

Capítulo 1

Preliminares

En este capítulo se presentan los elementos sobre los cuales se construye la presente investigación. En la sección 1.1 se presenta el Problema de Enrutamiento de Vehículos (VRP por sus siglas en inglés) así como sus algunas de sus variantes. En la sección 1.2 se presentan conceptos relacionados con los criterios de vecindad y región de una vecindad, esenciales para la exploración del espacio de búsqueda en algoritmos de búsqueda local. En la sección 1.3 se introduce la metaheurística Búsqueda en Vecindades Variables (VNS por sus siglas en inglés). Por último en las secciones 1.4 y 1.5 se abordan los elementos de la biblioteca de software GLPK y el lenguaje de programación *Common Lisp*, herramientas utilizadas durante el desarrollo de este trabajo.

1.1. Problema de Enrutamiento de Vehículos (VRP)

El Problema de Enrutamiento de Vehículos (VRP) es uno de los problemas de optimización combinatoria, más estudiado por su importancia tanto práctica como teórica. Formulado por Dantzig y Ramser en 1959 [7], el VRP consiste en diseñar rutas óptimas para una flota de vehículos que deben satisfacer la demanda de un conjunto de clientes dispersos geográficamente [22]. Estos vehículos parten y regresan a un depósito, con el objetivo de minimizar un criterio de costo, como la distancia total recorrida [3]. El VRP tiene aplicaciones prácticas en áreas como la logística, el transporte urbano y la distribución de bienes en grandes cadenas de suministro [9, 22, 23], entre otras.

Una característica esencial del VRP es su clasificación como problema NP-duro [11], lo que significa que encontrar una solución óptima es computacionalmente intratable para instancias grandes. Por ello, en la práctica, se recurre a heurísticas

y metaheurísticas para obtener soluciones aproximadas con alta calidad en tiempos razonables [14].

Dado que las aplicaciones del VRP en el mundo real presentan restricciones y características específicas, han surgido múltiples variantes del problema. Entre las más estudiadas se encuentran el Problema de Enrutamiento de Vehículos con restricciones de capacidad (CVRP), el Problema de Enrutamiento de Vehículos con Ventanas de Tiempo (VRPTW), y el Problema de Enrutamiento de Vehículos con Múltiples Depósitos (MDVRP). A continuación, se describe brevemente cada uno de ellos.

1. Problema de Enrutamiento de Vehículos Capacitado (CVRP):

En esta variante, la suma de las demandas de los clientes asignados a la ruta no puede exceder la capacidad máxima de cada vehículo. El CVRP es relevante en contextos logísticos donde el volumen o peso de los productos transportados juega un rol crucial, como en la distribución de alimentos o materiales de construcción. El principal desafío en el CVRP es balancear el uso eficiente de los vehículos mientras se minimizan los costos totales, respetando las restricciones de capacidad [5].

2. Problema de Enrutamiento de Vehículos con Ventanas de Tiempo (VRPTW):

En esta variante, además de encontrar soluciones que mejoren el costo de la actual, se debe respetar un intervalo de tiempo específico durante el cual se debe visitar a un cliente para realizar la entrega. Esta variante es común en servicios de entrega, como el correo o las empresas de paquetería, donde los clientes especifican franjas horarias para recibir sus pedidos. El VRPTW introduce complejidad adicional al problema, ya que las soluciones deben sincronizar las rutas y los tiempos de servicio, minimizando penalizaciones por retrasos o tiempos de espera [18].

3. Problema de Enrutamiento de Vehículos con Múltiples Depósitos (MDVRP):

En el MDVRP, los vehículos pueden partir de múltiples depósitos en lugar de un único punto central. Esto refleja escenarios reales donde empresas con centros de distribución regionales buscan optimizar sus operaciones de manera integrada. La asignación de clientes a depósitos y la planificación de rutas eficaces son desafíos clave en esta variante [6].

