

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN

FACULTAD DE INGENIERÍA MECÁNICA Y ELÉCTRICA



**PROBLEMA DINÁMICO DE RUTAS DE VEHÍCULOS
CENTRADO EN EL CLIENTE**

**POR
MAYRA ALEJANDRA LUNA PEÑA**

**COMO REQUISITO PARCIAL PARA OBTENER EL GRADO DE
MAESTRÍA EN CIENCIAS EN INGENIERÍA DE SISTEMAS**

JUNIO 2016

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN
FACULTAD DE INGENIERÍA MECÁNICA Y ELÉCTRICA
SUBDIRECCIÓN DE ESTUDIOS DE POSGRADO



**PROBLEMA DINÁMICO DE RUTAS DE VEHÍCULOS
CENTRADO EN EL CLIENTE**

**POR
MAYRA ALEJANDRA LUNA PEÑA**

**COMO REQUISITO PARCIAL PARA OBTENER EL GRADO DE
MAESTRÍA EN CIENCIAS EN INGENIERÍA DE SISTEMAS**

JUNIO, 2016

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

Los miembros del Comité de Tesis recomendamos que la Tesis «Problema dinámico de rutas de vehículos centrado en el cliente», realizada por la alumna Mayra Alejandra Luna Peña, con número de matrícula 1387785, sea aceptada para su defensa como requisito parcial para obtener el grado de Maestría en Ciencias en Ingeniería de Sistemas.

El Comité de Tesis



Dra. Ada Margarita Álvarez Socarrás

Asesor



Dr. José Marcos Moreno Vega

Revisor

Dra. Iris Abril Martínez Salazar

Revisor

Vo. Bo.



Dr. Simón Martínez Martínez

Subdirector de Estudios de Posgrado

San Nicolás de los Garza, Nuevo León, junio 2016

*A mi hija,
Amelie.*

ÍNDICE GENERAL

Resumen	XI
1. Introducción	1
1.1. Descripción del problema	1
1.2. Hipótesis de estudio	3
1.3. Justificación	3
1.4. Objetivo y alcance	4
1.5. Estructura de la tesis	4
2. Marco teórico	5
2.1. El problema de rutas de vehículos	5
2.1.1. El problema de rutas de vehículos dinámico	7
2.2. El problema de mínima latencia	9
2.3. Metodologías de solución a problemas de ruteo de vehículos	10
2.3.1. Algoritmos exactos	10
2.3.2. Algoritmos heurísticos	11
2.3.3. Algoritmos metaheurísticos	12

2.4. Metodologías de solución al problema de ruteo de vehículos capacitado con objetivo de mínima latencia	14
3. Descripción y modelación del problema	16
3.1. Descripción del problema	16
3.2. Condiciones y supuestos	17
3.3. Modelo matemático	18
3.3.1. Parámetros	18
3.3.2. Variables	19
3.3.3. Objetivo	19
3.3.4. Restricciones	19
4. Metodología de solución	21
4.1. Enfoque de solución periódico	22
4.1.1. Fase constructiva	23
4.1.2. Fase de mejora	25
4.2. Enfoque de solución continuo	30
4.2.1. Algoritmos de inserción de clientes dinámicos	31
5. Experiencia computacional	35
5.1. Instancias	36
5.2. Experimentación con el GVNS	36
5.2.1. Calibración de parámetros	36
5.3. Experimentación con el enfoque periódico	40

5.4. Experimentación con el enfoque continuo	42
6. Conclusiones y trabajo a futuro	46
A. Glosario de términos	48
B. Instancias	49
C. Mejores soluciones	55

ÍNDICE DE FIGURAS

1.	III
4.1. Intercambio de clientes	28
4.2. Relocalización de un cliente (Una ruta)	29
4.3. Relocalización de un cliente (Dos rutas)	29
4.4. 2-opt Intra ruta	30
4.5. 2-opt Inter rutas	30
5.1. Gráfica de efectos principales para Latencia	37
5.2. Gráfica de interacción entre factores para Latencia	38
5.3. CMT. Periódico 4-p	41
5.4. CMT. Periódico 6-p	41
5.5. CMT. Periódico 12-p	41
5.6. Enfoque periódico. CMT1 y CMT2	42
5.7. Enfoque periódico. CMT3 y CMT4	43
5.8. Enfoque periódico. CMT5, CMT11 y CMT12	44
5.9. Algoritmo de inserción 1. Enfoque continuo	45

5.10. Algoritmo de inserción 2. Enfoque continuo	45
B.1. Instancia CMT1	50
B.2. Instancia CMT2	51
B.3. Instancia CMT3	52
B.4. Instancia CMT4	53
B.5. Instancia CMT5	54

ÍNDICE DE TABLAS

5.1. Niveles para factor 1 (VND)	39
5.2. Niveles para factor 2 (Agitación)	39
C.1. Enfoque periódico en instancias CMT. Cuatro periodos	55
C.2. Enfoque periódico en instancias CMT. Seis periodos	56
C.3. Enfoque periódico en instancias CMT. Doce periodos	56
C.4. Enfoque periódico en instancias GWK. Cuatro periodos	57
C.5. Enfoque periódico en instancias GWK. Seis periodos	57
C.6. Enfoque periódico en instancias GWK. Doce periodos	58

RESUMEN

Mayra Alejandra Luna Peña.

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

Universidad Autónoma de Nuevo León.

Facultad de Ingeniería Mecánica y Eléctrica.

Título del estudio:

PROBLEMA DINÁMICO DE RUTAS DE VEHÍCULOS CENTRADO EN EL CLIENTE

Número de páginas: 64.

OBJETIVOS Y MÉTODO DE ESTUDIO: El problema de ruteo de vehículos capacitado con objetivo de mínima latencia (Cumulative Capacitated Vehicle Routing Problem, *CCVRP*) busca minimizar la suma de los tiempos de espera de los clientes. En la variante dinámica (*DCCVRP*), se considera que la totalidad de la información no se conoce al inicio del horizonte de planeación. En esta tesis se estudia el *DCCVRP*, considerando dinamismo en las solicitudes de servicio. De manera que existen dos tipos de clientes, los clientes estáticos son aquellos que se conocen al inicio del periodo de planeación de las rutas, los clientes dinámicos son conocidos a partir del comienzo de la ejecución de las rutas. En el presente trabajo se desarrollan algoritmos basados en metaheurísticas para encontrar soluciones al problema.

CONTRIBUCIONES Y CONCLUSIONES:

- Diseño e implementación de un método heurístico de dos fases para dar solución al problema.
- Diseño e implementación de algoritmos de inserción de clientes para el enfoque de solución continuo del problema.

Firma del asesor: _____
Dra. Ada Margarita Álvarez Socarrás

CAPÍTULO 1

INTRODUCCIÓN

1.1 DESCRIPCIÓN DEL PROBLEMA

El problema que se aborda en esta tesis trata la elaboración de rutas de vehículos con el fin de servir un conjunto de clientes dado. Este problema pertenece a uno de los conjuntos de problemas de optimización combinatoria conocido en la literatura científica como *VRP* (Vehicle Routing Problem).

El problema de rutas de vehículos clásico (*CVRP*, Capacitated Vehicle Routing Problem) consiste en diseñar un conjunto de rutas para una flota de vehículos dada, con el fin de dar servicio a un conjunto de clientes los cuales tienen una ubicación geográfica determinada. Se considera que los vehículos tienen capacidad limitada, por lo que se debe satisfacer la demanda de todos los clientes respetando la restricción de capacidad de los vehículos al elaborar las rutas. Además, cada cliente debe ser visitado por un solo vehículo. Se asume que la información de los vehículos, clientes, ubicaciones y demandas se conoce de antemano.

El objetivo del *CVRP* es encontrar el conjunto de rutas con las características citadas cuya ejecución derive en el mínimo costo posible. Esto muchas veces se traduce en la menor distancia recorrida posible, ya que el costo de transportación está directamente relacionado con la distancia recorrida. Esta tesis estudia una variante de dicho problema.

En muchas ocasiones, la distancia recorrida por los vehículos no es la prioridad en la planeación de las rutas, sino la rapidez con la que se atiende a los clientes. En

casos de emergencia, los clientes deben ser atendidos lo más pronto posible quedando en segundo término los costos de transportación; hoy en día, la competencia en el mercado obliga a las empresas a elevar sus estándares en la rapidez del servicio al cliente para seguir siendo preferidas por los consumidores, de manera que se debe pagar el precio de un mayor costo de transportación para obtener un mayor nivel de satisfacción de los clientes. En ambos casos el objetivo del problema de rutas se centra en el cliente.

Dicho objetivo se relaciona con lo que en la literatura se conoce como latencia. La latencia de una ruta es la suma de los tiempos de espera de todos los clientes que están asignados a dicha ruta, siendo la latencia de una solución la suma de las latencias de todas las rutas que la forman. El problema de rutas de vehículos centrado en el cliente busca minimizar la latencia de la solución, y es el caso de nuestro problema de estudio. Es conocido en la literatura como *CCVRP* (Cumulative Capacitated Vehicle Routing Problem).

En los problemas de optimización en general (de rutas en particular), se busca modelar problemas reales de la manera más exacta posible, para esto se añaden características o condiciones en el planteamiento del problema que describan la realidad. Una característica importante en los problemas de rutas que ha sido estudiada en años recientes es la adición de elementos dinámicos al problema, esta variante se conoce como *DVRP* (Dynamic Vehicle Routing Problem).

Debido a los avances tecnológicos actuales, es posible tener información de la situación de los vehículos de una flota a lo largo de su trayectoria y establecer contacto con el vehículo en cualquier momento, permitiendo que las rutas sean modificadas aun después de iniciar su recorrido. Gracias a esto se pueden resolver problemas dinámicos en los que la totalidad de la información no se conoce al momento de la planeación, siendo más bien revelada a lo largo del día.

En nuestro problema, el dinamismo se presenta en las solicitudes de los clientes. Se conoce la información de un conjunto de clientes al momento de planear las rutas, los cuales llamamos clientes estáticos, y a lo largo del día nuevas solicitudes de

servicio ingresan al sistema, siendo hasta ese momento cuando la información del nuevo cliente es conocida, a estos clientes los llamamos clientes dinámicos.

A lo largo del presente documento, llamaremos al problema tratado en el mismo *DCCVRP* (Dynamic Cumulative Capacitated Vehicle Routing Problem). Dada su complejidad, este problema no puede ser resuelto con métodos exactos en un tiempo razonable para problemas de tamaño real, por lo que se han desarrollado técnicas heurísticas para encontrar soluciones al problema.

Dos enfoques de solución han sido implementados. El enfoque de solución periódico se basa en dividir el horizonte de planeación en periodos de igual duración y en cada periodo se resuelve el problema de rutas estático con los clientes que se conocen hasta el momento. En el enfoque de solución dinámico cada vez que se conoce una nueva solicitud de servicio, se intenta asignar a alguna de las rutas de la solución actual mediante un algoritmo de inserción eficiente.

1.2 HIPÓTESIS DE ESTUDIO

Una heurística de dos fases que incorpora el método metaheurístico Búsqueda por Entorno Variable permite obtener buenas soluciones del problema.

1.3 JUSTIFICACIÓN

El objetivo de mínima latencia ha sido tratado en la literatura para el caso estático, en el cual toda la información se asume conocida; mientras que el dinamismo ha sido estudiado en problemas con objetivo de minimizar costo. Sin embargo, en la revisión literaria no hemos encontrado trabajos que traten dinamismo y mínima latencia al mismo tiempo, a pesar de la notable posibilidad de aplicaciones de ambas situaciones.

1.4 OBJETIVO Y ALCANCE

El objetivo de esta tesis es proponer una metodología para encontrar solución a problemas de tamaño real de manera rápida. Para esto se consideran los siguientes objetivos particulares:

1. Revisar literatura de problemas similares
2. Diseñar algoritmos de solución para el problema basados en estrategias metaheurísticas.
3. Implementar computacionalmente dichos algoritmos.
4. Evaluar el desempeño de los algoritmos mediante experimentación computacional.

1.5 ESTRUCTURA DE LA TESIS

La estructura de este documento es la siguiente: en el capítulo dos se describen los conceptos necesarios para el estudio del problema, así como algunos métodos de solución usados en la literatura para problemas similares al de la tesis. En el capítulo tres se presenta el planteamiento del problema así como un modelo matemático encontrado en la literatura para la versión estática del problema de ruteo de vehículos con objetivo de mínima latencia. En el capítulo cuatro se describe a detalle la metodología propuesta para dar solución al problema. En el capítulo cinco se muestran los resultados de los experimentos computacionales realizados para evaluar el desempeño del algoritmo. Finalmente en el capítulo seis se presentan las conclusiones y propuestas de trabajo a futuro.

CAPÍTULO 2

MARCO TEÓRICO

En este capítulo se presentan algunas definiciones que se necesitan para entender el problema que se aborda en la tesis, así como las características generales de los problemas de ruteo de vehículos. También se exponen algunos de los métodos de solución que se encuentran comúnmente en la literatura científica.

En la primera sección se presentan los conceptos básicos sobre el problema de ruteo de vehículos y una descripción de algunas variantes conocidas de los mismos. En la sección 2 se explica el problema de latencia mínima, ya que la latencia es la función objetivo del problema de ruteo tratado en este trabajo. Por último la sección 3 está dedicada a la descripción de los principales métodos de solución reportados en la literatura, métodos exactos, técnicas heurísticas y metaheurísticas.

2.1 EL PROBLEMA DE RUTAS DE VEHÍCULOS

El problema de ruteo de vehículos es uno de los problemas de optimización combinatoria más importantes y estudiados. Consiste en diseñar las rutas óptimas para la distribución de productos desde un centro de distribución hasta un número determinado de clientes dispersos geográficamente, utilizando para ello una flota de vehículos con cierta capacidad de transportación. Cada ruta inicia y termina en el depósito, cada cliente debe ser visitado exactamente una vez para satisfacer su demanda y además la suma de las demanda de los clientes visitados en cada ruta no excede la capacidad del vehículo asignado a dicha ruta. El objetivo clásico es minimizar el costo total de transportación.

La formulación de este problema fue introducida por Dantzig y Ramser [14], como una generalización del problema del agente viajero (Travelling Salesman Problem, *TSP*) presentado por Flood [16]. El VRP es generalmente definido en un grafo $G=(V, E, C)$ siendo $V = \{v_0, \dots, v_n\}$ el conjunto de vértices, $E = \{(v_i, v_j) \text{ tal que } (i, j) \in V, i \neq j\}$ el conjunto de arcos y $C = (c_{ij})$ una matriz de costos definida en E , representa distancias, tiempos de traslado o costos de traslado. El vértice v_0 es llamado depósito, mientras que los vértices restantes representan clientes que necesitan ser servidos.

Un gran número de variantes del VRP han sido estudiadas. Entre las más comunes están las siguientes:

- Problema de ruteo de vehículos con ventanas de tiempo (Vehicle Routing Problem with Time Windows, *VRP-TW*), en el cual se requiere que los clientes sean visitados durante un intervalo específico de tiempo [10].
- Problema de ruteo de vehículos con restricción de distancia, donde existe un límite máximo para la longitud o tiempo de traslado de las rutas [28].
- Problema de ruteo de vehículos con entrega y recolección (Vehicle Routing Problem with Pickup and Delivery, *VRP-PD* o *PDP*), donde los bienes tienen que ser entregados o recolectados en cantidades específicas para cada cliente [51].
- Problema de ruteo de vehículos abierto (*open VRP*), en el cual los vehículos no necesariamente regresan al depósito después de terminar su ruta [33].
- Problema periódico (*periodic VRP*), donde los clientes deben ser visitados varias veces durante el horizonte de planeación [17].
- Problema con entregas divididas (*split-delivery VRP*), donde cada cliente puede ser visitado por varios vehículos [2].
- Problema de ruteo de vehículos con múltiples depósitos (multi-depot Vehicle Routing Problem, *MDVRP*), donde existen varios depósitos ha donde pueden acudir los clientes [49].

- Problema de ruteo de vehículos con flota heterogénea (Heterogeneous fleet Vehicle Routing Problem, *HVRP*), donde los vehículos tienen diferentes capacidades [47].

Referimos a [43] para consultar el estado del arte sobre problemas de ruteo de vehículos.

2.1.1 EL PROBLEMA DE RUTAS DE VEHÍCULOS DINÁMICO

Un problema de ruteo se dice que es estático cuando todos los datos de entrada del problema son conocidos antes de que se construyan las rutas. En un problema dinámico, algunos de los datos se revelan o actualizan durante el periodo de tiempo en el que las operaciones se llevan a cabo.

Las aplicaciones del mundo real frecuentemente incluyen esta característica dinámica.

