ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE CIENCIAS

PROBLEMA DE RUTEO DE VEHÍCULOS CON CAPACIDAD DE CARGA LIMITADA PARA LA RECOLECCIÓN DIVIDIDA DE MÚLTIPLES PRODUCTOS

TRABAJO DE INTEGRACIÓN CURRICULAR PRESENTADO COMO REQUISITO PARA LA OBTENCIÓN DEL TÍTULO DE INGENIERA MATEMÁTICA

YOMAR ANABELA TAPIA TORRES

yomar.tapia@epn.edu.ec

DIRECTOR: DIEGO FERNANDO RECALDE CALAHORRANO

diego.recalde@epn.edu.ec

DMQ, AGOSTO 2022

CERTIFICACIONES

Yo, YOMAR ANABELA TAPIA TORRES, declaro que el trabajo de integración curricular aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

YOMAR ANABELA TAPIA TORRES

Certifico que el presente trabajo de integración curricular fue desarrollado por YOMAR ANABELA TAPIA TORRES, bajo mi supervisión.

DIEGO FERNANDO RECALDE CALAHORRANO

DIRECTOR

DECLARACIÓN DE AUTORÍA

A través de la presente declaración, afirmamos que el trabajo de integración curricular aquí descrito, así como el(los) producto(s) resultante(s) del mismo, es(son) público(s) y estará(n) a disposición de la comunidad a través del repositorio institucional de la Escuela Politécnica Nacional; sin embargo, la titularidad de los derechos patrimoniales nos corresponde a los autores que hemos contribuido en el desarrollo del presente trabajo; observando para el efecto las disposiciones establecidas por el órgano competente en propiedad intelectual, la normativa interna y demás normas.

YOMAR ANABELA TAPIA TORRES

DIEGO FERNANDO RECALDE CALAHORRANO

RESUMEN

En el presente trabajo se estudia un Problema de Ruteo de vehículos con capacidad de carga limitada para recolecciones divididas de múltiples productos. En primer lugar, se realiza un estudio general del problema de ruteo de vehículos (VRP), y más adelante nos centramos en Problemas de recolecciones divididas. En este tipo de problemas se asume que los clientes pueden ser visitados más de una vez, que es lo contrario a lo que generalmente se supone en los VRP, donde un cliente es visitado una sola vez. Para el planteamiento del problema se considera un punto de origen en el cual se encontrarán los vehículos disponibles y un punto de destino al cual deben llegar los vehículos seleccionados. Se busca encontrar las rutas óptimas que deben seguir los vehículos con el fin de minimizar los costos de operación y de transporte, y satisfacer la demanda de cada producto. Para ello se presenta un modelo de Programación Entera que es implementado en lenguaje de Programación de Python y el solver Gurobi, en el que se obtienen soluciones con un costo computacional elevado para algunas instancia por lo que se propone además un método heurístico que encuentra soluciones casi óptimas en tiempos reducidos. Finalmente se comparan las soluciones y tiempos de ejecución del modelo entero y la heurística, y se utiliza una técnica de programación que consiste en inicializar el solver con las soluciones factibles que nos da la heurística.

ABSTRACT

In this document, a Vehicle Routing Problem with limited capacity for divided collections of multiple products is studied. First, a general study of the Vehicle Routing Problem (VRP) is made, and later we focus on Split Pickup Problems. In this type of problem, it is assumed that clients can be visited more than once, which is the opposite of what is generally assumed in VRPs, where a client is visited only once. For the approach of the problem, a point of origin is considered where the available vehicles will be found and a destination point where the selected vehicles will arrive. It is required to find the optimal routes that the vehicles must follow in order to minimize the costs of operation and transportation, and satisfy the demand for each product. For this, an Integer Programming model is presented that is implemented in the Python Programming language and the Gurobi solver, in which solutions are obtained with a high computational cost for some instances, for which a heuristic method is also proposed that finds solutions almost best in short time. Finally, the solutions and execution times of the integer programming model and the heuristics are compared, and a programming technique is used that consists of initializing the solver with the feasible solutions given by the heuristics.

