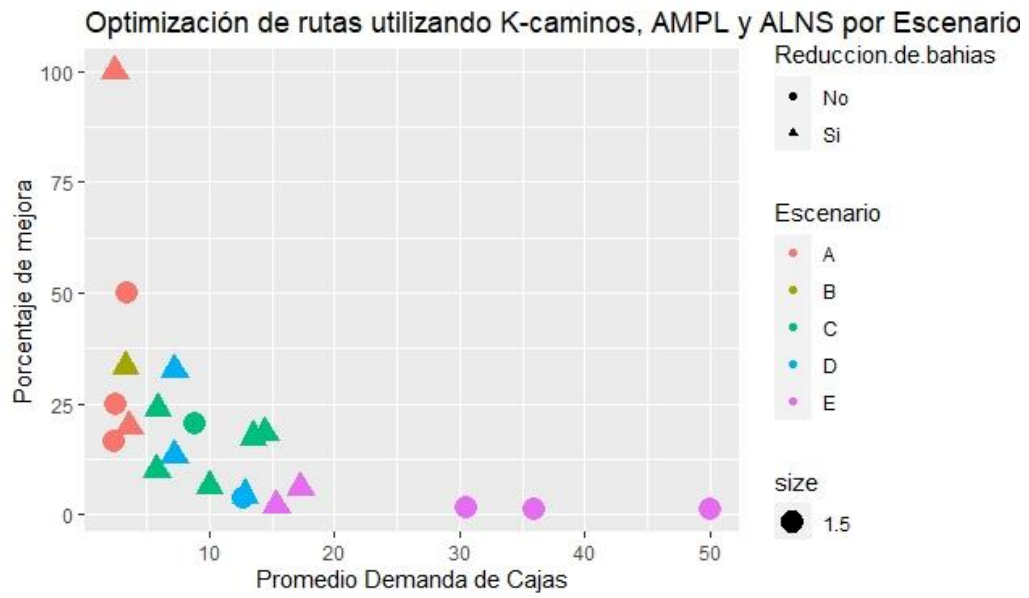
Después, se analizaron las mejoras conforme a escenarios y a empresas de acuerdo al promedio de demanda de clientes y si éstas se realizaron por una reducción de bahías en la Figura 26 y Figura 27, respectivamente. En la Figura 24 las observaciones con forma de triángulo indica que hubo una reducción de bahías para la optimización de esa observación en particular, el círculo representa la reducción por medio de nuevas bahías menos costosas. El color rosa representa las observaciones de la empresa 1, las verdes de la empresa 2, las moradas de la empresa 3 y las azules de la empresa 4. El eje “x” representa el promedio de demanda por cajas, mientras que el eje “y” representa el porcentaje de mejora.

FIGURA 26.  
Gráfico de dispersión de mejora según empresa



En la Figura 25 los triángulos representan las observaciones en donde hubo optimización debido a la reducción de bahías mientras que los círculos representan optimización por el reemplazo de bahías por aquellas menos costosas. El color representa el escenario de la observación siendo rosa para el escenario A, oro para el escenario B, verde para el escenario C, azul para el escenario D y morado para el escenario E. El eje horizontal representa el promedio de demanda de cajas, mientras que el eje vertical representa el porcentaje de mejora.

FIGURA 27.  
Gráfico de dispersión de mejora según escenario



## **Capítulo VI**

### **Conclusiones y trabajo futuro**

---

En este capítulo se describe la conclusión y los trabajos futuros que surgieron con el desarrollo de este proyecto de investigación.

## 6.1. Conclusión

Este trabajo presenta una propuesta para optimizar bahías de las rutas de reparto utilizando un algoritmo de Búsqueda Adaptativa de Vecindad Amplia (ALNS) tomando en cuenta un generador de rutas preliminar y una proyección de dichas rutas en un programa basado en Lenguaje de Programación de Modelo Algebraico (AMPL). Esta combinación de generador de rutas y ALNS brinda soluciones para el problema del camión y su remolque con restricciones de tiempo, capacidad y múltiples entregas.

Se tuvieron cuatro fases para el desarrollo de este trabajo de investigación. La primera fase consistió en analizar y evaluar trabajos relacionados con metaheurísticas que resolvían problemas similares. En esta fase se pudo observar que no había, en la literatura consultada, metaheurísticas que resolvieran el problema del presente trabajo tal como se describió en la Sección 3.1.

La segunda fase consistió en construir un modelo del proceso de distribución de las empresas de acuerdo a los datos adquiridos y estimados. En esta fase se desarrolló el modelo basado en AMPL que se utilizó para la proyección de las rutas resultantes del generador. Esta fase se describe en las secciones 4.2 y 4.2.1.