El problema de ruteo de vehículos dinámico (Dynamic Vehicle Routing Problem, *DVRP*) puede considerarse como una serie de instancias del problema estático. Al igual que el VRP estático, pertenece a la clase de problemas NP-duros. *BUSCAR REFERENCIA SOBRE LA COMPLEJIDAD DEL DVRP* La primera referencia a un VRP dinámico fue de Wilson y Colvin [56]. Estudiaron un (Dial A Ride Problem, *DARP*) con un solo vehículo, en el que las peticiones de los clientes, viajar desde un origen a un destino, aparecen de forma dinámica.

La fuente más común de dinamismo es la llegada de clientes durante la operación de las rutas. También pueden ser considerados dinámicos los tiempos de traslado, las demandas de los clientes o la disponibilidad de los vehículos. En este trabajo consideramos un problema de ruteo de vehículos con demandas dinámicas.

Para dar solución a los problemas dinámicos de ruteo de vehículos se distinguen dos enfoques de solución en la literatura, uno basado en re-optimización periódica y otro basado en re-optimización continua. En ambos casos, al inicio del horizonte de planeación, se genera una solución inicial que contiene al conjunto de todos los clientes estáticos.

La re-optimización periódica consiste en resolver el problema estático correspondiente a diferentes estados del sistema, a lo largo del tiempo. El inconveniente de este enfoque es que el tiempo dedicado a cada problema estático no se conoce con anticipación, por lo que la optimización puede ser detenida antes de que se haya alcanzado una buena solución. Por ello se debe tener un algoritmo de optimización rápido y eficiente.

Montemanni et al. [40] utilizan el enfoque periódico, resuelven el DCVRP (Dynamic Capacitated Vehicle Routing Problem), mediante un algoritmo basado en Colonia de Hormigas (Ant Colony System Algorithm).

Por otro lado, en el enfoque de reoptimización continua, cada vez que llegan una o varias solicitudes al sistema, se busca asignar dichas solicitudes a las rutas actuales. Por ello, es necesario que la inserción de los nuevos clientes y la mejora de las rutas se realicen en el menor tiempo posible. En los trabajos de Hong [24] y Chen y Xu [7] se utiliza el enfoque continuo para resolver el problema dinámico de ruteo de vehículos con ventanas de tiempo.

Ninguno de estos trabajos dinámicos considera la latencia como objetivo de optimización, sino la minimización de los costos de operación de las rutas.

Para una revisión del estado del arte de los problemas de ruteos de vehículos dinámicos se puede consultar [20], [25], [31].

En algunos problemas dinámicos se conoce más información sobre el futuro que en otros, por esto se han definido medidas del grado de dinamismo. Lund et al. [36] han definido el grado de dinamismo en un problema de ruteo como la relación entre el número de peticiones dinámicas y el número total de solicitudes. Larsen [30] define otra medida que tiene en cuenta el tiempo en el que se conocen las solicitudes dinámicas, llamado grado efectivo de dinamismo.

2.2 EL PROBLEMA DE MÍNIMA LATENCIA

En los problemas de ruteo mencionados previamente, el objetivo es la minimización de la distancia (costo total) recorrida por los vehículos. Sin embargo, en muchas aplicaciones el interés es minimizar el tiempo de espera de los clientes. Esa función objetivo está relacionada con lo que se denomina latencia de un nodo de un grafo.

Dado un conjunto de nodos (clientes), una matriz de costos, y una ruta que visita todos los nodos en algún orden, la latencia de un nodo es la distancia o tiempo necesario para llegar desde el depósito hasta dicho cliente, mientras que la latencia total de la ruta es la suma de todas las latencias de los clientes.

El problema del repartidor (Travelling Deliveryman Problem, *TDP*), el problema del reparador (Travelling Repairman Problem, *TRP*), el problema de minimizar el tiempo total de terminación de un trabajo o el tiempo total de flujo de trabajos que serán procesados en una sola máquina, son problemas que pueden ser tratados como problemas de mínima latencia (Minimum Latency Problem, *MLP*). En el problema del reparador y del repartidor, la localización de los clientes así como los tiempos de servicio del reparador (repartidor) en cada cliente son conocidos. En ambos casos, el agente tiene que visitar a los clientes para realizar algún servicio y el objetivo es encontrar una ruta que minimice el tiempo total de espera de los clientes. Estos problemas pueden ser vistos como problemas de ruteo centrados en el cliente debido a que el objetivo es minimizar el tiempo de espera de los clientes en lugar del tiempo total de viaje del vehículo.[1]

Recientemente, algunas variaciones interesantes donde el MLP se combina con otro problema, han sido estudiadas. Levin y Penn [34] estudian una combinación de MLP y secuenciación de máquinas donde n tareas deben ser procesadas en una sola máquina localizada en una planta y posteriormente son enviadas a n clientes por un solo vehículo. Existen tiempos de procesamiento para los trabajos y tiempos de traslado entre la planta y los clientes. El objetivo es determinar una secuencia de producción para los trabajos en la máquina y determinar el ruteo del vehículo de

manera tal que la suma de los tiempos de entrega de los trabajos a los clientes se minimice. Se permite que el vehículo recoja trabajos de la planta varias veces. Li, Vairaktarakis, y Lee [35] consideran una variación del mismo problema donde varios trabajos pueden ser solicitados al mismo cliente y el vehículo puede o no tener restricciones de capacidad.

Algunos artículos que reflejan el estado del arte son Blum et al [3], Wu et al [57], Silva et al [52], entre otros.

2.3 METODOLOGÍAS DE SOLUCIÓN A PROBLEMAS DE RUTEO DE VEHÍCULOS

Como se ha mostrado en la sección 2.1, existe una gran familia de problemas de ruteo de vehículos que tienen una estructura en común, por lo que se desean métodos de solución que produzcan resultados precisos en tiempos cortos de ejecución, pero también se busca que sean fáciles de entender y con pocos parámetros.

En esta sección se mencionan algunas de las metodologías reportadas en la literatura para dar solución a los problemas de ruteo de vehículos estáticos. Entre ellas tenemos métodos exactos, heurísticas y metaheurísticas.

2.3.1 ALGORITMOS EXACTOS

Los métodos exactos aplicados al CVRP son principalmente los métodos de ramificación y acotamiento y los de ramificación y corte.

- Ramificación y acotamiento. Hasta finales de los 80's los métodos más efectivos eran los algoritmos de ramificación y acotamiento basados en relajaciones combinatorias. En los últimos tiempos se han propuesto relajaciones lagrangianas, las cuales han podido incrementar el tamaño de los problemas que se pueden resolver. Los algoritmos de este tipo se concentran principalmente en determinar una buena cota inferior. Algunas referencias de este tema pueden encontrarse en [12], [15] y [54].

- Ramificación y corte. Los algoritmos de ramificación y corte tratan de generar desigualdades válidas para añadir planos de corte en el espacio factible de soluciones, tratando de acotar el árbol de ramificación no sólo en el nodo raíz. Estos algoritmos constituyen los mejores algoritmos exactos hoy en día para el problema de ruteo de vehículos. También se ha trabajado la combinación de algoritmos de ramificación y corte con otros algoritmos. Fukasawa et al. [40] proponen un método de solución que combina ramificación y corte con un algoritmo de generación de columnas (Branch and cut and Price, BCP). La idea de combinar columnas y generación de cortes es mejorar las cotas inferiores en el proceso de ramificación.

2.3.2 ALGORITMOS HEURÍSTICOS

Dado que el VRP es un problema NP-duro [32], los métodos exactos resuelven instancias de tamaño limitado. Por ello, heurísticas y metaheurísticas son usadas en la mayoría de las aplicaciones prácticas donde varios cientos de clientes son visitados diariamente. En cuanto a las técnicas heurísticas, Laporte y Semet [29] las clasifican en estos tres grupos:

- i. Métodos constructivos
- ii. Métodos de dos fases
- iii. Heurísticas de mejora

Los métodos constructivos combinan rutas existentes usando criterios de ahorro como la heurística de Clarke y Wright [9], o añaden de manera secuencial clientes a las rutas usando un criterio de inserción como la heurística de Mole y Jameson [39].

Los métodos de dos fases descomponen el VRP en asignación de clientes a rutas y ordenamiento de clientes dentro de las rutas. En los métodos de agrupación primero, ruteo después (Cluster-first, route-second) los clientes son agrupados en rutas que serán servidas por un solo vehículo, posteriormente resuelven un TSP para cada ruta. Por otra parte, las heurísticas de ruteo primero, agrupamiento después (Route-first,

cluster-second), primero construyen un circuito TSP visitando todos los clientes y después lo dividen en rutas factibles.

Las heurísticas de mejora aplican procedimientos de búsqueda local simples para explorar el vecindario de una solución. Cualquier heurística de mejora para el TSP puede ser usada para el VRP, como los clásicos procedimientos Or opt [42], 2-opt [16] y 3-opt [13]. Para una revisión de las búsquedas locales para problemas de ruteo de vehículos se recomienda leer Funke et al. [18].

2.3.3 ALGORITMOS METAHEURÍSTICOS

Las heurísticas realizan una exploración relativamente limitada del espacio de soluciones, proporcionando resultados buenos con tiempos de cómputo moderados. Sin embargo los resultados son de mejor calidad al utilizar técnicas metaheurísticas, las cuales han sido muy utilizadas en años recientes. Las metaheurísticas son procedimientos heurísticos de alto nivel diseñados para guiar otros métodos o procesos con el fin de encontrar buenas soluciones a problemas de optimización matemática. A diferencia de las heurísticas que terminan una vez que se ha alcanzado un óptimo local, estos métodos buscan salir de óptimos locales explorando un gran conjunto del espacio de solución con la esperanza de encontrar una solución más cercana a la óptima [53].

Algunos de los tipos fundamentales son las metaheurísticas para los métodos de relajación, las metaheurísticas para los procesos constructivos, las metaheurísticas para las búsquedas por entornos y las metaheurísticas para los procedimientos evolutivos [55].

- Las metaheurísticas de relajación se refieren a procedimientos de resolución de problemas que utilizan relajaciones del modelo original (es decir, modificaciones del modelo que hacen al problema más fácil de resolver), cuya solución facilita la solución del problema original.
- Las metaheurísticas constructivas se orientan a los procedimientos que tratan de la obtención de una solución a partir del análisis y selección paulatina de

las componentes que la forman.

- Las metaheurísticas de búsqueda guían los procedimientos que usan transformaciones o movimientos para recorrer el espacio de soluciones alternativas y explotar las estructuras de entornos asociadas.
- Las metaheurísticas evolutivas están enfocadas a los procedimientos basados en conjuntos de soluciones que evolucionan sobre el espacio de soluciones.

Es usual que se combinen componentes de diferentes metaheurísticas dando lugar a procedimientos híbridos, los cuales son buenos métodos de solución para diversos problemas de optimización [4]. Hay un gran número de métodos basados en metaheurísticos para la solución del VRP, entre ellos, GRASP (Greedy Randomized Adaptive Search Procedure) [46], algoritmos meméticos [45], colonias de hormigas [48], búsqueda tabu [11], búsqueda por grandes vecindarios (Adaptive Large Neighborhood Search, *ALNS*). En Gendreau et al. [19] podemos encontrar un estudio de metaheurísticas diseñadas para dar solución al CVRP.

La metodología implementada para dar solución al problema de esta tesis está basada en la metaheurística Búsqueda por Entorno Variable, por consiguiente se profundiza un poco más en la explicación de la misma.

BÚSQUEDA DE ENTORNO VARIABLE

La Búsqueda de Entorno Variable es una metaheurística propuesta por N. Mladenovic[38], se basa en cambiar sistemáticamente de estructura de entornos dentro de la búsqueda para escapar de óptimos locales. Se han realizado muchas extensiones, principalmente para permitir la solución de problemas de gran tamaño. En la mayoría de ellas, se ha hecho un esfuerzo por mantener la simplicidad del esquema básico, el cual consiste de tres componentes:

- Perturbación de la solución incumbente. El objetivo de este componente es extender el espacio de búsqueda para reducir la posibilidad de que el algoritmo se estanque en una solución óptima local.

- Búsqueda local. Define la calidad de la solución final, siendo el proceso que consume más tiempo en el VNS.
- Cambio de vecindario.

Gran cantidad de trabajos en la literatura sobre VRP han sido tratados mediante métodos basados en VNS, [5], [44], [27], [21], [23].

La versión determinística de la Búsqueda de Entorno Variable es llamada búsqueda de entorno variable descendente (Variable Neighbourhood Descent, VND), la cual realiza un cambio de estructura de entorno de forma determinística cada vez que se llega a un mínimo local. En cambio, la búsqueda de entorno variable básica (Basic Variable Neighbourhood Search, BVNS) combina cambios determinísticos y aleatorios de estructura de entornos.

La versión que se utiliza en esta tesis es la búsqueda por entorno variable general (General Variable Neighbourhood Search, GVNS). Es una generalización de la BVNS, en donde el proceso de búsqueda local se realiza mediante una VND.

2.4 METODOLOGÍAS DE SOLUCIÓN AL PROBLEMA DE RUTEO DE VEHÍCULOS CAPACITADO CON OBJETIVO DE MÍNIMA LATENCIA

El problema de ruteo de vehículos capacitado con objetivo de latencia (Cumulative Capacited Vehicle Routing Problem, CCVRP) es una variante relativamente nueva del VRP su primera referencia literaria (2010) se encuentra en [41], ha sido estudiado recientemente en varios artículos, los cuales son comentados en esta sección.

Ngueveu et al. [41] presentan un modelo matemático y varias propiedades del problema, usando estas características, los autores proponen cotas superiores e inferiores. Presentan además dos algoritmos meméticos. El primer individuo de la población es creado mediante la heurística del vecino más cercano, el segundo individuo se genera mediante el uso de una búsqueda local aplicada al primer individuo. El resto de la

población se se obtiene mediante diferentes operadores de cruza aplicando búsquedas locales de acuerdo a una regla probabilística.

Ribeiro y Laporte [50] proponen un heurística ALNS, la cual usa operadores de destrucción y reparación para generar nuevas soluciones. Una regla probabilista adaptativa es usada para seleccionar los operadores de destrucción y reparación más prometedores, actuando así el algoritmo en base a la experiencia. Su algoritmo permite infactibilidad en las soluciones, incorporando un procedimiento de penalización para las soluciones infactibles.

Chen et al. [6] proponen un algoritmo basado en Búsqueda Local Iterada (Iterated Local Search, *ILS*).

Ke y Feng [26] crean un algoritmo de dos fases, la primera busca hacer una agrupar los clientes de manera inteligente, mientras que la segunda fase se enfoca en minimizar la latencia de cada ruta.

Hasta donde conocemos, el único algoritmo exacto publicado para el CCVRP es el (Branch and Cut and Price, BCP) creado por Jens Lygaard y Sanne Wøhlk [37].

CAPÍTULO 3

DESCRIPCIÓN Y MODELACIÓN DEL PROBLEMA

En este capítulo se describen de manera detallada las características del problema bajo estudio. Como se mencionó en el capítulo anterior, el DCCVRP puede ser considerado, desde un enfoque de re-optimización periódica, como una serie de problemas de tipo CCVRP, por lo que en este capítulo presentamos el modelo matemático de este último problema.

3.1 DESCRIPCIÓN DEL PROBLEMA

El problema consiste en determinar un diseño de rutas para la transportación y entrega de bienes a través de una red de distribución definida por un grafo completo no dirigido $G = (V, E)$, donde $V = \{0, 1, \dots, n, n + 1\}$ es el conjunto de nodos, los nodos 0 y $n + 1$ corresponden al centro de distribución, cada nodo restante $i \in V$ representa un cliente que debe ser visitado.

E es el conjunto de aristas, cada arista del grafo $(i, j) \in E$ representa el camino desde el nodo i hasta el nodo j . A cada arista se asocia un valor positivo w_{ij} que representa el tiempo de traslado de un vehículo desde el nodo i hasta el nodo j .

A cada nodo i está asociado un valor positivo q_i que representa su demanda, se asocia también un valor no negativo ts_i que representa el tiempo necesario para que un vehículo cumpla con la entrega del producto, y un valor no negativo ta_i que representa el tiempo de llegada del cliente al sistema. Los clientes que son conocidos

al inicio del período de planeación se denominan clientes estáticos, mientras que aquellos que aparecen en el transcurso de la ejecución de las rutas, son llamados clientes dinámicos.

Se cuenta con un conjunto de R vehículos idénticos con capacidad limitada Q , los cuales visitarán al conjunto de clientes.