Índice general

1. Introducción 1

1.1. Objetivo general . . . 2

1.2. Objetivos específicos . . . 2

1.3. Alcance . . . 2

1.4. Marco teórico . . . 3

1.4.1. Problema de Ruteo de Vehículos . . . 3

1.4.2. La Familia de VRP . . . 4

1.4.3. Otras extensiones . . . 7

1.5. Pickup and Delivery Vehicle Routing Problems (PDVRP) . . . 8

1.5.1. Split Delivery Vehicle Routing Problem . . . 9

1.5.2. Split Pickup Vehicle Routing Problem . . . 10

1.5.3. Heurísticas para el problema de ruteo de vehículos capacitados (CVRP) . . . 11

2. Metodología 12

2.0.1. Modelización del problema . . . 13

2.0.2. Verificación de la validez del modelo . . . 17

2.0.3. Heurísticas . . . 20

3. Resultados, conclusiones y recomendaciones 22

3.1. Resultados 22

3.1.1. Pruebas Computacionales 22

3.1.2. Resultados de la Heurística 24

3.1.3. Modelo con inicialización 25

3.2. Conclusiones y recomendaciones 27

3.2.1. Conclusiones 27

3.2.2. Recomendaciones 28

Bibliografía **29**

Índice de figuras

2.1. Solución instancia 1 18

Capítulo 1

Introducción

El Problema de Ruteo de Vehículos (VRP), por sus siglas en inglés, juega un papel esencial en el campo de la gestión y logística de la cadena de suministro. Para mejorar el desempeño de la cadena de suministro es necesario minimizar los costos de operaciones, dentro de las cuales se encuentra el proceso de recolección. Este proceso se da en la primera fase del sistema de la cadena de suministro, donde la materia prima es recolectada, luego procesada y posteriormente entregada a los clientes.

En este trabajo se aborda un Problema de Ruteo de Vehículos para recolecciones divididas, con varios puntos de abastecimiento y múltiples productos. Para ello se considera un punto de partida, en el cual se encontrarán los vehículos disponibles y un punto de llegada. Lo que se busca es encontrar las rutas que deben seguir los vehículos por diferentes lugares de abastecimiento recogiendo determinadas cantidades de productos, con el fin de satisfacer una demanda de cada tipo de producto minimizando los costos de transporte y costos de operación vehicular. Además, los vehículos tienen una capacidad de transporte limitada e igualmente cada punto de abastecimiento tiene una oferta limitada. Para la modelización matemática de este problema, se usa un modelo de Programación Lineal Entera; la implementación y solución se realiza usando el lenguaje de Programación de Python y el solver de optimización Gurobi [6], que es el software de optimización matemática más rápido y potente disponible para estos problemas. Finalmente, se abordan métodos heurísticos con los que se realizan comparaciones y análisis de resultados frente a los métodos exactos basados en Programación Entera.

1.1. Objetivo general

Modelizar y resolver un Problema de Ruteo de Vehículos (VRP) para recolecciones divididas y de multi-productos, mediante el uso de técnicas de Programación Entera y métodos heurísticos.

1.2. Objetivos específicos

1. Estudiar y modelizar un problema de enrutamiento de vehículos para recolecciones divididas y de múltiples productos.
2. Usar Programación Entera y métodos heurísticos para resolver el problema presentado.
3. Realizar pruebas computacionales para comparar los métodos de solución planteados.

1.3. Alcance

Esta componente se desarrollará de acuerdo con la siguiente metodología:

a) Primero se realizará una revisión bibliográfica acerca del problema presentado, que permita adaptar las ideas encontradas a las necesitadas, a fin de definir el problema de manera detallada.

b) Una vez que se realice el estudio pertinente de la literatura, se formulará el problema usando Programación Entera. Se desarrollará el modelo explicando las variables, parámetros, y cada una de las restricciones; luego, se verificará la validez del modelo en instancias pequeñas. Para ello se usará la interfaz de Python de Gurobi como lenguaje de modelización computacional.

c) Luego de la validación del modelo se crearán diferentes instancias y se presentarán las soluciones y tiempos de ejecución.

d) Se desarrollarán métodos heurísticos y se realizarán pruebas computacionales con el fin de comparar las soluciones entre el método de programación entera y el método heurístico.

e) Finalmente, se analizarán los resultados obtenidos con los dos métodos usados, y se establecerán conclusiones.

1.4. Marco teórico

1.4.1. Problema de Ruteo de Vehículos

El Problema de ruteo de vehículos (VRP) es uno de los problemas de optimización combinatoria más importantes y estudiados debido a su gran aplicabilidad en problemas del mundo real, principalmente en el sector de transporte, logística, comunicaciones, fabricación, entre otros. Este problema fue introducido por primera vez por Dantzig y Ramser en 1959 [4]. Ellos describieron una aplicación del mundo real relacionada con la entrega de gasolina a estaciones de servicio y propusieron la primera formulación de programación matemática y un enfoque algorítmico de solución. En 1964 Clarke y Wright [3] propusieron una heurística efectiva para la solución aproximada del VRP y durante las últimas décadas se han realizado numerosas investigaciones sobre diferentes variantes del Problema de Ruteo de Vehículos que se han publicado en importantes revistas como Transportation Science, Operations Research, entre otras, y presentando diversos algoritmos, heurísticas y modelos matemáticos de este problema.