Debido a su clasificación como un problema NP-Duro, se han desarrollado múltiples enfoques para resolverlos. Los métodos exactos, como la programación

lineal entera y la ramificación y acotación, son efectivos para instancias pequeñas, pero su escalabilidad es limitada [2]. En instancias más grandes, se emplean heurísticas y metaheurísticas como la búsqueda de vecindad variable (VNS) [8], los algoritmos genéticos [14] y la colonia de hormigas [12]. Estas técnicas buscan un balance entre la exploración del espacio de soluciones y la explotación de las áreas más prometedoras.

En la implementación de heurísticas y metaheurísticas para la solución del VRP el concepto de solución juega un papel importante. En la siguiente sección se presenta la definición de solución del VRP y algunas de las operaciones que se pueden hacer sobre las mismas.

1.1.1. Soluciones del VRP

De manera general una solución consiste en un conjunto de rutas r a las cuales, en la mayoría de variantes, pertenecen un grupo de clientes c como se muestra en la figura 1.1. Según las características del problema la estructura de la solución puede variar.

$$S = \begin{pmatrix} r_1 : (c_1, c_2, c_3) \\ r_2 : (c_4, c_5) \\ r_3 : (c_6, c_7, c_8) \end{pmatrix}$$

Figura 1.1: Ejemplo de solución de un VRP.

En la figura 1.1 se muestra una solución a un VRP que presenta tres rutas y ocho clientes. Los clientes c_1, c_2, c_3 pertenecen a la ruta r_1 , los clientes c_4, c_5 a la ruta r_2 y los clientes c_6, c_7, c_8 a la ruta r_3 . Cada ruta de la solución representa los viajes de un vehículo en problemas con flota homogénea, por lo tanto la cantidad de rutas de una solución equivale a la cantidad de vehículos que posee el problema.

Este tipo de representación se usa en el desarrollo de las metaheurísticas como lo son los Algoritmos Genéticos [23, 5, 14] y las Colonias de Hormigas [12]. En estos enfoques, la solución se representa como un conjunto de rutas para capturar las características del problema, como la asignación de clientes, el cumplimiento de restricciones y la evaluación del costo asociado a las rutas. Además, esta forma estructurada facilita la manipulación y modificación de las soluciones durante la exploración del espacio de búsqueda.

Luego de haber definido qué es una solución del VRP, se puede definir que es un criterio de vecindad. En el siguiente epígrafe se profundiza en este tema.

1.2. Criterios de vecindad

Formalmente, los criterios de vecindad se definen en el espacio de soluciones X como una aplicación $\mu : X \rightarrow 2^X$, tal que a cada solución $x \in X$ le asigna un conjunto de soluciones $\mu(x) \subset X$. Los elementos de este conjunto se llaman soluciones vecinas de x , mientras que el conjunto $\mu(x)$ en su totalidad se denomina vecindad de x .

En los algoritmos de búsqueda local aplicados al VRP, los criterios de vecindad guían el proceso de mejora de la solución. La elección adecuada de estos criterios influye directamente en la eficiencia de los métodos de optimización [23].

A continuación, se presentan algunos ejemplos de Criterios de Vecindad:

- **Reubicación de un cliente:** Este criterio consiste en mover un cliente de una ruta a otra o a una posición diferente dentro de la misma ruta.
- **Intercambio de clientes:** En este caso, se intercambian dos clientes entre diferentes rutas o dentro de la misma ruta.
- **Reinserción de subrutas:** Esta operación consiste en extraer un grupo de clientes consecutivos de una ruta y reinsertarlos en una nueva posición dentro de la misma ruta o en una diferente, manteniendo el su orden original.
- **Intercambio de subrutas:** Estos criterios implican modificar subrutas intercambiando los clientes que pertenecen a las mismas, manteniendo el orden que poseían en su subruta. Por ejemplo, si se quieren intercambiar dos subrutas [1,2,3] y [8,7], se mueven todos los clientes de la primera subruta hacia la segunda en el mismo orden que se encontraban, lo mismo con los de la segunda subruta, por lo que el resultado sería [8,7] y [1,2,3].