El objetivo es encontrar el conjunto de rutas que inician y terminan en la misma ubicación (depósito), tales que, visitan a todos los clientes exactamente una vez para satisfacer su demanda, y se minimiza la suma de los tiempos de espera de los clientes para ser atendidos.

3.2 CONDICIONES Y SUPUESTOS

El problema estudiado en esta tesis tiene las siguientes características:

- Depósito único: Todos los vehículos inician y terminan en la misma ubicación.
- Capacidad limitada: Los vehículos tienen capacidad de entrega restringida.
- Vehículos homogéneos: Los vehículos tienen todos la misma capacidad.
- Entregas completas: El vehículo debe satisfacer en su totalidad la demanda del cliente que visita.
- Solo entrega: Solo se considera entrega de mercancía/servicios, no recolección.
- Se considera solo un tipo de producto
- Clientes dinámicos: La totalidad de clientes no es conocida al inicio del horizonte de planeación, sino que diversos clientes aparecen a lo largo de la jornada.
- Los tiempos de traslado entre clientes y entre los clientes y el depósito son conocidos.
- El tiempo de servicio en cada cliente es conocido.
- La demanda de cada cliente es conocida al momento de ingresar el cliente al sistema.

- Cada vez que llega un cliente dinámico, se requiere actualizar la información de los clientes que ya han sido servidos, los clientes que aún no han sido atendidos, y la ubicación de los vehículos en ese momento para poder modificar las rutas originales.
- Se debe respetar la restricción de capacidad de cada vehículo, es decir, la suma de las demandas de los clientes que son visitados por un vehículo dado no debe ser superior a la capacidad de dicho vehículo.

Se asume que existe comunicación entre el conductor y la base de operación, que no es posible usar datos estadísticos para inferir información sobre los clientes dinámicos, y que los tiempos de traslado son constantes a lo largo del horizonte de planeación.

3.3 MODELO MATEMÁTICO

Dada la naturaleza dinámica del problema, un modelo matemático (que no considere variables estocásticas) no puede ser usado para encontrar soluciones en tiempo real. Pero consideramos pertinente, para fines de una mejor comprensión del problema estudiado en esta tesis, mostrar el modelo que representa al CCVRP. El modelo que a continuación se describe se basa en el presentado por Ngueveu et al [41].

3.3.1 PARÁMETROS

n : Número de clientes.

q_i : Demanda del cliente i .

w_{ij} : Tiempo necesario por un vehículo para ir del cliente i al cliente j .

t_{si} : Tiempo de servicio del cliente i .

Q : Capacidad de los vehículos.

M : Número real positivo de gran tamaño.

T : Número de períodos de tiempo.

3.3.2 VARIABLES

t_i^k : tiempo de llegada del vehículo k al cliente i .

$$x_{ij}^k = \begin{cases} 1 & \text{si el vehículo } k \text{ se mueve directamente del cliente } i \text{ al cliente } j \\ 0 & \text{de otro modo} \end{cases} \quad (3.1)$$

3.3.3 OBJETIVO

$$\min \sum_{k=1}^R \sum_{i \in V'} t_i^k \quad (3.2)$$

El objetivo (3.2) es minimizar la suma de los tiempos de espera de cada uno de los clientes.

3.3.4 RESTRICCIONES

$$\sum_{i \in V} x_{ij}^k = \sum_{i \in V} x_{ji}^k, \quad \forall j \in V', \quad \forall k \in \{1, \dots, R\} \quad (3.3)$$

$$\sum_{k \in \{1, \dots, R\}} \sum_{j \in V} x_{ij}^k = 1, \quad \forall i \in V' \quad (3.4)$$

$$\sum_{i \in V'} \sum_{j \in V} x_{ij}^k q_i \leq Q, \quad \forall k \in \{1, \dots, R\} \quad (3.5)$$

$$\sum_{j \in V} x_{0j}^k = 1, \quad \forall k \in \{1, \dots, R\} \quad (3.6)$$

$$\sum_{j \in V} x_{i,n+1}^k = 1, \quad \forall k \in \{1, \dots, R\} \quad (3.7)$$

$$t_i^k + w_{ij} - (1 - x_{ij}^k)M \leq t_j^k, \quad \forall i \in V \setminus \{n+1\}, \quad \forall j \in V, \quad \forall k \in \{1, \dots, R\} \quad (3.8)$$

$$x_{ij}^k \in \{0, 1\}, \quad \forall i \in V, \quad \forall j \in V, \quad i \neq j, \quad \forall k \in \{1, \dots, R\} \quad (3.9)$$

Las restricciones (3.3) obligan a que cada vehículo que llega a un cliente debe salir de él. Las restricciones (3.4) garantizan que cada cliente sea servido unicamente por

un vehículo. Las restricciones (3.5) limitan la cantidad de clientes que se pueden asignar a cada vehículo, de manera que la suma de las demandas de los clientes asignados a un vehículo no puede ser mayor a la capacidad del mismo. Las restricciones (3.6) asignan al depósito como el lugar donde inicia cada ruta mientras que las restricciones (3.7) asignan al depósito como el lugar donde concluyen todas las rutas. Las restricciones (3.8) sirven para prevenir la formación de subtours.

CAPÍTULO 4

METODOLOGÍA DE SOLUCIÓN

El problema dinámico de rutas de vehículos puede considerarse como una generalización del problema estático, descomponiendo el primero en una serie de instancias del problema estático. Sabemos que el tiempo computacional necesario para encontrar soluciones óptimas de la versión estática del problema tratado en esta tesis puede llegar a ser muy grande. En general, los métodos exactos son efectivos para resolver problemas con menos de 200 clientes. Como podemos ver en Lysgaard y Wøhlk [41], el tiempo necesario para resolver un problema con 151 nodos es casi de siete horas. Dado que el problema dinámico involucra la reoptimización de varios problemas estáticos, el tiempo computacional requerido para obtener soluciones exactas puede ser demasiado grande. Esto justifica desarrollar métodos basados en metaheurísticas, que generen soluciones de buena calidad en un tiempo razonable.

En esta tesis tratamos el problema mediante los dos enfoques de solución para el DVRP que se encuentran en la literatura: enfoque de solución periódico y enfoque de solución continuo. En ambos casos se propone un algoritmo heurístico de dos fases, en la primera fase se construye una solución y en la segunda fase se mejora dicha solución mediante un algoritmo basado en la metaheurística Búsqueda por Entorno Variable. Adicionalmente, para el enfoque continuo se implementaron dos algoritmos de inserción de clientes.

En esta sección se describen los algoritmos propuestos.

4.1 ENFOQUE DE SOLUCIÓN PERIÓDICO

En el enfoque de re-optimización periódica, se considera al DVRP como una extensión del VRP estático descomponiendo el primero en una serie de instancias del segundo. Para esto se define una partición del horizonte de planeación en segmentos de tiempo con igual duración.

Al inicio de cada segmento de tiempo se resuelve el problema estático con los clientes que se conocen hasta dicho momento usando el algoritmo de optimización deseado. En nuestro caso utilizaremos un método de dos fases, la primera fase construye la solución inicial, la segunda fase intenta mejorar dicha solución.

En el Algoritmo 1 se muestra el esquema general de la metodología de solución del enfoque periódico.

Algoritmo 1 Enfoque periódico

Entrada: Conjunto de clientes estáticos, conjunto de entornos que se usarán en la GVNS-CCVRP.

Salida: Solución incumbente, x .

- 1: $x = \text{Constructivo1}()$;
- 2: $\text{GVNS-CCVRP}(x)$;
- 3: **para** $i = 1$; $i < \text{numPeriodos}$; $i++$ **hacer**
- 4: $x = \text{Constructivo2}()$;
- 5: Mejora(x);
- 6: **fin para**

Note que en el esquema mostrado se distingue entre dos tipos de algoritmos constructivos ($\text{Constructivo1}()$ y $\text{Constructivo2}()$), esto es debido a que en el primer periodo no existe solución alguna, mientras que en los periodos posteriores se tiene la solución del periodo anterior y a partir de ella se realiza una reconstrucción agregando los nuevos clientes que aparecieron a lo largo del periodo de tiempo anterior. De manera que la forma de construir la solución en ambos casos es diferente. En la siguiente sección se explican a detalle los dos métodos constructivos utilizados.

4.1.1 FASE CONSTRUCTIVA

En esta fase se requiere construir una solución al inicio de cada período de tiempo, se pueden diferenciar dos variantes: i) la construcción al inicio del horizonte de planeación (periodo $p = 0$), cuando aun no existen rutas creadas, y los clientes conocidos son todos los clientes estáticos, ii) la construcción al inicio de cada periodo de tiempo p ($p \neq 0$), para la cual se utiliza la solución generada en el periodo anterior y a partir de ella se reconstruye agregando a los clientes dinámicos que llegaron en el transcurso del periodo $p - 1$.

El Algoritmo 2 presenta el proceso de construcción al inicio del proceso de planeación de rutas, a este algoritmo le llamaremos Constructivo 1. En el Algoritmo 3 se muestra el proceso de construcción para los periodos de tiempo en los que se divide el horizonte de planeación, llamaremos a este proceso Constructivo 2.

PERIODO INICIAL, $p = 0$

Algoritmo 2 Constructivo 1

- 1: AbrirRutas();
 - 2: AgregarPrimerCliente();
 - 3: **mientras** hay clientes sin asignar y hay rutas con capacidad disponible **hacer**
 - 4: RutaAleatoria();
 - 5: AgregarUltimoCliente();
 - 6: **fin mientras**
-

La construcción de la solución inicial al inicio del horizonte de planeación comienza abriendo k rutas (línea 1), donde k es el menor número de rutas cuya suma de capacidades es mayor o igual a la suma de demandas de los clientes conocidos. A cada una de las rutas se le asigna un primer cliente (línea 2), siendo aquel más cercano al depósito que no ha sido asignado aún a ruta alguna.

Una vez que las rutas han sido inicializadas procedemos a insertar los clientes restantes. Seleccionamos de manera aleatoria una ruta (línea 4) con capacidad restante

mayor o igual que cero la cual será nuestra ruta actual, entre los clientes no asignados aún seleccionamos el cliente más cercano al último cliente de la ruta actual que además respete la restricción de capacidad de la ruta. Este cliente se asigna a la ruta actual como último cliente de la misma (línea 5).

El anterior proceso de inserción de clientes se realiza de manera iterativa hasta que no haya clientes sin asignar o no haya rutas con capacidad suficiente para asignar los clientes restantes. Si lo último sucede no se añaden nuevas rutas, en lugar de eso se rechazan las solicitudes de atención de los clientes no asignados.

PERIODOS POSTERIORES, $p \neq 0$

Algoritmo 3 Constructivo 2

- 1: ActualizarDatos();
 - 2: **mientras** hay clientes sin asignar y hay rutas con capacidad disponible **hacer**
 - 3: AgregarDinamico();
 - 4: **fin mientras**
 - 5: **si** hay clientes sin asignar y no hay suficiente capacidad restante en rutas **entonces**
 - 6: RechazarCliente();
 - 7: **fin si**
-

El algoritmo para construir la solución inicial en cada período de tiempo posterior al inicio de la ejecución de las rutas, parte de la solución obtenida después de la fase de mejora del período anterior. El primer paso es actualizar los datos de la solución actual (línea 1). Para esto se debe determinar cual es el cliente actual de cada una de las rutas, es decir, el próximo cliente a atender o cliente que está siendo atendido en el instante de tiempo en el que nos encontramos; y el estatus (visitado o no visitado) de todos los clientes de la solución hasta el momento.

Enseguida se insertan los clientes dinámicos que arribaron a lo largo del período anterior (línea 3), para esto, se ordenan los clientes dinámicos por orden de aparición y en ese orden se intenta insertar cada uno como se explica a continuación.

Se buscan todas las rutas con capacidad restante suficiente para atender al cliente dinámico actual, en cada una de ellas se calcula el cambio en la latencia al insertar el cliente en cada una de las posiciones posibles (posteriores al cliente actual de la ruta), al final el cliente se inserta en la posición de la ruta que genera el menor cambio en la latencia (en su mejor inserción). Si no existe ruta con capacidad restante suficiente para atender la solicitud del cliente dinámico esta se rechaza (línea 6).

4.1.2 FASE DE MEJORA

La segunda fase del algoritmo implementado tiene la finalidad de mejorar la solución construida. Como se ha dicho, es un algoritmo basado en la versión general de la metaheurística Búsqueda por Entorno Variable (*GVNS*, General Variable Neighborhood Search), en adelante lo llamaremos *GVNS-CCVRP*.

En el Algoritmo 4 se muestra el proceso de la *GVNS*, posteriormente se explica cómo se realizaron cada uno de los pasos del algoritmo para la implementación en este trabajo.

Algoritmo 4 VNS General

Entrada: Solución inicial, x ; conjunto de entornos para la Agitación, N_k ; y conjunto de entornos para la Mejora, N_j .

Salida: Solución incumbente, x .

```
1:  $it = 1$ 
2: mientras  $it < n$  hacer
3:    $k = 1$ 
4:   repetir
5:      $x_1 = \text{Agitación}(x, k)$ ;
6:      $x_2 = \text{Mejora}(x_1)$ ;
7:     si  $x_2$  es mejor que  $x$  entonces
8:        $x = x_2$ 
9:        $k = 1$ 
10:    si no
11:       $k = k + 1$ 
12:    fin si
13:  hasta que  $k = k_{max}$ 
14: fin mientras
```

La GVNS requiere como datos de entrada una solución inicial, y dos conjuntos de entornos. La solución inicial se obtiene mediante la fase constructiva que se explicó en la sección anterior, el primer conjunto de entornos (usados para el proceso de agitación) se compone de dos entornos generados por los movimientos *Re localización de un cliente* e *Intercambio* y el conjunto de entornos utilizado en el proceso de mejora se conforma de tres entornos generados por los movimientos *Intercambio*, *2-opt Inter rutas* y *2-opt Intra ruta*. La definición de los entornos se explica más adelante en esta misma sección.

En lo siguiente se procede a describir la mecánica de la GVNS. Se inicializa el valor de k en 1 (línea 2), en nuestro algoritmo k solo puede tomar dos valores, $k = 1$ corresponde al entorno definido por *Re localización de un cliente* y $k = 2$ corresponde

al definido por *Intercambio*. A partir de la solución inicial x (solución incumbente al inicio de la GVNS), el proceso *Agitación(x,k)* genera una solución aleatoria del k -ésimo entorno de x (línea 4) y a la solución resultante se le aplica un proceso de *Mejora* (línea 6). En la GVNS la mejora se realiza mediante la metaheurística VND, llamamos a la solución devuelta por la VND x_2 . Si la solución x_2 es mejor que la solución x , x_2 reemplaza a x convirtiéndose en la nueva solución incumbente (línea 8), además, el valor de k aumenta en una unidad, es decir, se cambia el entorno a ser explorado (línea 9). En el caso en que después de la mejora la solución x_2 no sea mejor que x , esta última se conserva y el valor de k vuelve a ser 1 (línea 11). Esta secuencia se repite mientras que el valor de k sea menor o igual que k_{max} (para nuestro caso $k_{max} = 2$).

Todo lo anterior se realiza n número de veces, en nuestro caso $n = 1$. La salida del algoritmo es la solución incumbente encontrada a lo largo del proceso.

En el algoritmo descrito, la mejora se realiza mediante la metaheurística VND, por lo que a continuación se detalla dicha metaheurística.

Algoritmo 5 VND

Entrada: Solución inicial, x ; y conjunto de entornos que se usarán en el descenso, N_k .

Salida: Solución incumbente, x .

- 1: $k = 1$
 - 2: **repetir**
 - 3: $x_2 = \text{ExploraciónDeEntorno}(x)$
 - 4: **si** x_2 es mejor que x **entonces**
 - 5: $x = x_2$
 - 6: $k = 1$
 - 7: **si no**
 - 8: $k = k + 1$
 - 9: **fin si**
 - 10: **hasta que** $k = k_{max}$
-

La solución inicial para la VND es la solución resultante del proceso de *Agitación* en

la GVNS, y los entornos utilizados son generados por los movimientos *Intercambio*, *2-opt Inter rutas* y *2-opt Intra ruta* que se explican más adelante. En el proceso *ExploraciónDeEntorno(x_1)* se busca la mejor solución del k -ésimo entorno de x_1 .

ENTORNOS

Un entorno $N(x)$ está definido como el conjunto de soluciones que se generan por un movimiento dado, aplicado a la solución x .

A continuación se describen los movimientos mediante los que son generados los entornos explorados en el GVNS-CCVRP. Con el fin de describir los movimientos se etiquetan a algunos de los clientes que participan en el movimiento. En las rutas originales los clientes p y u son el predecesor y sucesor, respectivamente, del cliente i , los clientes q y v son el predecesor y sucesor, respectivamente, del cliente j .