La gran cantidad de aplicaciones del mundo real ha demostrado que el uso de métodos de solución para los problemas de ruteo de vehículos, tanto a nivel de planificación como operativo, produce ahorros significativos en los costos de transporte global. Los problemas de generación de rutas para vehículos tienen una gran importancia en la actualidad y una de las grandes aplicaciones de los VRP en la cual se centra este trabajo es en área de la cadena de suministro de una empresa. Las empresas necesitan administrar eficazmente la cadena que suministro para reducir al máximo sus costos sin que esto disminuya la calidad de sus productos o servicios. Es aquí donde el VRP juega un rol importante; por ejemplo, dentro de la fase de recolección de materia prima o distribución de los productos finales a sus clientes.

El Problema de Ruteo de Vehículos (VRP), o el Problema de Ruteo de Vehículos Capacitados (CVRP), generalmente se describe como un problema en el que se requiere encontrar rutas de entrega de menor costo para dar servicio a los clientes que se encuentran ubicados en diferentes puntos geográficos; las restricciones incluyen, por ejemplo, capacidad limitada de los vehículos, uno o varios depósitos a fin de satisfacer las demandas de los clientes, entre otros.

Como menciona Vidal et al. en [10], la gama extremadamente amplia de aplicaciones en las que se encuentran problemas de enrutamiento conduce a la definición de muchas variantes de VRP con características y restricciones adicionales en diferentes aspectos como: la estructura del sistema (por ejemplo, varios depósitos, varios vehículos), requisitos del cliente (por ejemplo, visitas de períodos múltiples y ventanas de tiempo dentro de un período), reglas de operación de vehículos (por ejemplo, ubicación de la carga, restricciones de ruta en distancia o tiempo total y reglas de trabajo del conductor), y contexto de decisión (por ejemplo, congestión del tráfico y planificación en horizontes de tiempo prolongados). Combinados con el CVRP tradicional, estos atributos del problema conforman un amplio dominio de investigación, desarrollo y literatura. Las dimensiones de la mayoría de las instancias de los problemas de interés práctico dificultan la aplicabilidad de métodos exactos, mientras que los pocos sistemas de software que actualmente se presentan como *sorvers* heurísticos generales son cada vez más cuestionados a medida que crece el número y la variedad de atributos.

1.4.2. La Familia de VRP

Como mencionan Toth y Vigo en [9], las variantes más importantes del VRP presentadas en la literatura en los últimos años se pueden clasificar según la estructura de la red, el tipo de solicitudes de transporte, las restricciones que afectan a cada ruta individualmente, la composición y ubicación de la flota, las restricciones entre rutas, los objetivos de optimización y características de la red. Todas estas características se describen a continuación.

Características de la Red

En el CVRP (Problema de Ruteo de Vehículos Capacitados), las tareas de transporte están relacionadas con puntos en el espacio, por ejemplo, cuando se entregan mercancías en determinadas ubicaciones; estas ubicaciones se consideran como vértices de un grafo. En este caso el VRP correspondiente es el llamado problema de enrutamiento de nodos, lo contrario a lo que se sucede en el Problema de Ruteo de Arcos (ARP). Las tareas del ARP son servicios que se proveen en segmentos de una calle o también llamados conexiones.

Tipos de solicitudes de transporte

Los tipos de solicitudes de transporte se clasifican en:

- ***Delivery and Collection***. En este problema se consideran tareas que implican el movimiento consecutivo de mercancías. Una de sus aplicaciones en el mundo real es, por ejemplo, dentro de una cadena de suministro donde se realizan tareas como: recolección de materia prima en diferentes proveedores, transporte a centros de distribución y posteriormente entrega a los domicilio de cada cliente.

  Debido a la forma en la que se planteará el problema central de este trabajo, más adelante nos centraremos en algunos detalles de este tipo de problema.

- ***Simple Visits and Vehicle Scheduling***. El servicio que se presta consiste simplemente en visitar a un cliente en una ubicación, ya no con el objetivo de recoger o entregar mercancía. Por ejemplo, las visitas que realizan los técnicos de servicio de reparación de electrodomésticos en los hogares.