La tercera fase consistió en analizar el modelo de AMPL, seleccionar la ALNS debido a que fue la metaheurística más adecuada de acuerdo a las restricciones presentes y desarrollar el algoritmo. tal como se muestra en la Sección 4.2.3.

La cuarta fase consistió en evaluar la solución obtenida. Para evaluar se llevó a cabo un proceso de clasificación e identificación de las distintas órdenes de las empresas, tomando en cuenta la cantidad de clientes, el promedio de demanda de productos y la existencia de múltiples entregas. Una vez clasificadas las órdenes, se generaron las rutas resultantes de la propuesta de este trabajo y de las rutas reales con un algoritmo exacto. Después se evaluaron los resultados y se analizó la mejora de las rutas resultantes a comparación de las reales, tal como se describe en la Sección 5.1. La utilización de este binomio de generador-proyección de rutas junto con el optimizador de ALNS, logra construir rutas resultantes que generan un promedio de mejora en ganancia de 19.40% para cuatro empresas en la ciudad de Celaya, tal como se muestra en la Tabla 17. Dicha mejora se establece en un marco de referencia de 15 bahías de estacionamiento y atención a 48 clientes distintos.

Se cumplieron los objetivos, tanto el general como los particulares, planteados al inicio, ya que se optimizó la generación de rutas de reparto en base a las bahías de estacionamiento (véase Capítulo 5), se modeló

matemáticamente el problema de distribución de mercancías considerando las tres restricciones presentes: ventanas de tiempo, capacidad y demanda. Se desarrolló el ALNS aportando al estado del arte cinco consideraciones que no se encontraron en la literatura consultada (véase Sección 4.2.3), esto es:

- Incorporar restricciones de acceso a bahías de estacionamiento
- Incorporar múltiples ventanas de tiempo para descarga y entrega de productos
- Trabajar con escenario real
- Formular nuevos modelos de optimización, en este caso un generador de rutas, un optimizador en AMPL y un metaheurístico ALNS
- Desarrollar un algoritmo de ALNS, adhoc al ETTRP con las características descritas

Así mismo, se evaluó el desempeño de la solución propuesta, demostrando una mejora en las ganancias de las empresas evaluadas (véase Sección 5.1).

En la literatura consultada no se encontró un algoritmo que resolviera la problemática que presentaban estas empresas, por lo que la propuesta se expone como un aporte a la literatura para resolución de problema extendido del camión y su remolque.

## **6.2. Trabajo futuro**

Con lo observado durante el desarrollo de este trabajo, se han detectado algunos trabajos para evaluar mejor la solución propuesta donde se sugiere lo siguiente:

- a. Conversión de instancias de la literatura para realizar una comparación con otras metaheurísticas.
- b. Tomar en cuenta escenarios con una mayor cantidad de clientes y bahías para evaluar su funcionamiento.
- c. Tomar en cuenta escenarios donde existan más de cuatro empresas que comparten bahías para descarga y entrega de mercancía.
- d. Explorar nuevas estrategias de generación de rutas iniciales que alimenten al ALNS y evaluar su desempeño.
- e. Probar con otros operadores de destrucción-reparación.

## Referencias

- [1] I.-M. Chao, «A tabu search method for the truck and trailer routing problem,» *Computers & Operations Research*, vol. 29, pp. 33-51, 2002.
- [2] M. A. G.-V. a. O. G.-G. R. M. Campos-Garcia, «Relationship between cost and logistic practices in small and medium enterprises: A case study of Queretaro, Mexico,» *African Journal of Business Management*, vol. 5, n° 4, pp. 1245-1252, 2011.
- [3] M. A. a. M. Yousefikhoshbakht, «A Combination of Meta-heuristic and Heuristic Algorithms for the VRP, OVRP and VRP with Simultaneous Pickup and Delivery,» *Broad Research in Artificial Intelligence and Neuroscience*, vol. 8, n° 2, pp. 81-95, 2017.
- [4] A. M. a. R. R. S. Daya Ram Gaur, «Improved approximation algorithms for cumulative VRP with stochastic demands,» *Discrete Applied Mathematics*, vol. 1, n° 1, pp. 1-11, 2018.
- [5] G. Laporte, «The Vehicle Routing Problem: An overview of exact and approximate algorithms,» *European Journal of Operational Research*, pp. 345-358, 1992.
- [6] S. F. S. M. T. F. G. a. M. B. F. Hasan Hosseini-Nasab, «Classification of facility layout problems: a review study,» *International Journal of Advanced Manufacturing Technology*, vol. 94, n° 3, pp. 957-977, 2018.
- [7] S. M. & S. Ahmed, «Memetic Heuristic Approach for Solving Truck and Trailer Routing Problems with Stochastic Demands and Time Windows,» *Netw Spat Econ*, vol. 1, n° 1, p. 1093-1115, 2015.
- [8] M. R. G. a. D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, Portland, United States of America: W.H.Freeman & Co Ltd, 1979.
- [9] X.-S. Yang, «Metaheuristic Optimization: Algorithm Analysis and Open Problems,» *math.OC*, vol. I, n° 1212, 2012.
- [10] I. L. J. & S. P. Boussaid, «A survey on Optimization Metaheuristics,» *Information Sciences*, 2013.
- [11] M. P. L. S. T. & V. K. Birattari, «Classification of Metaheuristics and Design of Experiments for the Analysis of Components,» *Tech. Rep*, vol. 01, n° 05, 2001.
- [12] S. R. & D. Pisinger, «An Adaptive Large Neighborhood Search Heuristic for the Pickup and Delivery Problem with Time Windows,» *Transportation Science*, vol. 40, n° 4, pp. 455-472, 2006.
- [13] F. S. a. E. Taillard, «Solving real-life vehicle routing problems efficiently using tabu search,» *Annals of Operation Research*, vol. 41, n° 1, pp. 469-488, 1993.
- [14] S. Scheuere, «A tabu search heuristic for the truck and trailer routing problem,» *Computers & Operations Research*, vol. 33, n° 1, pp. 894-909, 2006.
- [15] V. F. Y. a. S.-Y. C. Shih-Wei Lin, «Solving the truck and trailer routing problem based on a simulated annealing heuristic,» *Computers & Operations Research*, vol. 36, pp. 1683-1692, 2009.