INTERCAMBIO DE CLIENTES ENTRE DOS RUTAS

Dadas dos rutas, k y k' , un cliente de cada ruta es elegido y se intercambian. El cliente i que estaba en la posición pi de la ruta k , pasa a estar en la ruta k' en la posición pj , mientras que el cliente j que originalmente estaba en la posición pj de la ruta k' ahora se encuentra en la posición pi de la ruta k . En la figura 4.1 se muestra el movimiento.

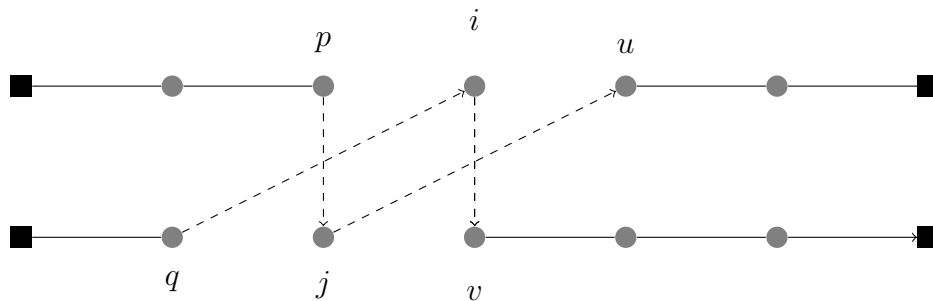


Figura 4.1: Intercambio de clientes

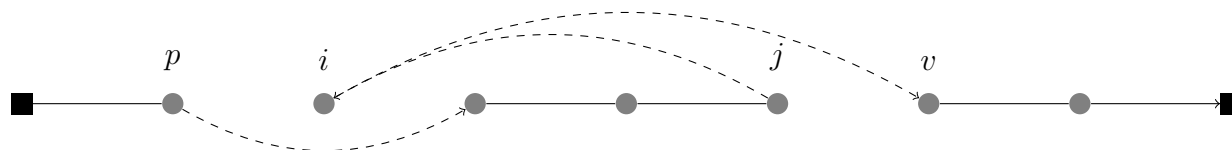


Figura 4.2: Relocalización de un cliente (Una ruta)

RELOCALIZACIÓN DE UN CLIENTE

Dadas dos rutas, k y k' , posiblemente iguales, se selecciona un cliente de cada ruta (clientes i y j), el cliente i de la ruta k se inserta como sucesor inmediato del cliente j de la ruta k' . Ambos clientes pueden estar en la misma ruta ($k_1 = k_2$), pero debe tomarse en cuenta que para realizar este movimiento el cliente i no puede ser originalmente sucesor inmediato del cliente j . En la figura 4.3 se muestra el movimiento si los clientes están en diferentes rutas, en la figura 4.2 se muestra el movimiento si ambos clientes están en la misma ruta.

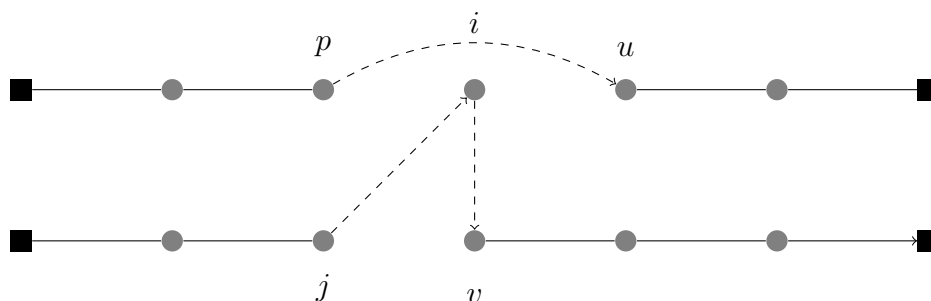


Figura 4.3: Relocalización de un cliente (Dos rutas)

2-OPT INTRA-RUTA

Dada una ruta k , se remueven dos arcos de la misma y se busca conectar de nuevo la ruta.

En la figura 4.4 se muestra el movimiento.

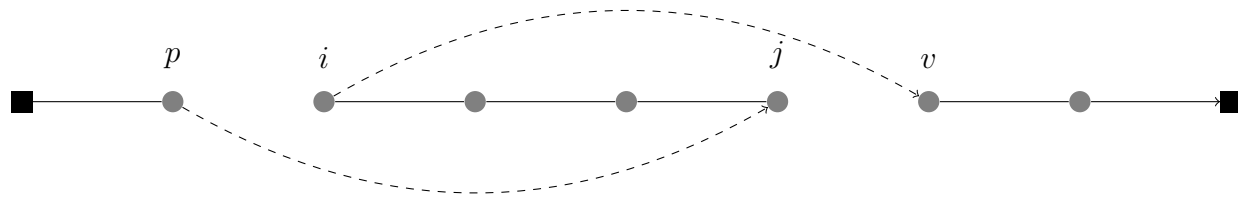


Figura 4.4: 2-opt Intra ruta

2-OPT INTER-RUTAS

Dadas dos rutas, k y k' , se remueve un arco de cada ruta y se agregan nuevos arcos tales que los nodos extremos sean clientes de diferentes rutas. En la figura 4.5 se muestra el movimiento.

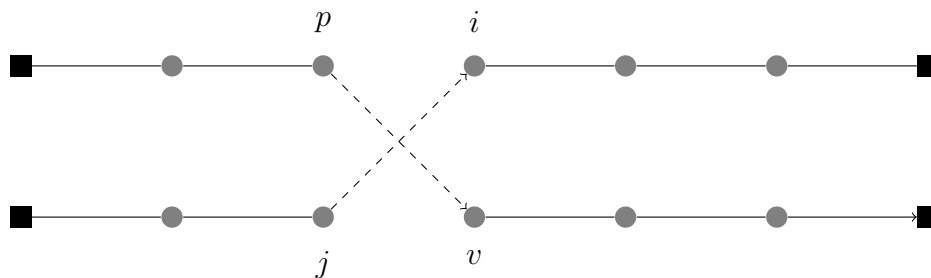


Figura 4.5: 2-opt Inter rutas

4.2 ENFOQUE DE SOLUCIÓN CONTINUO

En el enfoque continuo se obtiene una solución inicial con los clientes estáticos, y cada vez que llega la solicitud de un nuevo cliente dinámico, esta se busca asignar a algún vehículo de la solución actual mediante un algoritmo de inserción de clientes.

En esta tesis, la solución inicial para el enfoque continuo se obtiene con un algoritmo de dos fases. En la primera fase se utiliza el Constructivo 1 descrito en la sección 4.1.1, y en la segunda fase se utiliza el GVNS-CCVRP también descrito anteriormente.

En el Algoritmo 6 se muestra el esquema general de la metodología de solución del enfoque continuo.

Algoritmo 6 Enfoque continuo

Entrada: Conjunto de clientes estáticos, conjunto de entornos que se usarán en la GVNS-CCVRP.

Salida: Solución incumbente, x .

- 1: $x = \text{Constructivo1}()$;
 - 2: $\text{GVNS-CCVRP}(x)$;
 - 3: **mientras** Lleguen nuevos clientes **hacer**
 - 4: $\text{AlgoritmoDeInsercion}(x)$;
 - 5: **fin mientras**
-

En el DVRP los vehículos correspondientes a cada ruta de la solución inicial comienzan a realizar su servicio, pero en el transcurso de la ejecución de sus respectivas rutas es posible que nuevos clientes demanden servicio, por lo que se requiere modificar la solución inicial.

Una consideración importante para realizar la modificación de la solución es decidir si es posible o no utilizar más vehículos para la ejecución de las rutas, en caso negativo podría haber clientes dinámicos cuya solicitud de servicio no es posible atender por restricciones de capacidad. Para analizar el enfoque dinámico en cualquiera de los dos casos, en el presente trabajo se proponen dos algoritmos de inserción de clientes dinámicos. En el Algoritmo de inserción 1 se considera que no es posible tener acceso a más vehículos por lo que puede haber clientes rechazados (clientes pospuestos), mientras que en el Algoritmo de inserción 2 se considera la posibilidad de utilizar más vehículos para poder dar servicio a todos los clientes dinámicos, permitiendo la creación de rutas adicionales.

4.2.1 ALGORITMOS DE INSERCIÓN DE CLIENTES DINÁMICOS

Note que al agregar rutas extras a la solución nuestra función objetivo se ve beneficiada ya que la latencia disminuye, pero nosotros estamos buscando soluciones con el mínimo número de vehículos que cumplan con la restricción de capacidad, ya que en la práctica puede no ser posible disponer de vehículos extra. Sin embargo, existen

casos en los que con la cantidad limitada de vehículos no es posible atender a todos los clientes por lo que será necesario rechazar su solicitud de servicio.

Para analizar el comportamiento de la función objetivo en ambos casos citados, se han diseñado dos algoritmos de inserción. En el primero de ellos, *Algoritmo de inserción 1*, se rechazan los clientes que no pueden ser atendidos con el número de vehículos fijo, en el segundo algoritmo, *Algoritmo de inserción 2*, se abre una nueva ruta y se re optimiza la solución.

ALGORITMO DE INSERCIÓN 1

Algoritmo 7 Algoritmo de inserción 1

Entrada: Solución inicial con clientes estáticos, x_i ;

Salida: Solución final con clientes estáticos y dinámicos, x_f ;

```
1: mientras tiempo  $\leq$  T hacer
2:   ActualizarDatos();
3:   r=BuscarRutas();
4:   si r = 0 entonces
5:     Reacomodar();
6:     BuscarRutas();
7:     si r = 0 entonces
8:       RechazarDinamico();
9:     si no
10:      InsertarDinamico();
11:    fin si
12:  si no
13:    InsertarDinamico();
14:  fin si
15: fin mientras
```

El Algoritmo de Inserción 1 inicia con la solución obtenida mediante el Constructivo 1 y el GVNS-CCVRP que se explican anteriormente para el enfoque estático. A

partir de ésta, cada vez que llega un cliente dinámico se actualizan los datos de la solución (línea 2), tanto los clientes que han sido atendidos hasta ese momento como el cliente que está siendo atendido o que será próximo a atender en cada una de las rutas. Llamaremos a éste cliente actual. Enseguida se buscan las rutas con capacidad restante mayor o igual a la demanda del cliente dinámico (línea 3); si existe al menos una ruta, se busca la mejor inserción del cliente en ella considerando solo las posiciones posteriores al cliente actual (línea 13). La mejor inserción es aquella con menor cambio en la latencia. Se agrega al cliente en la posición de la ruta que es su mejor inserción.

En caso de no encontrar ninguna ruta con capacidad suficiente para atender al cliente, se verifica si la suma de las capacidades restantes de todas las rutas es mayor o igual a la demanda del cliente, si es así se procede como sigue. En todas las rutas cuya capacidad restante es mayor que cero se re-acomodan los clientes que no han sido atendidos hasta el momento, buscando dejar capacidad suficiente en una ruta para insertar al cliente dinámico (línea 5). Este re-acomodo se hace ordenando las rutas en orden descendente referente a la capacidad restante, y los clientes de cada ruta se ordenan en orden ascendente referente a su demanda. Se intenta eliminar los clientes con demandas más chicas de las rutas con mayor capacidad restante, buscando insertarlos en las rutas con menor capacidad restante. Si ha sido posible dejar una ruta con capacidad para atender al cliente, éste se agrega en la mejor inserción (línea 6). Al realizar este movimiento de re-acomodo de clientes permitimos que la latencia aumente, ya que por el momento el fin es cumplir con la restricción de capacidad, y evitar el uso de más vehículos o de rechazo de la solicitud del cliente. Si al finalizar este proceso no es posible agregar al cliente dinámico, éste se rechaza (línea 8).

ALGORITMO DE INSERCIÓN 2

Algoritmo 8 Algoritmo de inserción 2

Entrada: Solución inicial con clientes estáticos, x_i ;**Salida:** Solución final con clientes estáticos y dinámicos, x_f ;

```
1: mientras tiempo  $\leq$  T hacer
2:   ActualizarDatos();
3:   r=BuscarRutas();
4:   si r = 0 entonces
5:     AbrirNuevaRuta();
6:     GVNS-CCVRP();
7:   si no
8:     InsertarDinamico();
9:   fin si
10: fin mientras
```

En el segundo algoritmo de inserción de clientes se realiza, al igual que en el primero, la actualización de los datos (línea 2), y la búsqueda de rutas en las cuales sea factible insertar al cliente (línea 3), si existe alguna ruta se agrega al cliente en su mejor inserción (línea 8). La diferencia con el algoritmo 1 radica en la acción a realizar si no existe ruta alguna donde se pueda agregar al cliente dinámico, en este caso se abre una nueva ruta (línea 5) y dado que al abrir una nueva ruta es posible disminuir la latencia de la solución, se llama al GVNS-CCVRP para realizar un proceso de mejora considerando todos los clientes que no han sido atendidos aun (incluyendo al cliente dinámico).

CAPÍTULO 5

EXPERIENCIA COMPUTACIONAL

En esta tesis se han propuesto algoritmos para dar solución al problema planteado. Debido a la naturaleza heurística de los mismos no se garantiza obtener soluciones óptimas para el problema, por lo que es necesario evaluar dichos algoritmos de forma estadística.

Para la evaluación de los algoritmos se realizaron los siguientes experimentos:

- Experimentación con el GVNS. Se realiza experimentación para calibrar los parámetros de este algoritmo.
- Experimentación con el enfoque periódico. Se realiza experimentación para comparar los valores de latencia obtenidos con distintos valores del número de periodos en el que se particiona el horizonte de planeación.
- Experimentación con el enfoque continuo. Se realiza experimentación con los dos algoritmos de inserción de clientes para evaluar su desempeño.

Es importante mencionar que para realizar la experimentación la metodología propuesta se implementó en el lenguaje C++, en el ambiente de Visual Studio 2012. Esto en un procesador Intel Core i5 bajo el sistema operativo Windows 8.

En esta sección se describe el estudio computacional realizado para evaluar el desempeño de los algoritmos propuestos. Primeramente se exponen las características de las instancias utilizadas, luego se procede a la explicación de los experimentos mencionados anteriormente y se muestran los resultados obtenidos, por último se realiza el análisis de resultados.

5.1 INSTANCIAS

Para la evaluación de los algoritmos presentados se realizó experimentación sobre un conjunto de instancias adaptadas de aquellas usadas en la literatura para el CVRP. Se describen en lo siguiente.

- Instancias chicas. Se usaron 7 instancias de pocos clientes, la más chica con 50 clientes y la de mayor tamaño tiene 199 clientes. [8]
- Instancias grandes. Se usaron 10 instancias a las que llamamos grandes, las cuales tienen desde 200 hasta 396 clientes. [22]

En el apéndice A se visualiza la dispersión geográfica de los clientes para cada una de las instancias.

Como se ha dicho, dichas instancias fueron propuestas para la versión estática del problema de rutas, por lo que para todos los clientes el tiempo en que ingresa al sistema es el inicio del horizonte de planeación. Para ser usadas en nuestra evaluación se generaron tiempos de llegada de los clientes de forma aleatoria con una distribución uniforme entre 0 y 720.

Por cada una de las instancias se generan cinco nuevas instancias con grados de dinamismo 10, 30, 50, 70 y 90, que corresponden a porcentajes del 10% al 90% de clientes dinámicos. En total se experimenta con 85 instancias diferentes del problema.

5.2 EXPERIMENTACIÓN CON EL GVNS

5.2.1 CALIBRACIÓN DE PARÁMETROS

El desempeño del algoritmo de mejora depende de los valores de sus parámetros, esto es, el número de vecindarios utilizados, el orden de los vecindarios y el número de iteraciones del algoritmo; por lo que se realiza experimentación para calibrar dichos parámetros.

La elección del número de vecindarios tanto para la VND como para el proceso de Agitación fue realizada mediante experimentación pre-eliminar, de acuerdo al buen desempeño del algoritmo.

Para la calibración del parámetro *orden de vecindarios*, así como la elección de los vecindarios tanto para la VND como para el proceso de Agitación se realizó un diseño factorial con dos factores. El primer factor es el orden de los vecindarios para la búsqueda local, de las 24 diferentes permutaciones al utilizar 3 de los 4 entornos descritos anteriormente, se eligieron las 7 mejores (véase tabla 5.1 de acuerdo a experimentación pre-eliminar para realizar el experimento; el segundo factor es el orden de los vecindarios para el proceso de Agitación, en el cual hay 12 distintas maneras de ordenar 2 entornos seleccionados de entre los 4 disponibles (véase tabla 5.2). El experimento consiste en 10 repeticiones de ejecución del algoritmo GVNS-CCVRP con los 84 diferentes formas de combinar los 7 niveles del factor 1 con los 12 niveles del factor 2.