- ***Alternative and Indirect Services***. En este problema, quien presta el servicio puede elegir entre diferentes alternativas, por ejemplo, para realizar la entrega de un pedido a un cliente se puede optar por su domicilio, su trabajo u otro lugar. Por lo tanto, la selección de los recorridos consiste en visitar aquellos lugares más cercanos al cliente.

- ***Point-to-Point Transportation***. Los problemas de recogida y entrega son VRP en los que las solicitudes de transporte consisten en transportes punto a punto. Es decir, cada solicitud de transporte consiste en el movimiento de bienes o personas entre dos lugares particulares, uno donde se recoge a alguien o algo, y un lugar correspondiente para la entrega.

- ***Repeated Supply***. Una de las aplicaciones de este problema en la vida real es la entrega de mercancías a supermercados, donde los clientes (supermercados) pueden exigir a la empresa entregas repetidas de mercancía, pueden ser en días diferentes siempre y cuando no se queden sin abastecimiento.

- ***Non-split and Split Services***. A diferencia de lo que se asume generalmente, donde todas las tareas de servicio las realiza un solo vehículo en una sola operación de servicio, en el *Split Delivery Vehicle Routing Problem* (SDVRP) se dividen los servicios, es decir, más de un vehículo puede visitar al mismo cliente, esto puede tener una ventaja y es que, al dividir un servicio en servicios más pequeños se pueden reducir los costos,

- ***Combined Shipment and Multi-modal Service***. En este problema, los vehículos transportan el producto desde su proveedor hasta el cliente mediante puntos de transferencia intermedios, por ejemplo, se utilizan camiones grandes para el transporte de larga distancia de carga completa desde las fábricas hasta los centros de distribución y camiones pequeños para el transporte de carga a clientes finales (ver Toth y Vigo [9]).

- ***Routing with Profits and Service Selection***. Al considerar una flota limitada puede resultar imposible cumplir con todas las solicitudes de transporte. En este caso se considera el cumplimiento de un subconjunto de solicitudes, por lo tanto, se deberá elegir aquellas solicitudes que optimicen el ruteo de vehículos y minimicen los costos.

Restricciones dentro de la ruta

Al definir las diferentes variantes de VRP se debe considerar aspectos como el tipo de restricciones que determinan si una ruta es factible o no. Estas restricciones están relacionadas con la reutilización de los vehículos, la longitud de la ruta, la carga, los horarios, y las diversas combinaciones de estos tipos de restricciones que se pueden dar en la práctica.

Características de la flota

En algunas variantes del VRP se supone que los vehículos son idénticos, pero existen casos en los que no se considera este supuesto, sino que los vehículos tendrán diferentes características en cuanto al costo, capacidad, velocidad, entre otras, y además las flotas de vehículos se encontrarán estacionadas en diferentes depósitos.

Restricciones entre rutas

En muchas variantes de VRP la viabilidad de una solución depende de las restricciones entre rutas, es decir, de como se combinen las rutas y sus horarios. Por ejemplo, las restricciones de equilibrio, donde no se considera que la diferencia de la duración máxima y mínima de la ruta no puede exceder un tiempo límite; las restricciones de recursos entre rutas, donde se pueden restringir la cantidad de rutas que tienen una determinada característica; y restricciones relacionadas con problemas de sincronización, donde las rutas y los horarios de los vehículos son interdependientes y deben coordinarse.

1.4.3. Otras extensiones

Pueden resultar problemas bastante interesantes al considerarse otras actividades logísticas, así como también diferentes objetivos de optimización. Por lo general en el VRP se utilizan objetivos de minimización de costos de enrutamiento, sin embargo pueden modelarse metas con diferentes objetivos, como objetivos jerárquicos u optimización multi-criterio. Una revisión detallada de todas las variantes de VRP descritas anteriormente se encuentra en Toth y Vigo [9].