Los criterios de vecindad definen qué tan grande o pequeño puede ser el cambio entre una solución y sus vecinas, y equilibran entre dos enfoques de búsqueda: intensificación y diversificación. La intensificación se enfoca en explorar soluciones cercanas para mejorar la calidad local, mientras que la diversificación permite moverse a regiones distantes del espacio de soluciones para evitar quedarse atrapado en óptimos locales. Sin embargo, en una vecindad existen regiones que poseen un

mayor número de soluciones que mejoran a la actual, por lo tanto, explorar dichas regiones puede conducir a la solución de manera más rápida. En la siguiente sección se define que es una región.

1.2.1. Región de una vecindad

En el contexto de la exploración de vecindades, una región se define como un subconjunto del espacio de soluciones vecinas, agrupadas en función de características comunes determinadas por el criterio de vecindad aplicado [19]. Estas regiones surgen como resultado de particionar el espacio de soluciones, permitiendo realizar búsquedas focalizadas e intensivas en áreas específicas que se consideran prometedoras con respecto a encontrar soluciones óptimas o cercanas al óptimo.

La partición en regiones facilita la exploración eficiente al reducir la complejidad asociada con el análisis exhaustivo de vecindades grandes. Este enfoque divide el espacio de búsqueda en unidades manejables, lo que permite identificar y priorizar regiones con alto potencial, evitando la dispersión de los recursos computacionales en zonas menos relevantes [19].

El concepto de región no solo organiza la exploración del espacio de soluciones, sino que también se alinea con estrategias de optimización híbridas que separan fases de exploración y explotación [12]. Durante la exploración, las regiones se analizan superficialmente para clasificar su calidad. Posteriormente, en la fase de explotación, se intensifica la búsqueda dentro de las regiones clasificadas como más prometedoras, maximizando la eficiencia del proceso de optimización.

Se pueden combinar las regiones con la metaheurística Búsqueda en Vecindad Variable, permitiendo guiar la búsqueda a áreas específicas de las vecindades generadas. En la siguiente sección se describe el algoritmo Búsqueda de Vecindad Variable.

1.3. Búsqueda en Vecindades Variables (VNS)

La Búsqueda en Vecindades Variables (VNS) es una metaheurística de optimización introducida por Mladenović y Hansen en 1997 [15], diseñada para resolver problemas de optimización combinatoria y global mediante la exploración sistemática de múltiples vecindades en el espacio de soluciones. A diferencia de otros métodos de búsqueda local, que suelen trabajar con una única vecindad, VNS se basa en la idea de que cambiar dinámicamente entre diferentes estructuras de vecindad.

Esta metaheurística se basa en dos principios fundamentales:

1. Un mínimo local para una estructura de vecindad no lo es necesariamente para otra.
2. Un mínimo se considera global si es un mínimo local para cualquiera sea el criterio de vecindad.

Estos principios indican el uso de varios criterios de vecindad para resolver el problema, debido a que si se encuentra un óptimo local en uno de ellos, sea posible cambiar de criterio y continuar la exploración. En cambio, si hay un óptimo local para todos los criterios entonces se puede afirmar que es el óptimo global del problema. Esta es la idea de la VNS.

El algoritmo parte de una solución inicial y utiliza un conjunto finito de criterios de vecindad predefinidos. Durante cada iteración, se explora el entorno de la solución actual. Si se encuentra una mejora, la solución se actualiza y el proceso reinicia con la primera estructura, pero si no hay mejoras, se cambia a la siguiente vecindad hasta cumplir con las condiciones de parada.

En el contexto del VRP y sus variantes, VNS se utiliza para encontrar soluciones cercanas al óptimo [15, 8] adaptándose para cumplir las restricciones de cada variante. Por ejemplo, en el Problema de Enrutamiento de Vehículos con restricciones de capacidad (CVRP), las modificaciones suelen implicar el intercambio de clientes entre rutas o la reasignación de clientes dentro de una misma ruta [1]. En el Problema de Enrutamiento de Vehículos con Ventanas de Tiempo (VRPTW), VNS se ajusta para cumplir con las restricciones temporales, mediante operadores que reorganizan las secuencias de visitas según las ventanas de tiempo permitidas [4]. Asimismo, en variantes como el Problema de Enrutamiento de Vehículos con Múltiples Depósitos (MDVRP), los criterios de vecindad incluyen movimientos que redistribuyen clientes entre depósitos, optimizando las rutas de manera global [20].