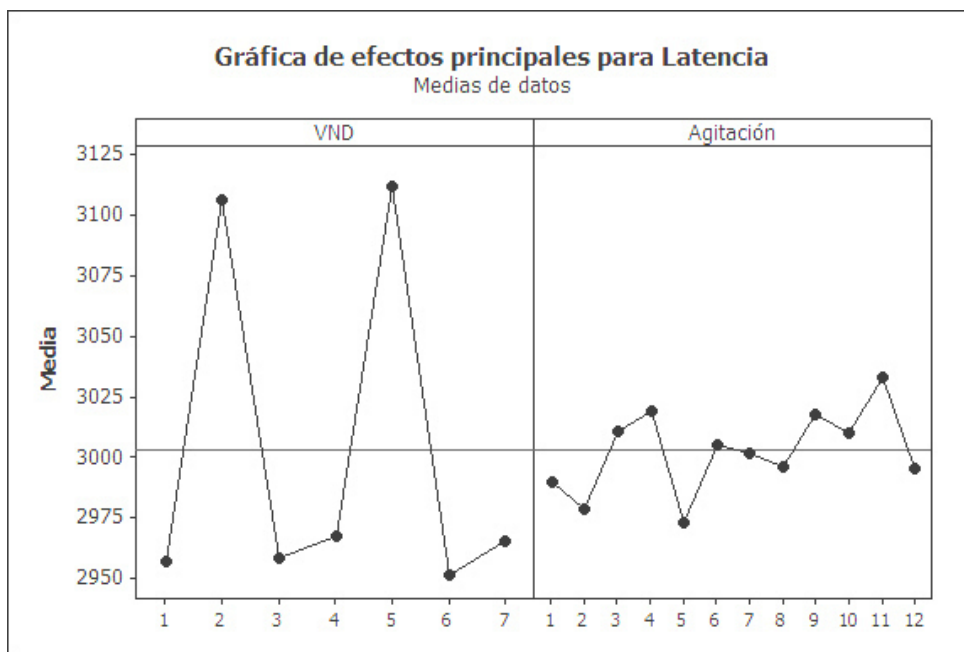


Figura 5.1: Gráfica de efectos principales para Latencia

Dado que los valores observados de latencia resultantes de la experimentación no cumplen con supuestos de normalidad, se ha utilizado la prueba no paramétrica de

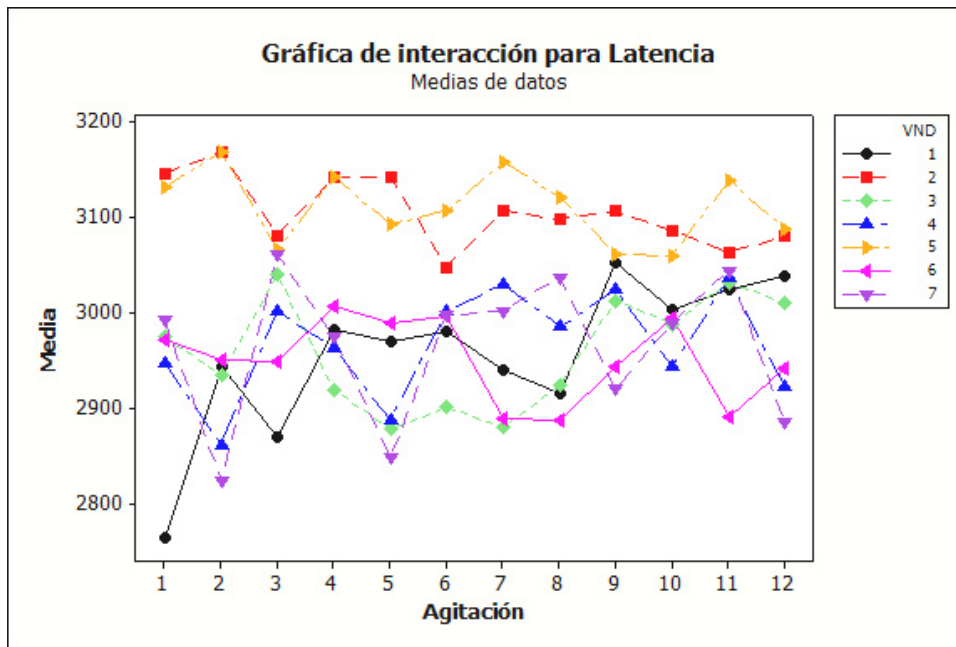


Figura 5.2: Gráfica de interacción entre factores para Latencia

Friedman. El resultado de esta prueba con un nivel de significancia de 0.01, muestra que si hay diferencia significativa entre los efectos de las configuraciones para la VND en el valor de la latencia de las soluciones, mientras que las distintas configuraciones de los entornos para la Agitación no presentan diferencia significativa en el valor de la latencia.

La mejor configuración para la VND (factor 1) es Intercambio, 2-opt Inter rutas, 2-opt Intra rutas (nivel 6), mientras que la mejor configuración para los entornos de la Agitación (factor 2) fijando la configuración para la VND mencionada anteriormente es 2-opt Intra rutas y Relocalización (nivel 8). En las figuras 5.1 se muestra el promedio en la latencia (eje y) obtenido en las 10 iteraciones para los distintos niveles de los factores (eje x) y en la figura 5.2 se muestra el comportamiento de la interacción entre los factores, en el eje x se muestran los distintos niveles del factor 2 (Agitación) y en el eje y se muestra la media de la latencia. En la tabla 5.1 se muestran las configuraciones de movimientos que corresponden a los distintos niveles del factor 1 utilizados en la experimentación y en la tabla 5.2 se muestran las configuraciones correspondientes al factor 2.

Factor 1. Entornos en VND

Niveles:

1	Intercambio, Relocalización, 2-opt intra ruta
2	Intercambio, Relocalización, 2-opt inter rutas
3	Intercambio, 2-opt intra ruta, 2-opt inter rutas
4	Intercambio, 2-opt intra ruta, Relocalización
5	Intercambio, 2-opt inter rutas, Relocalización
6	Intercambio, 2-opt inter rutas, 2-opt intra ruta
7	Relocalización, Intercambio, 2-opt intra ruta

Tabla 5.1: Niveles para factor 1 (VND)

Factor 2. Entornos en Agitación

Niveles:

1	Intercambio, relocalización
2	Intercambio, 2-opt intra ruta
3	Intercambio, 2-opt inter rutas
4	Relocalización, Intercambio
5	Relocalización, 2-opt intra ruta
6	Relocalización, 2-opt inter rutas
7	2-opt intra ruta, Intercambio
8	2-opt intra ruta, Relocalización
9	2-opt intra ruta, 2-opt inter rutas
10	2-opt inter rutas, Intercambio
11	2-opt inter rutas, Relocalización
12	2-opt inter rutas, 2-opt intra ruta

Tabla 5.2: Niveles para factor 2 (Agitación)

Para decidir el número de iteraciones en la GVNS se experimentó con los valores 1, 5, 10 y 50. Sin embargo solo se producen soluciones de mejoría notable en el caso de 50 iteraciones, mientras que el tiempo de cómputo se multiplica tanto como el número de iteraciones del algoritmo. En instancias chicas donde el tiempo de cómputo es apenas algunos segundos podría utilizarse este valor para el número de iteraciones, pero en instancias grandes, donde el tiempo de cómputo llega a ser mayor a una hora, no se puede optar por valores grandes para el parámetro. En este trabajo se fija el valor del parámetro *número de iteraciones* en 1.

5.3 EXPERIMENTACIÓN CON EL ENFOQUE PERIÓDICO

En el enfoque periódico es posible realizar distintas particiones del período de planeación. Es de interés conocer el comportamiento del algoritmo para distintos valores del número de períodos en que se divide el horizonte de planeación, por lo que se realizaron pruebas con 4, 6 y 12 períodos.

La metodología propuesta para el enfoque periódico no permite abrir nuevas rutas, por lo que en caso de existir clientes que no pueden ser atendidos la solicitud de servicio de los mismos se rechaza. En vista de lo anterior, el número de clientes pospuestos se suma a las variables que nos interesa analizar, a saber: i) latencia de la solución, ii) tiempo de cómputo para resolver la instancia, y iii) número de clientes pospuestos.

En las figuras 5.3, 5.4 y 5.5, se grafican los promedios obtenidos para la latencia, el tiempo de ejecución del algoritmo para encontrar la solución y el número de clientes pospuestos. En cada una de las gráficas se muestran los resultados para cada una de las instancias chicas con sus diferentes grados de dinamismo. En cada figura se muestran los resultados con una de las distintas particiones del horizonte de planeación.

En las figuras 5.6, 5.7 y 5.8 se grafican los promedios obtenidos para la latencia, el tiempo de cómputo empleado y el número de clientes pospuestos. En cada gráfica

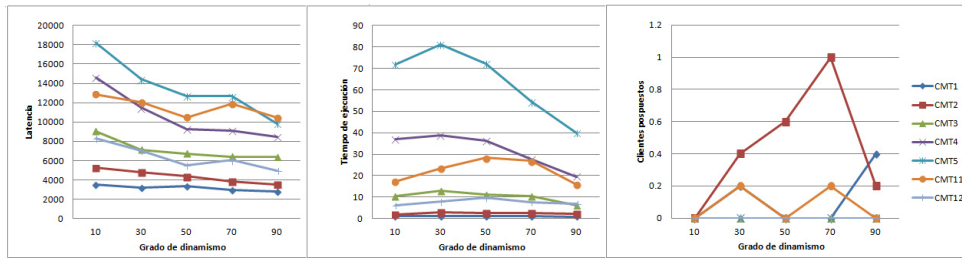


Figura 5.3: CMT. Periódico 4-p

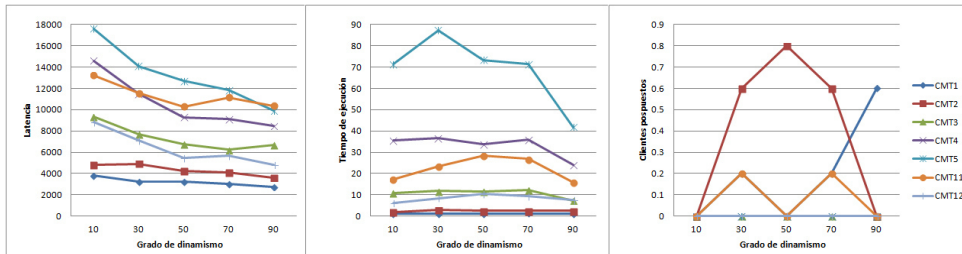


Figura 5.4: CMT. Periódico 6-p

se muestran los resultados para cada una de las instancias chicas con sus diferentes grados de dinamismo comparando los promedios obtenidos con las diferentes particiones de tiempo con las que se experimentaron (4, 6 y 12 periodos).

Se puede observar que en cada instancia, existe una tendencia en la latencia, esta disminuye conforme disminuye el grado de dinamismo. Esto puede deberse a que al haber un menor número de clientes dinámicos son menos los cambios que deben hacerse en la ruta inicial posterior al inicio de las trayectorias de los vehículos, evitando aumentar en gran manera la latencia de la solución inicial. Por otra parte, el tiempo para resolver la instancia aumenta conforme disminuye el grado de dinamismo.

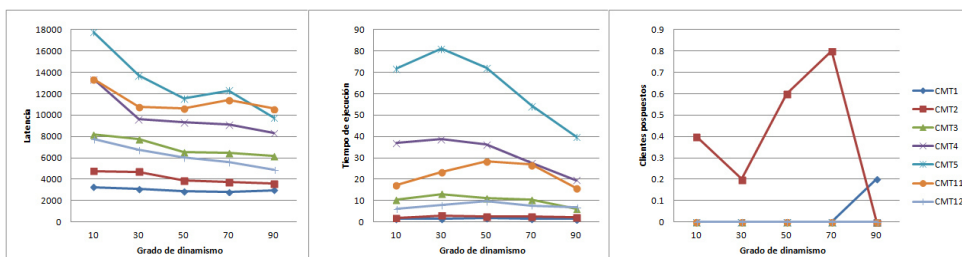


Figura 5.5: CMT. Periódico 12-p

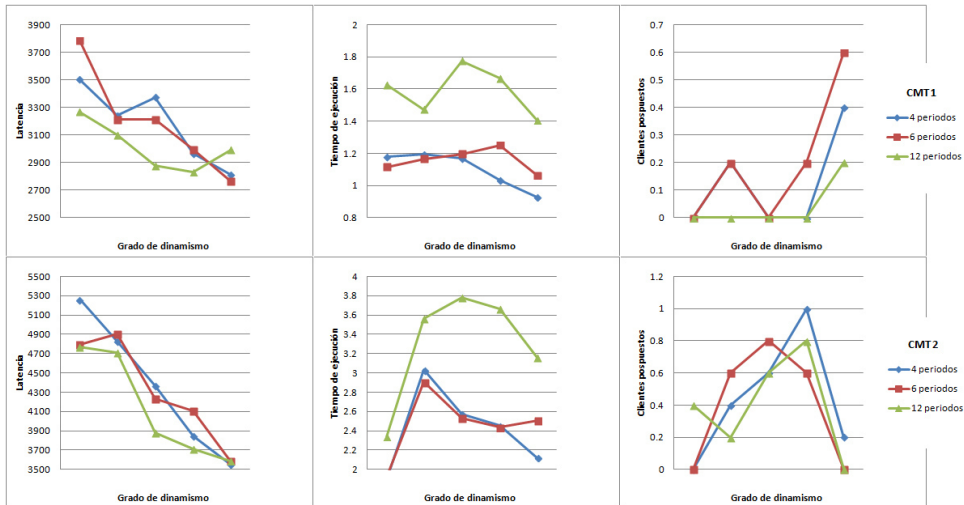


Figura 5.6: Enfoque periódico. CMT1 y CMT2

En lo que respecta a las distintos valores del número de periodos con los que se experimentaron, podemos ver que las mejores soluciones se encuentran con una partición de 12 periodos, por lo que podemos suponer que a mayor número de periodos las soluciones son mejores. Sin embargo, como se esperaba, el enfoque periódico con 12 periodos es el que utiliza un mayor tiempo de cómputo en todas las instancias, pero esta diferencia podría despreciarse, ya que son algunos segundos o décimas de segundo.

5.4 EXPERIMENTACIÓN CON EL ENFOQUE CONTINUO

En los siguientes gráficos se muestra el desempeño de la metodología para el enfoque continuo usando el Algoritmo de inserción 1, en el eje horizontal encontramos la latencia de la solución, mientras que el eje vertical representa el tiempo de cómputo en segundos. Cada gráfico muestra el resultado de 10 réplicas del experimento para cada instancia con sus diferentes grados de dinamismo.

En ambos algoritmos de inserción se observa que el valor de la latencia aumenta conforme el grado de dinamismo aumenta, esto puede deberse a que al tener un mayor grado de dinamismo existe un mayor porcentaje de clientes desconocidos al inicio de la planeación, los cuales llegarán después de iniciadas las rutas, por lo que a

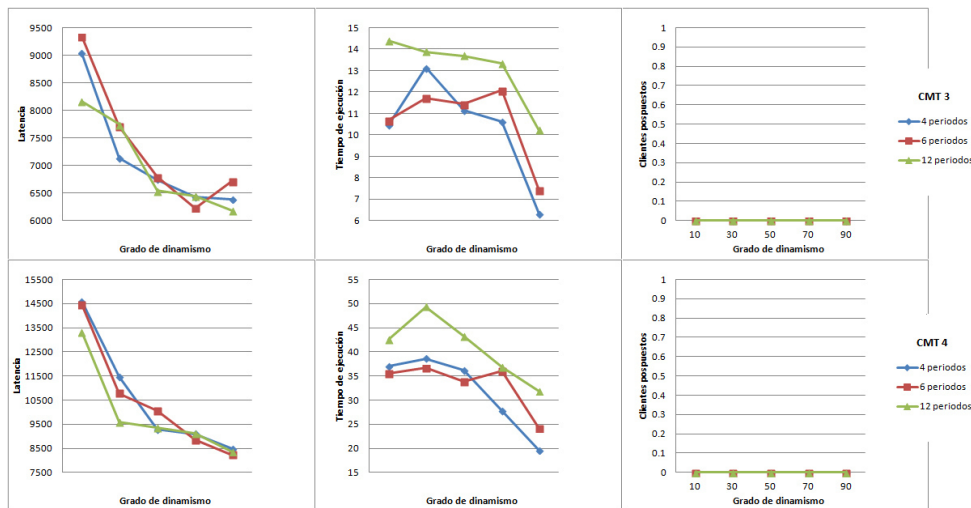


Figura 5.7: Enfoque periódico. CMT3 y CMT4

lo largo del horizonte de planeación se deben hacer más modificaciones. En algunos casos ya no será posible cambiar clientes que pudieron haberse asignado en otra ruta donde el aumento de la latencia podría haber sido menor, debido a que dichos clientes ya fueron asignados y atendidos por un vehículo.