1.5. Pickup and Delivery Vehicle Routing Problems (PDVRP)

Como se mencionó anteriormente, debido a la formulación del problema central de este proyecto es necesario precisar algunos detalles sobre el problema de recogida y entrega. Los PDVRP constituyen una familia importante de problemas de ruteo de vehículos en los que las mercancías deben transportarse desde diferentes orígenes a diferentes destinos. Los orígenes y destinos a donde se debe llevar las mercancías, representan los vértices dentro de un grafo. Los PDVRP se pueden clasificar en tres categorías principales según el tipo de demanda y la estructura de ruta que se considere. En los problemas *many-to-many* (M-M), cada producto puede tener múltiples orígenes y destinos y cualquier ubicación puede ser el origen o destino de múltiples productos. Los problemas *one-to-many-to-one* (1-M-1) se caracterizan por la presencia de algunas mercancías que se entregarán desde un depósito a muchos clientes y de otras mercancías que se recogerán en los clientes y se transportarán de vuelta al depósito. Estos tienen aplicaciones, por ejemplo, en la distribución de bebidas y la recogida de latas y botellas vacías. Finalmente, en problemas *one-to-one* (1-1), como ya se describió anteriormente, cada producto tiene un solo origen y destino entre los cuales debe ser transportado. Una de sus aplicaciones es, por ejemplo, los servicios de mensajería urbana (ver Battarra et al. [2]).

En el campo de los problemas de recogida y entrega (ver Parragh et al. [8]), se distinguen dos clases de problemas. La primera clase se enfoca en el transporte de mercancías desde el depósito hasta los clientes de transporte de línea y desde los clientes de transporte de retorno hasta el depósito. Esta clase se denomina *Vehicle Routing Problems with*

Backhauls (VRPB). En esta clase se encuentran cuatro subtipos: *Vehicle Routing Problem with Clustered Backhauls* (VRPCB - todas las líneas antes de las redes de retorno), *Vehicle Routing Problem with Mixed linehauls and Backhauls* (VRPMB - cualquier secuencia de líneas y rutas de retorno permitidas), *Vehicle Routing Problem with Divisible Delivery and Pickup* (VRPDDP - los clientes pueden ser visitados dos veces), y *Vehicle Routing Problem with Simultaneous Delivery and Pickup* (VRPSDP – los clientes que demandan estos servicios deben ser visitados exactamente una vez). En la segunda clase, se consideran todos aquellos problemas en los que se transportan mercancías entre los lugares de recogida y entrega. Estos son: *Pickup and Delivery Vehicle Routing Problem* (PDVRP - puntos de recogida y entrega no emparejados), el *Pickup and Delivery Problem* (PDP - puntos de recogida y entrega emparejados) y el problema *Dial-A-Ride* (DARP - transporte de pasajeros entre puntos de recogida y entrega emparejados y se tienen en cuenta las molestias del usuario).

Todos estos problemas mencionados anteriormente tienen que ver con procesos tanto de recogida como de entrega a la vez, sin embargo, para nuestro estudio nos enfocaremos en cada proceso de forma independiente, es decir, primero hablaremos sobre el *Split Delivery Vehicle Routing Problem*, y por otro lado trataremos el *Split Pickup Vehicle Routing Problem.*

1.5.1. Split Delivery Vehicle Routing Problem

El problema de ruteo de entrega dividida (SDVRP),por sus siglas en inglés, fue propuesto por Dror y Trudeau (1989) [5] para eliminar el supuesto de que cada cliente sea visitado una sola vez. Como mencionan Archetti, Speranza y Hertz en [1], a diferencia de la formulación general del problema del ruteo de vehículos, en el cual se tiene la condición de que un cliente sea visitado por un solo vehículo, en el SDVRP se permite que cada cliente sea visitado por más de un vehículo si esto reduce el costo total. Es posible que un cliente deba ser atendido más de una vez, al contrario de lo que generalmente se supone en los problemas de enrutamiento de vehículos. Cada vez que un vehículo visita a un cliente, recoge una cantidad entera. No se considera ninguna restricción en el número de vehículos disponibles y cada vehículo parte y regresa al depósito al final de cada recorrido. El objetivo es minimizar la distancia total recorrida por los vehículos para atender a todos los clientes.

1.5.2. Split Pickup Vehicle Routing Problem

Como hemos visto, el VRP juega un papel esencial en el campo de la gestión y logística de la cadena de suministro y por lo tanto encontrar el conjunto óptimo de rutas ayudará a tomar decisiones que reduzcan el costo de la cadena de suministro y aumentar la ganancia. Dentro de la cadena de suministro existen varias operaciones, como: el proceso de recolección de materia prima, el transporte logístico a centros de distribuciones, y finalmente la entrega a los clientes finales. Este trabajo se centra en el proceso de recolección, y una de la variantes del VRP es el problema en el cual se consideran recolecciones divididas, donde los vehículos parten desde un punto inicial, visitan diferentes lugares de abastecimiento o proveedores recolectando determinados productos, por ejemplo, para el caso de recolección de materia prima de una fábrica textil, es necesario encontrar las mejores rutas para los vehículos que minimicen los costos operativos y de transporte, hasta llegar a un punto de destino en donde será procesada la materia prima. Para este problema pueden existir pequeñas variaciones, por ejemplo, en lugar de considerar un punto de inicio diferente al punto de origen, podríamos considerar el punto de inicio como el mismo punto de destino, incluso podemos suponer diferentes puntos de inicio y diferentes depósitos, entre otros. Además, los lugares de abastecimiento pueden ser visitados por más de un vehículo, lo mismo que se considera para el caso de la entrega a los clientes.