- [16] M. C. a. F. Guerriero, «A heuristic approach for the truck and trailer routing problem,» *The Journal of the Operational Research Society*, vol. 61, n° 7, pp. 1168-1180, 2010.
- [17] M. D. a. S. I. Ann-Kathrin Rothenbächer, «Branch-and-Price-and-Cut for the Truck-and-Trailer Routing Problem with Time Windows,» *Transportation Science*, vol. 1, n° 1, p. 1-17, 2018.
- [18] J.-F. C. Sophie N. Parragh, «Branch-and-price and adaptive large neighborhood search for the truck and trailer routing problem with time windows,» *Computers and Operations Research*, vol. 83, n° 1, pp. 28-44, 2017.
- [19] M. P. a. U. V. Ulrich Derigs, «Truck and trailer routing - Problems, heuristics and computational experience,» *Computers and Operations research*, vol. 40, n° 2, pp. 536-546, 2013.
- [20] G. Dodig-Crnkovic, «Scientific Methods in Computer Science,» de *Conf. for the Promotion of Research in IT*, Skövde, Sweden, 2002.
- [21] I. S. Grechikhin, «Iterative Local Search Heuristic for Truck and Trailer Routing Problem,» *Models, Algorithms, and Technologies for Network Analysis*, vol. 197, pp. 67-77, 2016.
- [22] J. P. C. P. C. M. A. a. V. N. Villegas, «A GRASP with evolutionary path relinking for the truck and trailer routing problem,» *Computers and Operations Research*, vol. 38, n° 9, pp. 1319-1334, 2011.
- [23] S. Yuan, J. Fu, F. Cui y X. Zhang, «Truck and Trailer Routing Problem Solving by a Backtracking Search Algorithm,» *Journal of Systems Science and Information*, vol. 8, n° 3, pp. 253-272, 2020.
- [24] Q. Xu, Z. Li, L. Shen, Z. Bian, L. Xing y Z. Jin, «Optimization on combined scheduling of tractor and trailer routing problem considering synchronized operations,» *International Journal of Innovative Computing, Information and Control*, vol. 15, n° 3, pp. 983-995, 2019.
- [25] D. Sacramento y D. a. R. S. Pisinger, «An adaptive large neighborhood search metaheuristic for the vehicle routing problem with drones,» *Transportation Research Part C*, n° 102, pp. 289-315, 2019.

## Apéndice A

TABLA A1.  
Rutas actuales y resultantes con sus respectivas ganancias

Tabla donde se muestra, remarcado en amarillo, las rutas en donde existe mejora en la ganancia, esto es, en 33 rutas de 44.