En el Algoritmo de inserción 2 no existe una tendencia en el cambio del promedio en los valores de latencia para los grados de dinamismo 10, 30 y 70 % en cada instancia. Solo se observa el aumento en el valor de la latencia para los grados de dinamismo 70 y 90 %.

El aparente comportamiento similar en las instancias con grados de dinamismo 10, 30 y 70 % se explica por la manera en que el algoritmo de inserción 2 funciona. Como se ha mencionado anteriormente, permite la creación de nuevas rutas, por lo que soluciones para la misma instancia tienen un valor de latencia mucho menor que otras debido a que se han creado más rutas, de esta manera, la media de los valores de la latencia no puede usarse como fuente de comparación entre la misma instancia con diferentes grados de dinamismo. En los casos de grados de dinamismo 70 y 90 % las soluciones encontradas tienen todas el mismo número de rutas, por lo que la situación explicada anteriormente no se presenta.

En cuanto al tiempo computacional empleado, éste disminuye al aumentar el grado

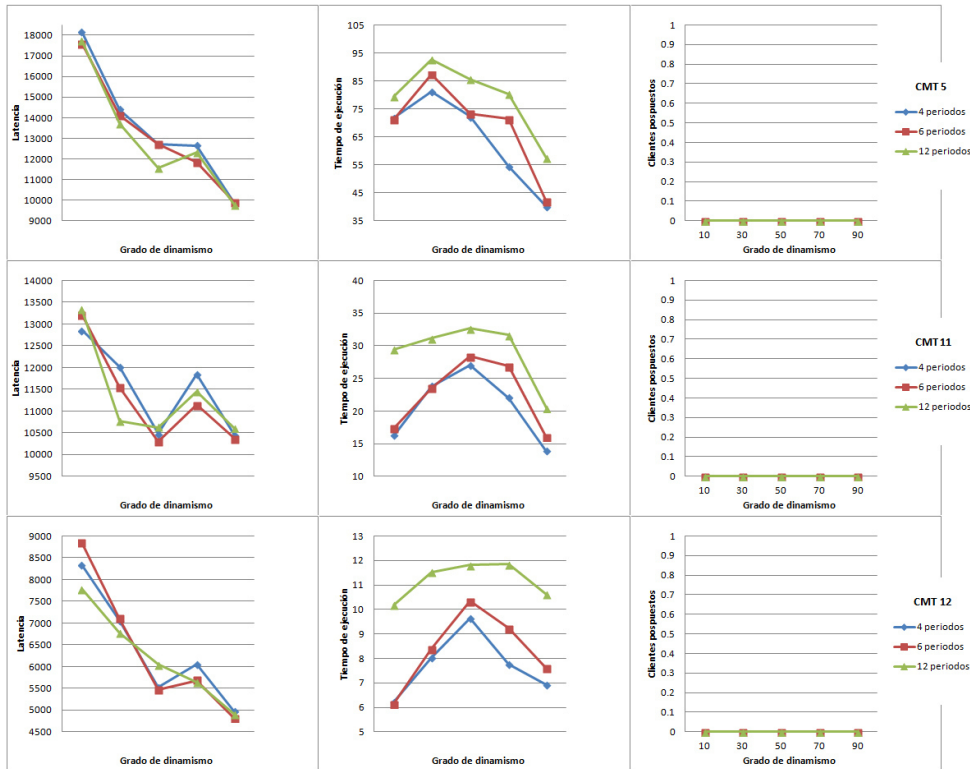


Figura 5.8: Enfoque periódico. CMT5, CMT11 y CMT12

de dinamismo, posiblemente porque existen menos posibles lugares donde se pueden agregar a los clientes dinámicos (posiciones posteriores al cliente actual de cada ruta), dejando pocas soluciones factibles para explorar.

Note que ambos algoritmos no son comparables entre si, debido a que el Algoritmo de inserción 1 no permite la creación de nuevas rutas, mientras que en el Algoritmo de inserción 2 sí se abren más rutas. Con un mayor número de rutas, la latencia final de la solución puede disminuir siendo injusta una comparación entre soluciones.

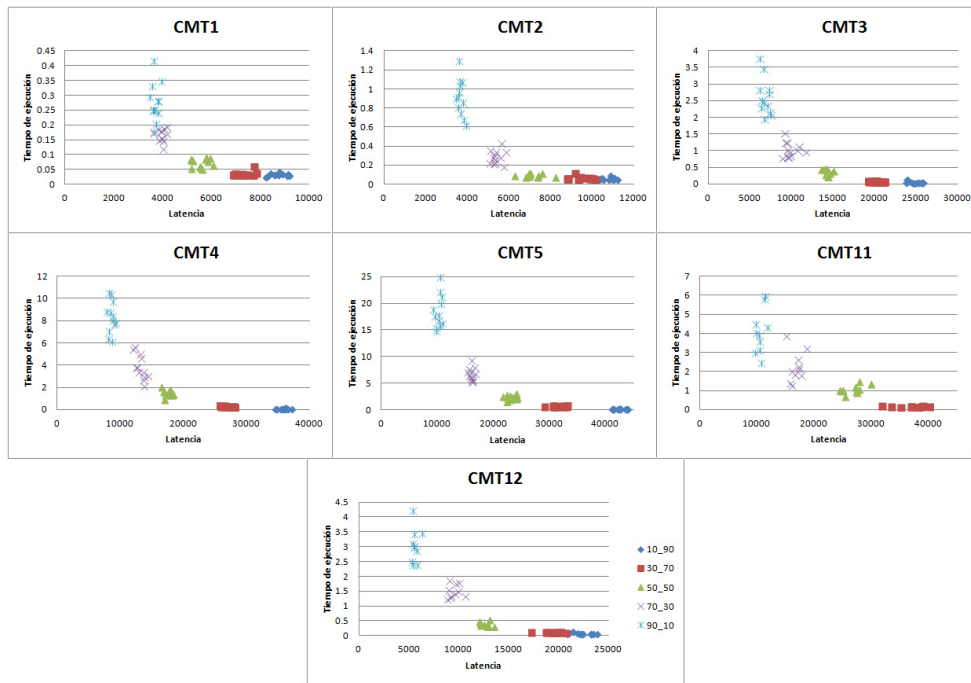


Figura 5.9: Algoritmo de inserción 1. Enfoque continuo

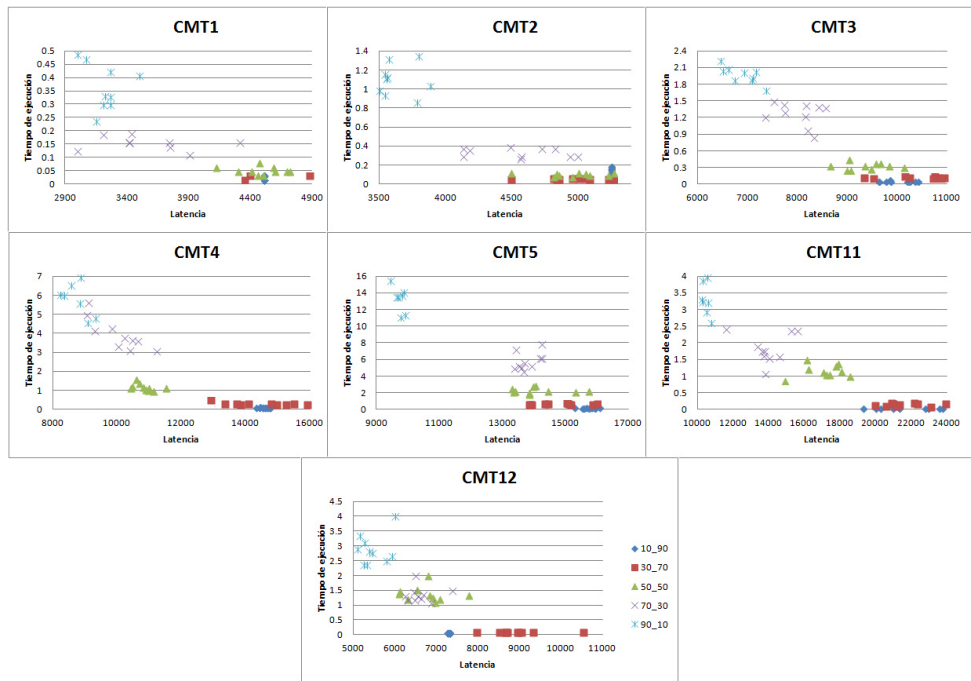


Figura 5.10: Algoritmo de inserción 2. Enfoque continuo

CAPÍTULO 6

CONCLUSIONES Y TRABAJO A FUTURO

En este trabajo se estudió un problema de rutas de vehículos con el objetivo de minimizar la suma de los tiempos de espera de los clientes considerando clientes dinámicos.

Se diseñaron algoritmos heurísticos utilizando dos enfoques de solución. El enfoque de solución periódico se basa en dividir el horizonte de planeación en periodos de igual duración y en cada periodo se resuelve el problema de rutas estático con los clientes que se conocen hasta el momento. En el enfoque de solución dinámico cada vez que se conoce una nueva solicitud de servicio, se intenta asignar a alguna de las rutas de la solución actual mediante un algoritmo de inserción eficiente.

En ambos enfoques de solución se ha utilizado un algoritmo de dos fases, la primera de ellas tiene el objetivo de construir una solución, la segunda fase busca mejorar dicha solución. Para realizar la mejora se desarrollo un algoritmo basado en la metaheurística Búsqueda por Entorno Variable. En el enfoque de solución continuo, además de utilizar el algoritmo de dos fases, se han diseñado e implementado dos algoritmos de inserción de clientes.

En el primer algoritmo de inserción de clientes se asume que no es posible disponer más vehículos de los que iniciaron sus rutas al inicio del período de planeación, en el segundo algoritmo se permite añadir más vehículos a la solución para poder atender a todos los clientes.

De acuerdo a los resultados de la experimentación realizada podemos concluir lo siguiente:

- En el enfoque de solución continuo las soluciones obtenidas mejoran conforme aumenta el número de periodos en el que se divide el horizonte de planeación. En esta experimentación en particular, se dividió el día en 4, 6 y 12 periodos, siendo este último el que obtiene los mejores valores de latencia, sacrificando poco tiempo de cómputo adicional.
- En el enfoque de optimización continuo el algoritmo de solución obtiene soluciones con mayor valor de latencia conforme hay un mayor número de clientes dinámicos.
- El enfoque de solución continuo ofrece mejores soluciones que el enfoque de solución periódico, en un tiempo de cómputo menor.

Como trabajo futuro se pretende probar con otros métodos constructivos para determinar en que medida es posible mejorar las soluciones encontradas por los algoritmos expuestos en el presente trabajo. Además se tiene la idea de mejorar el algoritmo basado en GVNS incorporando memoria en el método.

APÉNDICE A

GLOSARIO DE TÉRMINOS

Centro de distribución Ubicación geográfica donde inician y terminan todas las rutas, también se denomina deposito.

Cliente Solicitudes de servicio que deben ser atendidas. Cada uno tiene asociada inicialmente ubicación geográfica, demanda y tiempo de arribo.

Demanda Tamaño de orden de servicio de un cliente, sus unidades dependerán del problema particular a resolver.

Deposito Ubicación geográfica donde inician y terminan todas las rutas, también se denomina centro de distribución.

Entorno Conjunto de soluciones que se generan por un movimiento aplicado a una solución.

Heurística Técnica o procedimiento inteligente para realizar una tarea que no es producto de un riguroso análisis formal, sino de conocimiento experto sobre la tarea.

Incumbente Mejor solución encontrada.

Instancia Conjunto de datos que satisfacen las condiciones de un problema particular.

Latencia Para un nodo es el tiempo de espera para ser atendido; para una solución es la suma de los tiempos de espera de todo los clientes en ella.

Metaheurística Estrategia general para diseñar procedimientos heurísticos con el fin de resolver un problema de optimización mediante un proceso de búsqueda en un cierto espacio de soluciones alternativas.

APÉNDICE B

INSTANCIAS

En este apéndice se muestra la distribución geográfica de los clientes en las instancias utilizadas en la fase experimental.