En el presente trabajo se va a suponer que se tiene un lugar de origen en el cual se encontrarán todos los vehículos que dispone una empresa. Estos vehículos tienen una misma capacidad de carga y un costo de operación; dentro de este costo se considera el costo de mantenimiento del vehículo, contratación del chofer, entre otros. Además, estos vehículos deben pasar por determinados puntos de abastecimiento recolectando diferentes clases de productos; para cada uno de estos productos se tiene una demanda determinada, y cada punto de abastecimiento tiene una oferta limitada para cada tipo de producto. También consideramos que más de un vehículo puede pasar por un mismo punto de abastecimiento. Finalmente, lo que se busca es minimizar los costos de operación y los costos de transporte satisfaciendo la demanda.

1.5.3. Heurísticas para el problema de ruteo de vehículos capacitados (CVRP)

Como indica Vidal et al. [10], existen diferentes categorías de métodos de solución heurística para los CVRP entre ellas están: heurísticas constructivas, las cuales se caracterizan por operar de manera glotona, produciendo un conjunto de decisiones definitivas que no se pueden revertir; heurísticas de mejora local, las mismas que, basadas en una solución inicial s, exploran una vecindad $\mathcal{N}(s)$, generalmente definida por perturbaciones en s, para encontrar una solución mejorada s' que reemplace s para una nueva iteración de la heurística; y también se encuentran las metaheurísticas las cuales continúan la búsqueda más allá del primer óptimo local encontrado a través de métodos híbridos y métodos de búsqueda paralelos y cooperativos.

En este trabajo nos enfocaremos en un proceso a dos pasos; primero, calculamos una solución factible inicial basada en un procedimiento de construcción de ruta inicial. Y luego utilizaremos un modelo de programación entera en el cual fijaremos valores iniciales para cada variable.

Capítulo 2

Metodología

La investigación se compone de enfoques, tipos, técnicas e instrumentos de investigación que son útiles para llevar a cabo los objetivos planteados con eficiencia y confiabilidad. Estos métodos nos sirven de guía para recopilar y analizar información.

La metodología de investigación de este trabajo presenta tres partes: en primer lugar se desarrolla un modelo de programación entera, el cual se validará mediante algunos ejemplos. Luego, se aborda un método heurístico de solución y una comparación entre el modelo entero y heurístico. Por otro lado, tomaremos las soluciones que se obtengan del método heurístico para darle un punto de inicio al modelo entero. Finalmente, analizaremos la ejecución de cada una de estas partes y también en conjunto.

Antes de iniciar con la modelización del problema presentamos algunas definiciones básicas. Un **grafo no dirigido** G es un par ordenado $G = (V, E)$ que está formado por un conjunto finito no vacío $\{v_1, v_2, \ldots, v_n\} = V$ cuyos elementos se llaman nodos o vértices y un multi-conjunto finito $E = \{\{v_i, v_j\} : vi \in V, v_j \in V\}$ cuyos elementos se llaman aristas. Dado un grafo no dirigido G al conjunto de nodos del grafo G se nota $V(G)$, y al conjunto de aristas se nota $E(G)$. Ahora, un **grafo dirigido** G es un par ordenado (V, A) compuesto por un conjunto V de nodos y A es un conjunto de pares ordenados de nodos, denominados arcos, es decir, $A \subseteq V \times V$; este grafo se denomina también **digrafo**.

Por otro lado, un **camino** es un grafo no vacío $P = (V, A)$ de la forma $V = \{v_0, v_1, \ldots, v_k\}$ y $A = \{a_1, a_2, \ldots, a_k\}$ donde los elementos de V son distintos. Así, dado un grafo dirigido $G = (V, A)$ y un camino entre dos nodos v_0 y v_k, $P = v_0, a_1, v_1, a_2, \ldots, a_k, v_k$, el costo del camino se nota por $c(P)$ y se define por $c(P) = \sum_{i=1}^{k} c(a_i)$ donde a_i es el arco i del grafo G. También recordemos que el Problema de Caminos más Cortos (**Shortest Path Problem SPP**) se define de la siguiente manera: dado un grafo dirigido $G = (V, A)$, un nodo $r \in V$ y una función de costos sobre los arcos $c : A \longrightarrow \mathbb{R}$ se desea encontrar un conjunto de caminos de costo mínimo desde r hasta cada nodo $v \in V - \{r\}$. Esta definición es necesaria ya que se aborda una heurística en la que se usa esta idea.