Una de las principales ventajas de VNS es su simplicidad y flexibilidad, ya que puede integrarse fácilmente con otros enfoques de optimización [8]. Sin embargo, su efectividad depende de un diseño adecuado de las vecindades y de los operadores utilizados, ya que estos determinan la calidad de las soluciones exploradas y la capacidad del algoritmo para escapar de óptimos locales [16].

Otra herramienta para la solución de problemas de optimización es GLPK (GNU Linear Programming Kit), una biblioteca de *software* para la solución de problemas de programación lineal, de la cual se trata en la sección siguiente.

1.4. GLPK

GNU Linear Programming Kit (GLPK) es una biblioteca de *software* diseñada para resolver problemas de programación lineal (PL) y programación lineal en enteros mixtos (MIP) [13]. GLPK es parte del proyecto GNU y se distribuye bajo la Licencia Pública General de GNU [10], lo que lo convierte en una opción de código abierto ampliamente adoptada tanto en entornos académicos como industriales. Ofrece herramientas robustas y flexibles para modelar y resolver problemas de optimización combinatoria, siendo particularmente útil en aplicaciones como logística, transporte y gestión de recursos.

GLPK se centra en la resolución de problemas formulados en términos de modelos matemáticos, donde se maximiza o minimiza una función objetivo lineal sujeto a un conjunto de restricciones lineales. Estos modelos se pueden especificar utilizando el lenguaje de modelado *GNU Math Programming Language* (GMPL) [17]. GMPL permite describir problemas de manera declarativa y concisa, separando la definición del modelo de los datos específicos, lo que facilita la reutilización y la adaptación del modelo a diferentes escenarios.

Una de las características destacadas de GLPK es su implementación del método simplex revisado y del método primal-dual para resolver problemas de programación lineal [REF]. Para problemas más complejos, como los de programación lineal en enteros mixtos, GLPK utiliza un enfoque de ramificación y acotación (*branch-and-cut*). Este enfoque combina técnicas de ramificación para dividir el problema en subproblemas más pequeños y métodos de corte para eliminar regiones inviables del espacio de búsqueda, mejorando la eficiencia del proceso.

La siguiente sección está dedicada al lenguaje de programación *Common Lisp* y las razones que lo hacen eficiente para la solución del VRP. eE presente estudio forma parte de un proyecto más amplio, desarrollado en *Common Lisp*, cuyo objetivo es resolver instancias de VRP en poco tiempo.

1.5. Common Lisp

Common Lisp (LISt Processing) es un lenguaje de programación creado en la década de 1980 como un estándar para unificar las distintas variantes de *Lisp* que existían hasta ese momento. Su diseño permite trabajar con programación funcional, simbólica y orientada a objetos, lo que permite su uso en el desarrollo de inteligencia artificial, sistemas expertos y optimización matemática. En la Facultad de Matemática y Computación de la Universidad de La Habana, *Common Lisp* se usa en la investigación del VRP y sus variantes [6, 21, 3, 19, 16].

Una de las principales razones para emplear Common LISP en este contexto es su flexibilidad para prototipar rápidamente y su potente sistema de macros [3]. Estas características permiten diseñar lenguajes de alto nivel para abordar problemas específicos con un esfuerzo humano reducido, simplificando tanto la creación como la modificación de modelos y algoritmos.

En trabajos previos, LISP ha sido empleado para la generación automática de criterios de vecindad mediante lenguajes formales definidos con gramáticas libres del contexto [16]. Estos lenguajes permiten crear cadenas que representan criterios de vecindad adaptados a las características específicas de cada variante del problema. Aunque este proyecto no utiliza directamente macros ni lenguajes formales, se enmarca dentro del sistema desarrollado en la facultad, que tributa a estos avances previos.

Por lo tanto, el uso de Common LISP en este trabajo no solo continúa esta línea de investigación, sino que también aprovecha las ventajas del lenguaje para estructurar y realizar operaciones complejas en el análisis de soluciones del VRP.