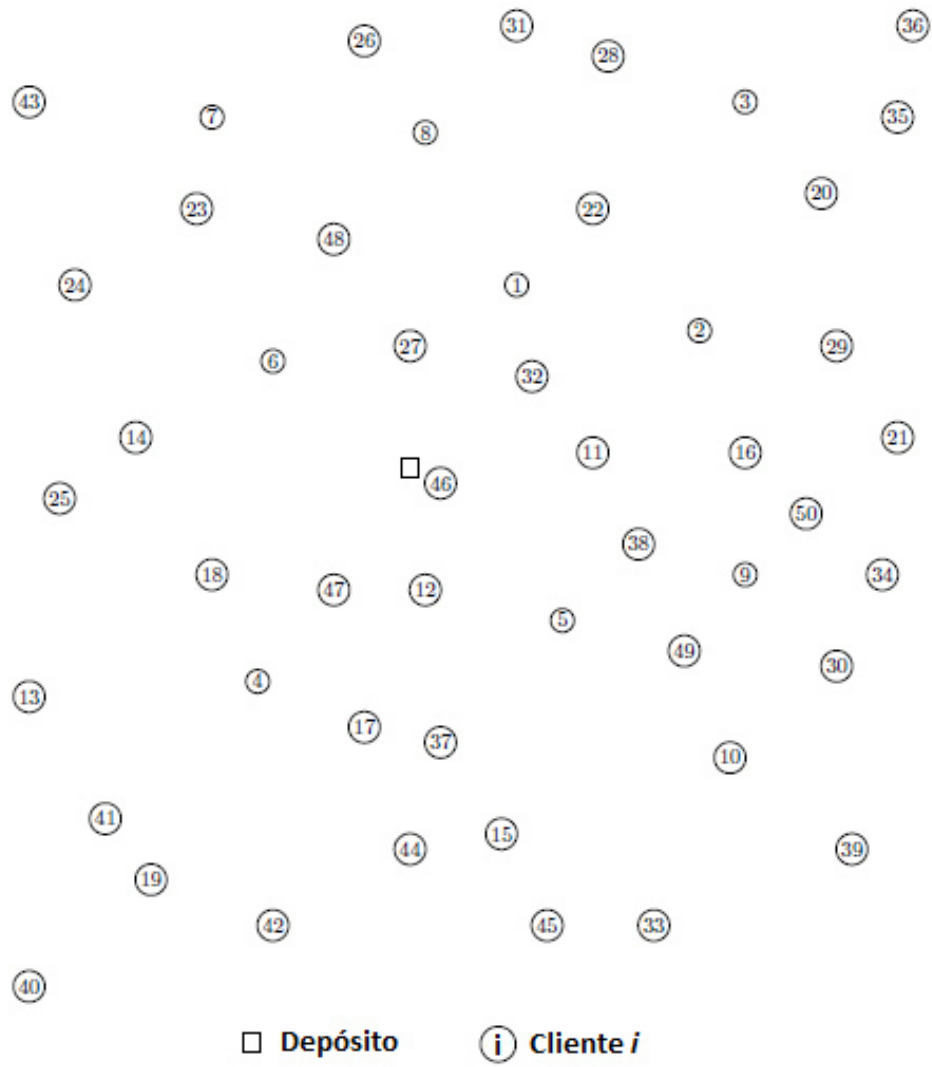


Figura B.1: Instancia CMT1

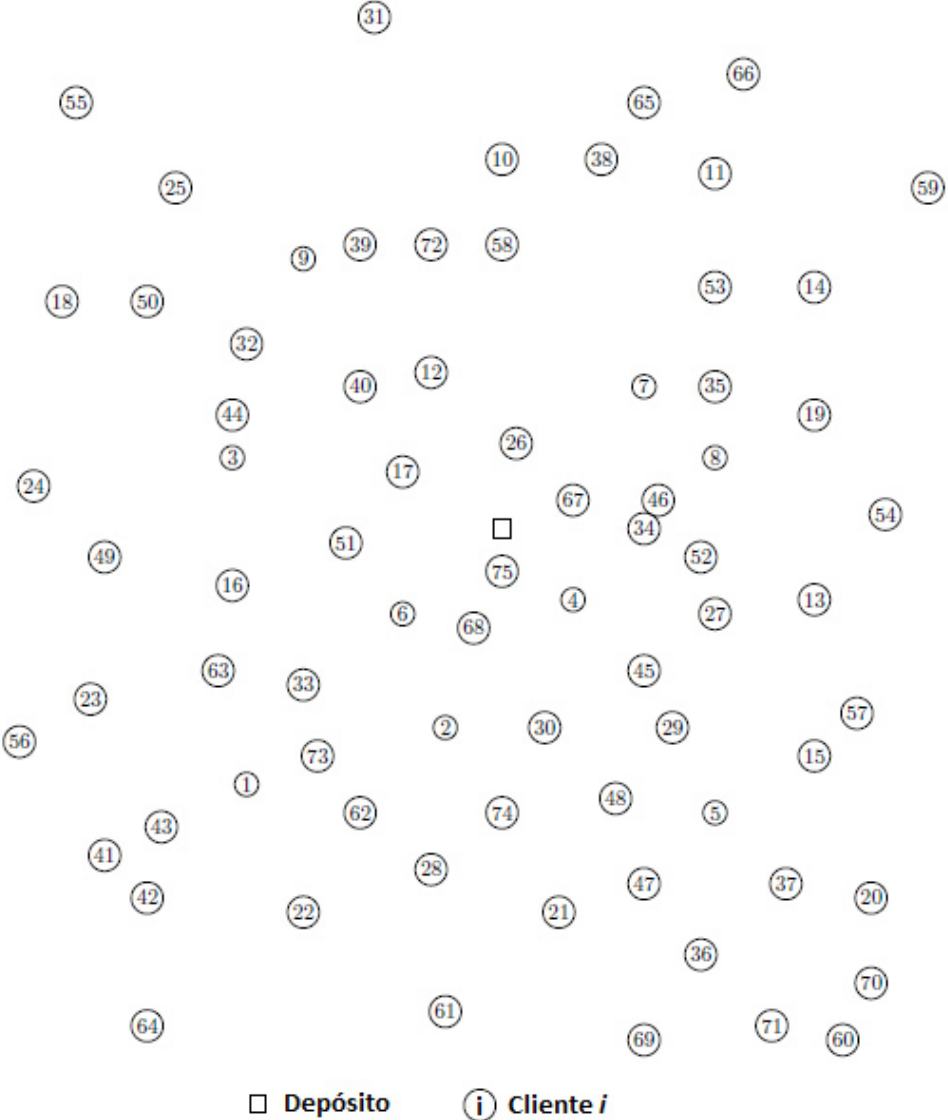


Figura B.2: Instancia CMT2

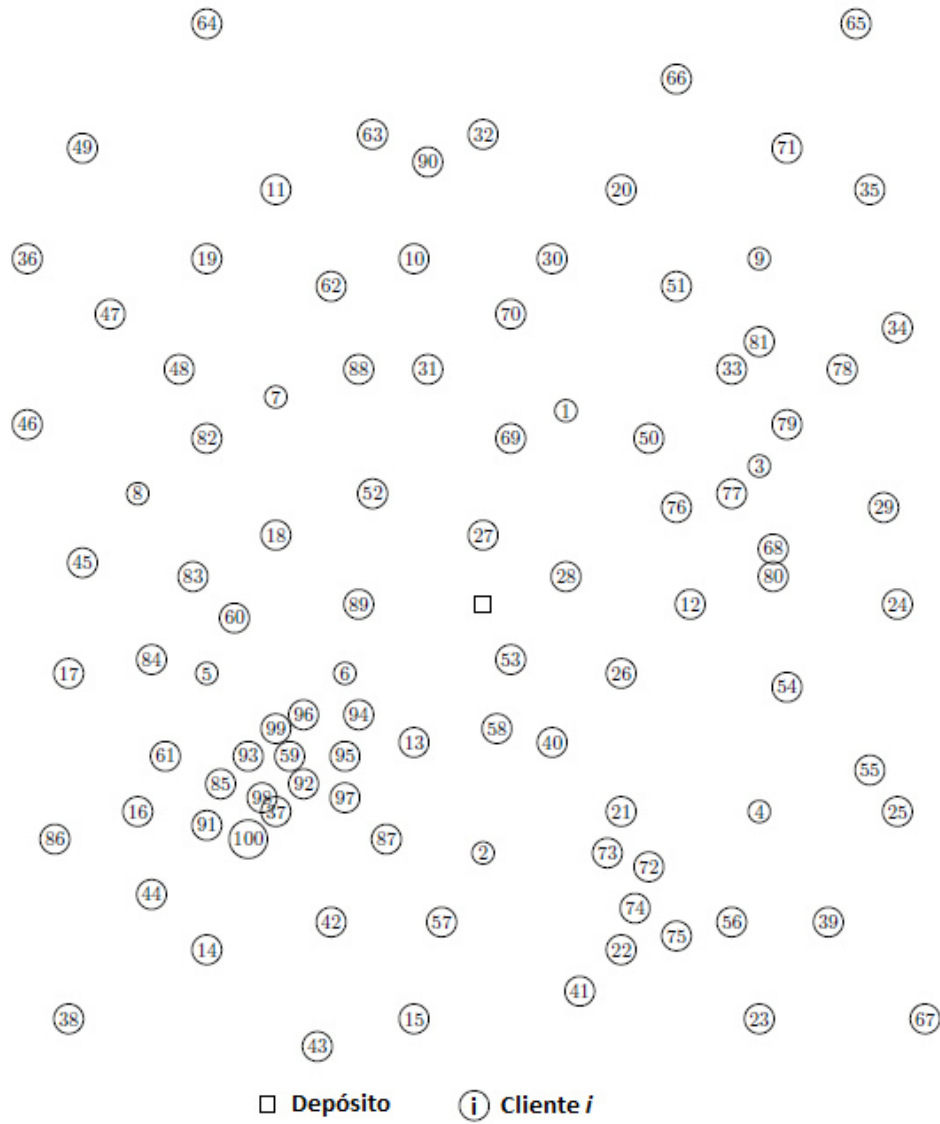


Figura B.3: Instancia CMT3

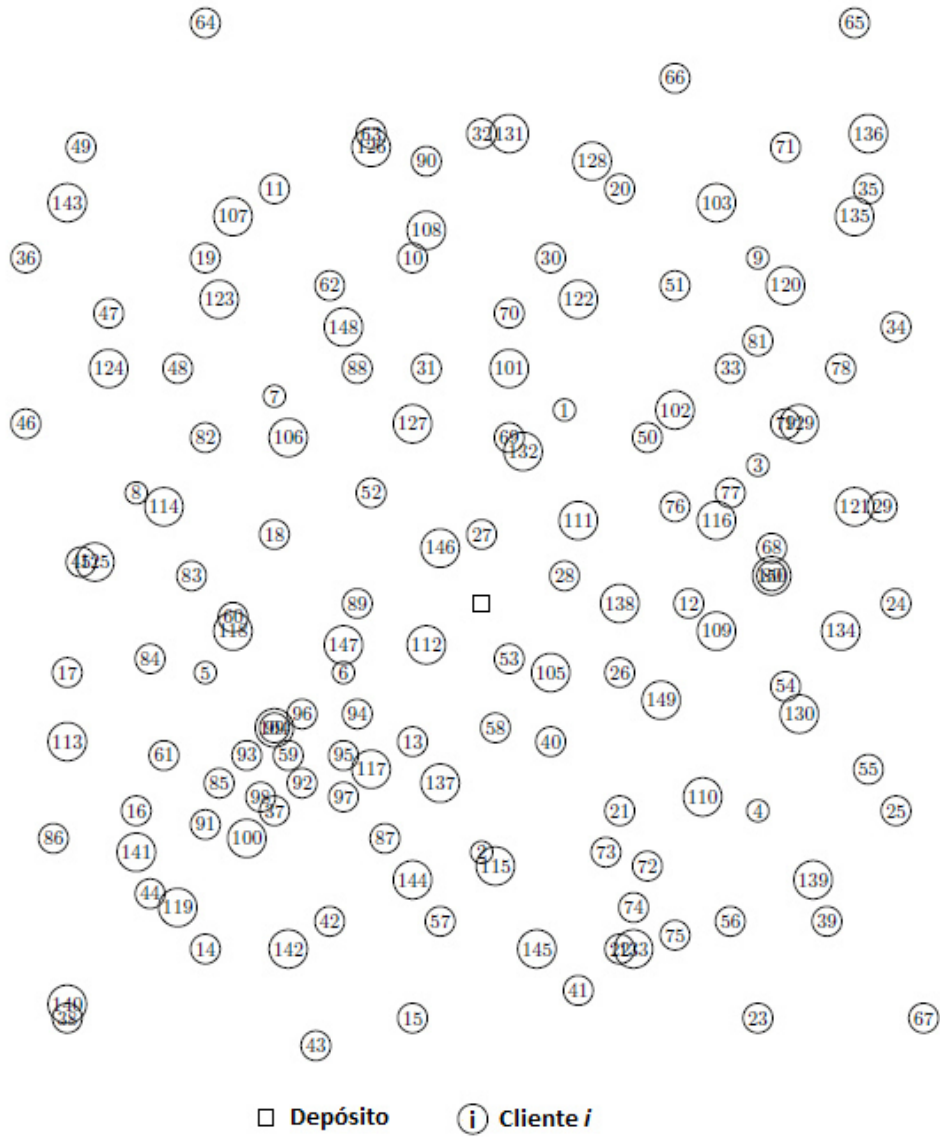


Figura B.4: Instancia CMT4

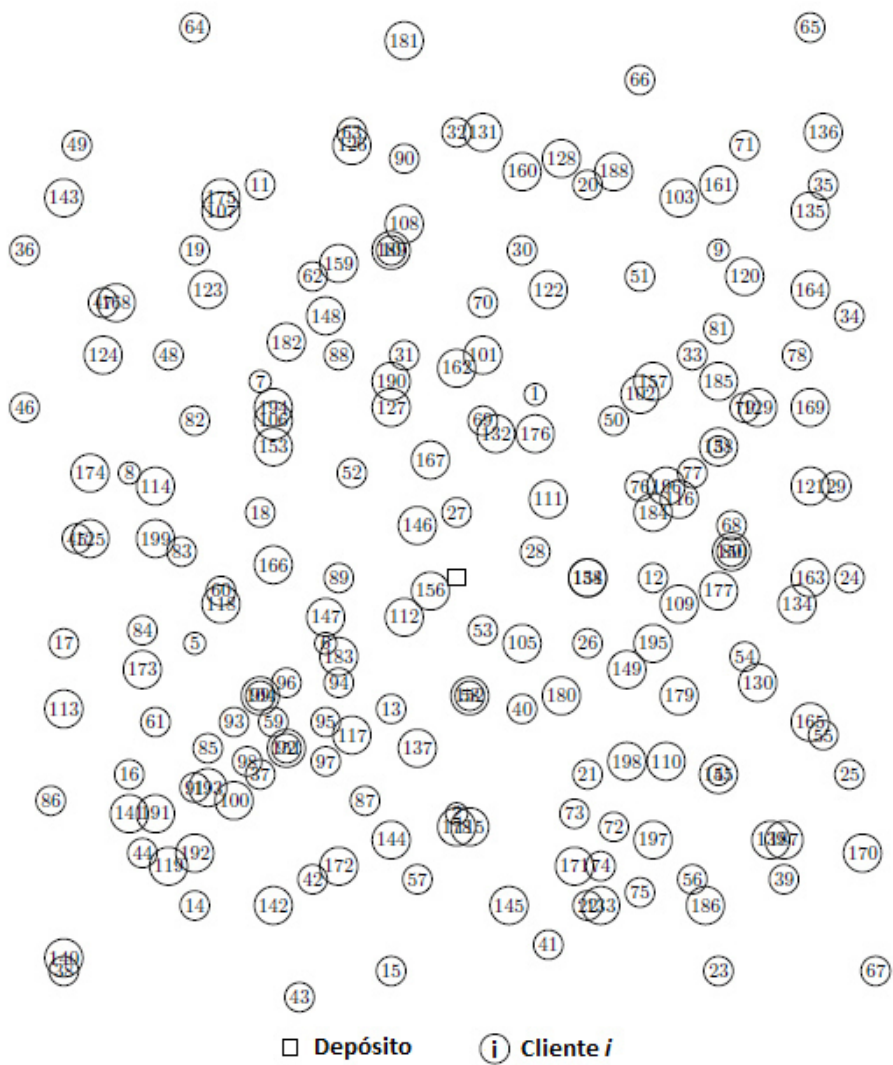


Figura B.5: Instancia CMT5

APÉNDICE C

MEJORES SOLUCIONES

En este apéndice se muestran los valores de latencia de las mejores soluciones encontradas, tanto en el enfoque periódico como en el enfoque continuo. Se expone también el tiempo que se necesitó para resolver la instancia y encontrar dichas soluciones.

Cada una de las 85 instancias con las que se experimentó se ha resuelto cinco veces. Se muestra el valor de la mejor solución considerando solo soluciones sin clientes pospuestos.

En la primera columna de cada tabla se encuentran los nombres de las instancias, en la primera fila se encuentran los valores del grado de dinamismo de cada instancia. En la columna *lat* se presenta el valor de la latencia de la mejor solución encontrada y en la columna *t* se muestra el tiempo empleado en resolver la instancia, para cada instancia con su respectivo grado de dinamismo.

	10 %		30 %		50 %		70 %		90 %	
	lat	t	lat	t	lat	t	lat	t	lat	t
CMT1	3239.6	1.2	3110.2	1.3	3033.6	1.2	2612.8	1.5	2894.5	1.1
CMT2	4930.3	1.6	4841.0	2.9	4445.6	2.4	3457.9	1.8	3309.3	2.2
CMT3	8316.4	13.4	6215.3	13.6	6274.1	12.9	6011.2	12.3	5977.1	5.7
CMT4	13877.1	35.8	1068.9	36.0	8433.3	39.8	8200.7	27.7	7545.9	19.9
CMT5	17589.3	73.2	12385.0	77.3	12180.2	79.3	12401.2	47.9	9346.8	38.9
CMT11	11370.9	25.6	11457.2	21.5	9892.4	30.4	10949.5	23.9	9250.2	12.7
CMT12	7950.1	4.5	5732.4	8.9	4545.2	12.5	4970.3	9.7	4386.7	8.7

Tabla C.1: Enfoque periódico en instancias CMT. Cuatro periodos

	10 %		30 %		50 %		70 %		90 %	
	lat	t	lat	t	lat	t	lat	t	lat	t
CMT1	3158.61	1.4	3073.57	1.5	3003.79	1.2	2936.04	1.2	2466.95	0.7
CMT2	4052.78	2.7	4456.01	3.4	4830.36	2.2	4000.6	2.1	3420.21	2.6
CMT3	7705.94	10.1	6097.32	14.6	6297.45	12.1	5914.11	11.3	6517.81	7.8
CMT4	13108.9	41.8	9352.27	38.0	9566.52	32.7	8615.07	33.9	7871.29	23.0
CMT5	16917.7	69.6	12661.6	83.2	11363.8	77.7	11194.4	70.4	9786.14	44.1
CMT11	11312.6	21.7	10177.3	27.3	9515.12	31.5	10440.1	31.1	10027.5	19.5
CMT12	7008.91	8.0	5978	9.5	4183.08	13.8	4818.78	10.2	4381.26	8.7

Tabla C.2: Enfoque periódico en instancias CMT. Seis periodos

	10 %		30 %		50 %		70 %		90 %	
	lat	t	lat	t	lat	t	lat	t	lat	t
CMT1	3239.6	1.2	3110.2	1.3	3033.6	1.2	2612.8	1.5	2894.5	1.1
CMT2	4930.3	1.6	4841.0	2.9	4445.6	2.4	3457.9	1.8	3309.3	2.2
CMT3	8316.4	13.4	6215.3	13.6	6274.1	12.9	6011.2	12.3	5977.1	5.7
CMT4	13877.1	35.8	1068.9	36.0	8433.3	39.8	8200.7	27.7	7545.9	19.9
CMT5	17589.3	73.2	12385.0	77.3	12180.2	79.3	12401.2	47.9	9346.8	38.9
CMT11	11370.9	25.6	11457.2	21.5	9892.4	30.4	10949.5	23.9	9250.2	12.7
CMT12	7950.1	4.5	5732.4	8.9	4545.2	12.5	4970.3	9.7	4386.7	8.7