Por otro lado, recordemos que un **Programa Lineal Entero** es el problema de maximizar una función lineal dada sobre el conjunto de todos los vectores que satisfacen un sistema dado de ecuaciones y desigualdades lineales y cuyas soluciones son variables enteras. Un programa lineal entero se puede transformar fácilmente a la forma:

$$\begin{aligned} \text{máx} \quad & c^T x \\ \text{s.a.} \quad & Ax \leq b, \end{aligned}$$

donde, A es una matriz $\mathbb{R}^{m \times n}$ dada, $c \in \mathbb{R}^n$, $b \in \mathbb{R}^m$ son vectores dados y $x \in \mathbb{Z}^n$ es la solución factible que satisface todas las restricciones.

2.0.1. Modelización del problema

En este trabajo se aborda un Problema de Ruteo de Vehículos para recolecciones divididas y de múltiples productos. Para ello se considera un grafo dirigido $G = (V, A)$, donde V es el conjunto de vértices que representan los lugares de abastecimiento, el nodo de origen y el nodo de destino. A es el conjunto de arcos que representan las rutas directas entre un nodo i y un nodo j. Notaremos por M al conjunto de vehículos disponibles los cuales se encontrarán inicialmente en el nodo O y deberán llegar al nodo de destino D, y notaremos R al conjunto de productos. Cada vehículo m tiene una capacidad de carga notada por Q y un costo de operación C_m. Además, notaremos como tc_{ij} al costo de transporte desde el nodo i al nodo j y o_{ir} a la oferta de producto r en el nodo i. El objetivo es encontrar las rutas que minimicen los costos de operación y transporte satisfaciendo la demanda d_r de cada producto.

Para modelar el Problema de Ruteo de Vehículos para recolecciones divididas y de múltiples productos, usamos los siguientes conjuntos y variables de decisión:

Conjuntos:

- V : Conjunto de lugares de abastecimiento, $|V|$=n.
- $V_O = V \cup O$
- $V_D = V \cup D$
- A : Conjunto de arcos entre los abastecimientos.
- M: Conjunto de vehículos disponibles.
- R : Conjunto de productos.

Parámetros:

- $Q \in \mathbb{Z}^+$: capacidad máxima de carga de los vehículos.
- $o_{ir} \in \mathbb{Z}^+$: oferta en el nodo i del producto r
- $tc_{ij} \in \mathbb{R}^+$: costo de transporte del nodo i al nodo j.
- $C_m \in \mathbb{R}^+$: costo de operación del vehículo m.
- $d_r \in \mathbb{Z}^+$: demanda del producto r.

Cabe indicar que los costos de operación de los vehículos varían a pesar de que tienen la misma capacidad y esto se puede justificar en el mundo real, por la diferencia de los años de uso de cada vehículo, su velocidad, el costo de neumáticos para mantenimiento, o incluso por el funcionamiento a Diesel o a gasolina.

Variables de decisión:

Consideraremos las siguientes variables: $x_{ij}^m \in \{0,1\}$ que toma el valor de 1 si el vehículo m se mueve del nodo de recogida i al nodo de recogida j y vale 0 en el caso contrario; $y_i^m \in \{0,1\}$ que toma el valor 1 si el vehículo m visita el nodo de recogida i y vale 0 caso contrario; $p_{ir}^m \in \mathbb{Z}^+$ que es el peso del producto r recogido en el nodo i por el vehículo m y $u_i^m \in [n]$ que indica la posición del nodo i para cada vehículo.