En este capítulo se presentaron los elementos fundamentales sobre los cuales se construye la investigación. Se introdujo el Problema de Enrutamiento de Vehículos (VRP) y algunas de sus variantes, destacando sus múltiples aplicaciones prácticas. También se abordaron los conceptos de vecindades y regiones de una vecindad, que permiten guiar la búsqueda local en el espacio de soluciones.

Además, se presentó la metaheurística Búsqueda en Vecindades Variables (VNS), que ofrece un enfoque sistemático para explorar múltiples vecindades y superar óptimos locales. Por último, se describieron las herramientas principales empleadas en este trabajo, como la biblioteca GLPK para la resolución de modelos matemáticos y el lenguaje de programación Common Lisp. El siguiente capítulo presenta los modelos de optimización propuestos en este trabajo para explorar vecindades. Estos modelos permiten identificar soluciones óptimas dentro de las vecindades.

Bibliografía

- [1] Marwa Amous. A variable neighborhood search algorithm for the capacitated vehicle routing problem. 2017.
- [2] Roberto baldacci. Recent exact algorithms for solving the vehicle routing problem under capacity and time window constraints. 2012.
- [3] Daniela González Beltrán. Generación automática de gramáticas para la obtención de infinitos criterios de vecindad en el problema de enrutamiento de vehículos. mayo 2019.
- [4] Olli Bräysy. A reactive variable neighborhood search for the vehicle-routing problem with time windows. 2003.
- [5] Juan Pablo Orrego Cardozo. Solución al problema de ruteo de vehículos con capacidad limitada (cvrp) usando una técnica metaheurística. 2016.
- [6] Gabriela Mijenes Carrera. Análisis de vecindades en problemas de enrutamiento de vehículos con múltiples depósitos y flotas heterogéneas. noviembre 2021.
- [7] G. B. Dantzig and J. H. Ramser. The truck dispatching problem. 1959.
- [8] João Paulo Queiroz dos Santos. Reactive search strategies using reinforcement learning, local search algorithms and variable neighborhood search. 2014.
- [9] Burak Eksioglu. The vehicle routing problem: A taxonomic review. 2009.
- [10] Free Software Foundation. Gnu general public license, version 3, 2007. Last accessed: 2024-12-14.
- [11] J. K. Lenstra and A. H. G. Rinnooy Kan. Complexity of vehicle routing and scheduling problems. 1981.

- [12] Silvia Mazzeo; Irene Loiseau. An ant colony algorithm for the capacitated vehicle routing. 2004.
- [13] Andrew Makhorin. *GLPK (GNU Linear Programming Kit) Reference Manual*. Free Software Foundation, 2001. Last accessed: 2024-12-14.
- [14] Abdul Kadar Muhammad Masum. Solving the vehicle routing problem using genetic algorithm. 2011.
- [15] Nenad Mladenovic and Pierre Hansen. Variable neighborhood search. 1997.
- [16] Camila Pérez Mosquera. Primeras aproximaciones a la búsqueda de vecindad infinitamente variable. 2017.
- [17] GNU Project. Glpk modeling language (gmpl), 2024. Last accessed: 2024-12-14.
- [18] Yutao Qi. A decomposition based memetic algorithm for multi-objective vehicle routing problem with time windows. 2015.
- [19] Héctor Felipe Massón Rosquete. Exploración de vecindades grandes en el problema de enrutamiento de vehículos usando técnicas estadísticas. mayo 2020.
- [20] Mir Ehsan Hesam Sadati. A hybrid variable neighborhood search approach for the multi-depot green vehicle routing problem. 2021.
- [21] José Jorge Rodríguez Salgado. Una propuesta para la evaluación automática de soluciones vecinas en un problema de enrutamiento de vehículos a partir del grafo de evaluación de una solución. junio 2020.
- [22] Eduardo Uchoa. New benchmark instances for the capacitated vehicle routing problem. 2016.
- [23] Vidal. Hybrid genetic search for the cvrp: Open-source implementation and swap* neighborhood. 2021.