Tabla C.3: Enfoque periódico en instancias CMT. Doce periodos

	10 %		30 %		50 %		70 %		90 %	
	lat	t	lat	t	lat	t	lat	t	lat	t
GWK1	102998.0	293.2	95303.2	222.0	78232.5	282.0	68142.5	166.6	56413.7	60.7
GWK5	167283.0	340.1	151226.0	344.4	149660.0	295.1	129534.0	133.5	119145.0	53.1
GWK9	8833.5	222.4	7794.0	191.0	7294.4	166.1	5984.7	62.0	5609.2	23.6
GWK10	12716.1	491.4	11153.0	416.6	10127.6	339.8	9252.6	160.6	7921.4	50.0
GWK13	8564.9	101.1	6786.2	79.9	6328.0	51.2	4582.4	25.3	4041.7	17.5
GWK14	12669.6	176.9	10643.0	162.9	8835.3	153.2	6484.4	87.4	5916.5	35.5
GWK15	18099.2	416.8	13270.9	376.2	12150.7	300.4	9434.8	143.3	7902.4	51.5
GWK17	6698.4	128.6	6413.1	80.7	5008.4	95.7	4361.6	51.1	3908.0	27.6
GWK18	9387.2	241.9	9005.3	153.4	8076.3	152.6	6365.1	99.0	5455.5	58.5
GWK19	14018.6	332.4	12416.9	348.9	10248.7	215.2	8419.2	141.9	7037.0	103.6

Tabla C.4: Enfoque periódico en instancias GWK. Cuatro periodos

	10 %		30 %		50 %		70 %		90 %	
	lat	t	lat	t	lat	t	lat	t	lat	t
GWK1	99452.8	255.1	93157.6	291.8	79428.2	274.9	69284.7	189.7	56596.2	83.7
GWK5	156356.0	368.5	141878.0	372.0	154105.0	250.1	127538.0	211.3	119781.0	51.6
GWK9	8794.6	191.0	7297.1	178.1	7294.5	154.6	5609.1	97.6	5598.4	33.6
GWK10	13322.9	417.2	11364.2	417.4	9989.2	389.5	9181.3	225.5	7931.8	60.3
GWK13	8786.0	91.1	6756.7	103.1	6003.5	149.7	4424.5	44.5	4119.5	26.9
GWK14	11351.2	236.9	9806.1	243.3	8604.5	149.7	6344.4	93.7	5700.3	54.2
GWK15	16922.1	439.7	14060.8	422.2	11790.8	390.1	9066.2	135.6	8277.1	98.2
GWK17	6685.4	164.9	6102.4	101.9	4886.4	120.9	4343.5	69.2	3883.3	42.2
GWK18	8316.4	228.3	8681.8	242.2	7798.6	201.2	6418.4*	93.126*	5386.3	89.6
GWK19	13076.8	450.6	11779.5	408.8	10142.0	270.9	7380.9	191.0	7181.7	125.6

* No se encontraron soluciones sin clientes pospuestos

Tabla C.5: Enfoque periódico en instancias GWK. Seis periodos

	10 %		30 %		50 %		70 %		90 %	
	lat	t	lat	t	lat	t	lat	t	lat	t
GWK1	90076.3	373.7	89189.4	332.3	73262.7	419.8	67679.4	192.6	57954.5	107.3
GWK5	155505.0	409.6	136245.0	493.9	148248.0	346.8	123591.0	228.7	121360.0	91.4
GWK9	8247.8	227.7	7627.0	198.9	7408.9	194.1	5749.8	113.9	5608.7	69.1
GWK10	13339.0	399.8	10942.4	513.7	9733.1	425.1	9067.4	216.5	7724.7	120.0
GWK13	8520.1	128.3	6816.6	102.3	5939.4	90.6	4639.1	59.7	4106.4	38.6
GWK14	11012.8*	242.6*	10073.3*	217.9*	8677.0	173.7	6570.3	115.9	5665.1	78.5
GWK15	16561.6	537.8	13506.3	449.6	11498.7	441.6	8942.7	254.3	7897.1	115.4
GWK17	6271.9	128.6	6110.4	115.6	4835.9	124.2	4236.9	78.7	3917.5	51.5
GWK18	9533.5	248.7	7990.6	228.9	8692.3	189.4	6609.4	139.0	5426.6	113.9
GWK19	12738.0	409.5	10899.2	402.5	9676.5	333.8	8300.5	240.4	7104.3	120.9

* No se encontraron soluciones sin clientes pospuestos

Tabla C.6: Enfoque periódico en instancias GWK. Doce periodos

BIBLIOGRAFÍA

- [1] ANGEL-BELLO, F., A. ALVAREZ y I. GARCÍA, «Two improved formulations for the minimum latency problem», *Applied Mathematical Modelling*, **37**(4), págs. 2257–2266, 2013.
- [2] ARCHETTI, C. y M. SPERANZA, «Vehicle routing problems with split deliveries», *International transactions in operational research*, **19**(1-2), págs. 3–22, 2012.
- [3] BLUM, A., P. CHALASANI, D. COPPERSMITH, B. PULLEYBLANK, P. RAGHAVAN y M. SUDAN, «The minimum latency problem», en *Proceedings of the twenty-sixth annual ACM symposium on Theory of computing*, ACM, págs. 163–171, 1994.
- [4] BLUM, C. y A. ROLI, «Hybrid metaheuristics: an introduction», en *Hybrid Metaheuristics*, Springer, págs. 1–30, 2008.
- [5] BRÄYSY, O., «A reactive variable neighborhood search for the vehicle-routing problem with time windows», *INFORMS Journal on Computing*, **15**(4), págs. 347–368, 2003.
- [6] CHEN, P., X. DONG y Y. NIU, «An iterated local search algorithm for the cumulative capacitated vehicle routing problem», en *Technology for education and learning*, Springer, págs. 575–581, 2012.
- [7] CHEN, Z.-L. y H. XU, «Dynamic column generation for dynamic vehicle routing with time windows», *Transportation Science*, **40**(1), págs. 74–88, 2006.
- [8] CHRISTOFIDES, N., «The vehicle routing problem. Combinatorial optimization. Christofides N., Mingozzi A., Toth P., Sandi C.(eds) J», , 1979.

-
- [9] CLARKE, G. U. y J. W. WRIGHT, «Scheduling of vehicles from a central depot to a number of delivery points», *Operations research*, **12**(4), págs. 568–581, 1964.
- [10] CORDEAU, J.-F. y Q. GROUPE D'ÉTUDES ET DE RECHERCHE EN ANALYSE DES DÉCISIONS (MONTRÉAL, *The VRP with time windows*, Montréal: Groupe d'études et de recherche en analyse des décisions, 2000.
- [11] CORDEAU, J.-F., G. LAPORTE, A. MERCIER *et al.*, «A unified tabu search heuristic for vehicle routing problems with time windows», *Journal of the Operational research society*, **52**(8), págs. 928–936, 2001.
- [12] CORDEAU, J.-F., G. LAPORTE, M. W. SAVELSBERGH y D. VIGO, «Vehicle routing», *Transportation, handbooks in operations research and management science*, **14**, págs. 367–428, 2006.
- [13] CROES, G. A., «A method for solving traveling-salesman problems», *Operations research*, **6**(6), págs. 791–812, 1958.
- [14] DANTZIG, G. B. y J. H. RAMSER, «The truck dispatching problem», *Management science*, **6**(1), págs. 80–91, 1959.
- [15] FISHER, M. L., «Optimal solution of vehicle routing problems using minimum k-trees», *Operations research*, **42**(4), págs. 626–642, 1994.
- [16] FLOOD, M. M., «The traveling-salesman problem», *Operations Research*, **4**(1), págs. 61–75, 1956.
- [17] FRANCIS, P. M., K. R. SMILOWITZ y M. TZUR, «The period vehicle routing problem and its extensions», en *The vehicle routing problem: latest advances and new challenges*, Springer, págs. 73–102, 2008.
- [18] FUNKE, B., T. GRÜNERT y S. IRNICH, «Local search for vehicle routing and scheduling problems: Review and conceptual integration», *Journal of heuristics*, **11**(4), págs. 267–306, 2005.
- [19] GENDREAU, M., G. LAPORTE y J.-Y. POTVIN, «Metaheuristics for the capacitated VRP», en *The vehicle routing problem*, Society for Industrial and Applied Mathematics, págs. 129–154, 2001.

- [20] GHIANI, G., F. GUERRIERO, G. LAPORTE y R. MUSMANNO, «Real-time vehicle routing: Solution concepts, algorithms and parallel computing strategies», *European Journal of Operational Research*, **151**(1), págs. 1–11, 2003.
- [21] GOEL, A. y V. GRUHN, «A general vehicle routing problem», *European Journal of Operational Research*, **191**(3), págs. 650–660, 2008.
- [22] GOLDEN, B. L., E. A. WASIL, J. P. KELLY y I.-M. CHAO, «The impact of metaheuristics on solving the vehicle routing problem: algorithms, problem sets, and computational results», en *Fleet management and logistics*, Springer, págs. 33–56, 1998.
- [23] HEMMELMAYR, V. C., K. F. DOERNER y R. F. HARTL, «A variable neighborhood search heuristic for periodic routing problems», *European Journal of Operational Research*, **195**(3), págs. 791–802, 2009.
- [24] HONG, L., «An improved LNS algorithm for real-time vehicle routing problem with time windows», *Computers & Operations Research*, **39**(2), págs. 151–163, 2012.
- [25] ICHOUA, S., M. GENDREAU y J.-Y. POTVIN, «Planned route optimization for real-time vehicle routing», en *Dynamic Fleet Management*, Springer, págs. 1–18, 2007.
- [26] KE, L. y Z. FENG, «A two-phase metaheuristic for the cumulative capacitated vehicle routing problem», *Computers & Operations Research*, **40**(2), págs. 633–638, 2013.
- [27] KYTÖJOKI, J., T. NUORTIO, O. BRÄYSY y M. GENDREAU, «An efficient variable neighborhood search heuristic for very large scale vehicle routing problems», *Computers & Operations Research*, **34**(9), págs. 2743–2757, 2007.
- [28] LAPORTE, G., M. DESROCHERS y Y. NOBERT, «Two exact algorithms for the distance-constrained vehicle routing problem», *Networks*, **14**(1), págs. 161–172, 1984.

- [29] LAPORTE, G. y F. SEMET, «Classical heuristics for the capacitated VRP», en *The vehicle routing problem*, Society for Industrial and Applied Mathematics, págs. 109–128, 2001.
- [30] LARSEN, A. y O. B. MADSEN, *The dynamic vehicle routing problem*, Tesis Doctoral, Technical University of Denmark Danmarks Tekniske Universitet, Department of Transport Institut for Transport, Logistics & ITS Logistik & ITS, 2000.
- [31] LARSEN, A., O. B. MADSEN y M. M. SOLOMON, «Recent developments in dynamic vehicle routing systems», en *The Vehicle Routing Problem: Latest Advances and New Challenges*, Springer, págs. 199–218, 2008.
- [32] LENSTRA, J. K. y A. KAN, «Complexity of vehicle routing and scheduling problems», *Networks*, **11**(2), págs. 221–227, 1981.
- [33] LETCHFORD, A. N., J. LYSGAARD y R. W. EGGLESE, «A branch-and-cut algorithm for the capacitated open vehicle routing problem», *Journal of the Operational Research Society*, **58**(12), págs. 1642–1651, 2007.
- [34] LEVIN, A. y M. PENN, «Approximation algorithm for minimizing total latency in machine scheduling with deliveries», *Discrete Optimization*, **5**(1), págs. 97–107, 2008.
- [35] LI, C.-L., G. VAIRAKTARAKIS y C.-Y. LEE, «Machine scheduling with deliveries to multiple customer locations», *European Journal of Operational Research*, **164**(1), págs. 39–51, 2005.
- [36] LUND, K., O. B. MADSEN y J. M. RYGAARD, «Vehicle routing with varying degree of dynamism.», *Informe técnico*, 1996.
- [37] LYSGAARD, J. y S. WØHLK, «A branch-and-cut-and-price algorithm for the cumulative capacitated vehicle routing problem», *European Journal of Operational Research*, **236**(3), págs. 800–810, 2014.
- [38] MLADENOVIC, N., «A variable neighborhood algorithm—a new metaheuristic for combinatorial optimization», en *papers presented at Optimization Days*, pág. 112, 1995.

- [39] MOLE, R. y S. JAMESON, «A sequential route-building algorithm employing a generalised savings criterion», *Operational Research Quarterly*, págs. 503–511, 1976.
- [40] MONTEMANNI, R., L. M. GAMBARDELLA, A. E. RIZZOLI y A. V. DONATI, «Ant colony system for a dynamic vehicle routing problem», *Journal of Combinatorial Optimization*, **10**(4), págs. 327–343, 2005.
- [41] NGUEVEU, S. U., C. PRINS y R. W. CALVO, «An effective memetic algorithm for the cumulative capacitated vehicle routing problem», *Computers & Operations Research*, **37**(11), págs. 1877–1885, 2010.
- [42] OR, I., *Traveling salesman-type combinatorial problems and their relation to the logistics of regional blood banking*, Xerox University Microfilms, 1976.
- [43] PISINGER, D. y S. ROPKE, «A general heuristic for vehicle routing problems», *Computers & operations research*, **34**(8), págs. 2403–2435, 2007.
- [44] POLACEK, M., R. F. HARTL, K. DOERNER y M. REIMANN, «A variable neighborhood search for the multi depot vehicle routing problem with time windows», *Journal of heuristics*, **10**(6), págs. 613–627, 2004.
- [45] PRINS, C., «A simple and effective evolutionary algorithm for the vehicle routing problem», *Computers & Operations Research*, **31**(12), págs. 1985–2002, 2004.
- [46] PRINS, C., «A GRASP× evolutionary local search hybrid for the vehicle routing problem», en *Bio-inspired algorithms for the vehicle routing problem*, Springer, págs. 35–53, 2009.
- [47] PRINS, C., «Two memetic algorithms for heterogeneous fleet vehicle routing problems», *Engineering Applications of Artificial Intelligence*, **22**(6), págs. 916–928, 2009.
- [48] REIMANN, M., K. DOERNER y R. F. HARTL, «D-ants: Savings based ants divide and conquer the vehicle routing problem», *Computers & Operations Research*, **31**(4), págs. 563–591, 2004.

-
- [49] RENAUD, J., G. LAPORTE y F. F. BOCTOR, «A tabu search heuristic for the multi-depot vehicle routing problem», *Computers & Operations Research*, **23**(3), págs. 229–235, 1996.
- [50] RIBEIRO, G. M. y G. LAPORTE, «An adaptive large neighborhood search heuristic for the cumulative capacitated vehicle routing problem», *Computers & Operations Research*, **39**(3), págs. 728–735, 2012.
- [51] RØPKE, S., J.-F. CORDEAU y G. LAPORTE, «Models and a branch-and-cut algorithm for pickup and delivery problems with time windows», *Networks*, **49**(4), págs. 258–272, 2007.
- [52] SILVA, M. M., A. SUBRAMANIAN, T. VIDAL y L. S. OCHI, «A simple and effective metaheuristic for the minimum latency problem», *European Journal of Operational Research*, **221**(3), págs. 513–520, 2012.
- [53] SILVER, E. A., «An overview of heuristic solution methods», *Journal of the operational research society*, **55**(9), págs. 936–956, 2004.
- [54] TOTH, P. y D. VIGO, *Vehicle routing: problems, methods, and applications*, tomo 18, Siam, 2014.
- [55] VEGA, J. M. M., M. B. M. BATISTA y J. A. M. PÉREZ, «Metaheurísticas: Una visión global», *Inteligencia artificial: Revista Iberoamericana de Inteligencia Artificial*, **7**(19), págs. 7–28, 2003.
- [56] WILSON, N. H. y N. J. COLVIN, *Computer control of the Rochester dial-a-ride system*, Massachusetts Institute of Technology, Center for Transportation Studies, 1977.
- [57] WU, B. Y., Z.-N. HUANG y F.-J. ZHAN, «Exact algorithms for the minimum latency problem», *Information Processing Letters*, **92**(6), págs. 303–309, 2004.

Ficha autobiográfica

Mayra Alejandra Luna Peña

Candidato para el grado de Maestría en Ciencias con
Orientación en Ingeniería de Sistemas

Universidad Autónoma de Nuevo León
Facultad de Ingeniería Mecánica y Eléctrica

Tesis:

Problema dinámico de rutas de vehículos centrado en el cliente

Nací el 9 de octubre de 1990 en Monterrey, Nuevo León. Realicé mis estudios de Licenciatura en Matemáticas en la Facultad de Ciencias Físico Matemáticas de la Universidad Autónoma de Nuevo León, graduándome en el año 2012. Actualmente soy candidata al grado de Maestría en Ciencias en Ingeniería de Sistemas por parte de la misma Universidad.