Emp 1	Veh.	Lunes				Martes				Miércoles			
		Real	K	ALNS	% mejora	Real	K	ALNS	% mejora	Real	K	ALNS	% mejora
	2	4	1	6	50	3	3	3	0	5	1	5	0
	4	6	0	9	50	6	3	9	50	5	1	7	40
Emp 1	Veh.	Jueves				Viernes				Sábado			
		Real	K	ALNS	% mejora	Real	K	ALNS	% mejora	Real	K	ALNS	% mejora
	2	0	5	6	div. 0	6	3	8	33.33	6	8	8	33.33
	4	-2	0	3	negativo	9	3	12	33.33	2	1	2	0
Emp 2	Veh.	Martes				Miércoles				Viernes			
		Real	K	ALNS	% mejora	Real	K	ALNS	% mejora	Real	K	ALNS	% mejora
	2	42	65	68	61.9	20	19	20	0	60	66	66	10
	4	91	64	94	3.3	15	15	18	20	60	62	70	16.67
Emp 2	Veh.	Sábado											
		Real	K	ALNS	% mejora								
	2	12	16	17	41.67								
	4	17	13	18	5.88								
Emp 3	Veh.	Lunes				Miércoles				Jueves			
		Real	K	ALNS	% mejora	Real	K	ALNS	% mejora	Real	K	ALNS	% mejora
	2	N/A	N/A	N/A		24	24	25	4.17	N/A	N/A	N/A	
	4	40	43	47	17.5	37	34	40	8.11	57	54	58	1.75
Emp 3	Veh.	Viernes				Sábado							
		Real	K	ALNS	% mejora	Real	K	ALNS	% mejora				
	2	48	47	48	0	20	19	20	0				
	4	47	45	48	2.13	47	45	48	2.13				
Emp 4	Veh.	Lunes				Martes				Miércoles			
		Real	K	ALNS	% mejora	Real	K	ALNS	% mejora	Real	K	ALNS	% mejora
	1	14	12	15	7.14	30	30	30	0	55	55	57	3.64
	3	95	92	97	2.11	31	31	35	12.9	45	42	45	0
	4	58	54	59	1.72	N/A	N/A	N/A		9	9	12	33.33
Emp 4	Veh.	Jueves				Viernes				Sábado			
		Real	K	ALNS	% mejora	Real	K	ALNS	% mejora	Real	K	ALNS	% mejora
	1	6	5	6	0	15	15	15	0	14	17	18	28.57
	3	113	122	123	8.85	35	30	36	2.86	4	4	4	0
	4	N/A	N/A	N/A		34	35	35	2.94	9	11	12	33.33

TABLA A2.

**Rutas actuales y resultantes representando la optimización por reducción de bahías**

Tabla donde se muestra que de las 33 rutas que presentaron mejora en ganancia, 16 tuvieron una reducción del número de bahías.

Emp 1	Veh.	Lunes				Martes				Miércoles			
		Real	K	ALNS	% reducción	Real	K	ALNS	% reducción	Real	K	ALNS	% reducción
	2	3	3	3	0	2	2	2	0	2	3	2	0
	4	5	6	5	0	3	4	3	0	4	4	3	25
Emp 1	Veh.	Jueves				Viernes				Sábado			
		Real	K	ALNS	% reducción	Real	K	ALNS	% reducción	Real	K	ALNS	% reducción
	2	3	2	1	66.67	3	4	3	0	2	2	2	0
	4	3	3	3	0	5	5	4	20	2	2	2	0
Emp 2	Veh.	Martes				Miércoles				Viernes			
		Real	K	ALNS	% reducción	Real	K	ALNS	% reducción	Real	K	ALNS	% reducción
	2	5	4	3	40	2	2	2	0	6	3	3	50
	4	3	6	2	33.33	3	2	1	66.67	4	5	4	0
Emp 2	Veh.	Sábado											
		Real	K	ALNS	% reducción								
	2	3	2	2	33.33								
	4	2	2	1	50								
Emp 3	Veh.	Lunes				Miércoles				Jueves			
		Real	K	ALNS	% reducción	Real	K	ALNS	% reducción	Real	K	ALNS	% reducción
	2		N/A	N/A		2	1	1	50		N/A	N/A	
	4	4	3	3	25	1	2	1	0	1	2	1	0
Emp 3	Veh.	Viernes				Sábado							
		Real	K	ALNS	% reducción	Real	K	ALNS	% reducción				
	2	1	1	1	0	1	1	1	0				
	4	1	1	1	0	1	1	1	0				
Emp 4	Veh.	Lunes				Martes				Miércoles			
		Real	K	ALNS	% reducción	Real	K	ALNS	% reducción	Real	K	ALNS	% reducción
	1	2	2	2	0	1	1	1	0	3	2	2	33.33
	3	3	4	3	0	3	2	2	33.33	1	2	1	0
	4	1	3	2	-100		N/A	N/A		2	2	2	0
Emp 4	Veh.	Jueves				Viernes				Sábado			
		Real	K	ALNS	% reducción	Real	K	ALNS	% reducción	Real	K	ALNS	% reducción
	1	1	1	1	0	1	1	1	0	2	2	2	0
	3	5	3	3	40	2	3	1	50	1	1	1	0
	4		N/A	N/A		1	1	1	0	2	2	2	0