Modelo 1

$$
\begin{aligned}
\text{mín} \quad & \sum_{i\in V_O}\sum_{j\in V_D}\sum_{m\in M} tc_{ij}x_{ij}^m + \sum_{m\in M}\sum_{j\in V} C_m x_{Oj}^m && && (1)\\
\text{s.a.} \quad & \sum_{j\in V} x_{Oj}^m \leq 1, && \forall m \in M, && (2)\\
& \sum_{j\in V} x_{jD}^m \leq 1, && \forall m \in M, && (3)\\
& \sum_{j\in V_O} x_{ji}^m = y_i^m, && \forall m \in M, i \in V, && (4)\\
& \sum_{i\in V_O} x_{ik}^m = \sum_{j\in V_O} x_{kj}^m, && \forall m \in M, k \in V, && (5)\\
& u_j^m \geq u_i^m + (1+n)x_{ij}^m - n, && \forall m \in M, \forall (i,j) \in A, i \neq O, j \neq D, && (6)\\
& \sum_{m\in M}\sum_{j\in V} x_{Oj}^m \leq M, && && (7)\\
& \sum_{m\in M} p_{ir}^m \leq o_{ir}, && \forall i \in V, \forall r \in R, && (8)\\
& \sum_{i\in V}\sum_{r\in R} p_{ir}^m \leq Q, && \forall m \in M, && (9)\\
& p_{ir}^m \leq o_{ir} y_i^m, && \forall i \in V, \forall m \in M, \forall r \in R, && (10)\\
& \sum_{i\in V}\sum_{m\in M} p_{ir}^m = d_r, && \forall r \in R, && (11)\\
& x_{ij}^m \in \{0,1\} && \forall (i,j) \in A, \forall m \in M, && (12)\\
& y_i^m \in \{0,1\} && \forall i \in V, \forall m \in M, && (13)\\
& p_{ir}^m \in \mathbb{Z}^+ && \forall i \in V, \forall r \in R, \forall m \in M, && (14)\\
& u_i \in [n]. && && (15)
\end{aligned}
$$

La función objetivo minimiza el costo total de operación y transporte determinando la mejor ruta y el número óptimo de vehículos utilizados. La restricción (2) garantiza que si se elige un vehículo, este no puede partir desde O a más de un nodo a la vez. La restricción (3) asegura que un vehículo no puede llegar al destino desde más de una ruta. La restricción (4) muestra que si se visita el nodo i, el vehículo m solo puede entrar por una ruta. La restricción (5) es una restricción de conservación de flujo y garantiza el movimiento consecutivo de vehículos, es decir, todo vehículo que entra al nodo i tiene que salir. La restricción (6) indica que para cada vehículo y cada arco (i, j) cuyos dos extremos sean distintos al nodo O, si el arco es seleccionado dentro de la solución, entonces debe cumplirse que $u_j^m \geq u_i^m + 1$, es decir, indica la posición de cada nodo dentro del tour, asumiendo que el nodo O ocupa la primera posición. Recordemos que, una manera alternativa de eliminar subciclos consiste en usar las conocidas restricciones de corte, que requiere un número exponencial de restricciones. Sin embargo, otra alternativa que requiere únicamente un número polinomial de restricciones fue propuesta en 1960 por Miller-Tucker-Zemlin [7] y consiste en introducir variables auxiliares de ordenamiento, que indican la posición de cada nodo dentro del tour; esta última es la alternativa que se usará en el desarrollo de este trabajo. La restricción (7) muestra que el número de vehículos que parten del nodo O no puede exceder al número de vehículos disponibles. La restricción (8) garantiza que la suma de los pesos del producto r recogidos por todos los vehículos que pasan por el nodo i debe ser menor o igual a la oferta de producto r en ese nodo. La restricción (9) muestra que la suma total de los pesos de los productos recogidos por el vehículo m, de los diferentes nodos por lo que pase este vehículo, no puede exceder el peso máximo que puede transportar el vehículo. La restricción (10) es una restricción de activación y garantiza que si no se pasa por el nodo i entonces no se recoge ningún producto. La restricción (11) garantiza que la cantidad total de producto r recogida por todos los vehículos debe ser igual a la demanda de este producto y las restricciones (12), (13), (14) y (15) corresponden a los conjuntos de pertenencia de las variables de decisión.

2.0.2. Verificación de la validez del modelo

Para validar el modelo planteado se lo implementó en el lenguaje de programación Python con el solver Gurobi y se consideró una instancia pequeña de 4 nodos, 10 aristas, 3 vehículos de capacidad 200, 3 productos y los siguientes datos:

	Producto 1	Producto 2	Producto 3
Nodo 1	35	30	10
Nodo 2	30	50	65
Nodo 3	45	20	25
Nodo 4	70	75	60

Cuadro 2.1: Oferta del producto p en el nodo i

	Demanda
Producto 1	170
Producto 2	110
Producto 3	120

Cuadro 2.2: Demanda de los productos

arco	costo	arco	costo
O-1	87	3-1	70
O-3	80	3-4	65
1-3	64	4-1	68
1-4	73	4-2	78
2-D	85	4-D	78

Cuadro 2.3: Costos de transporte

	Costo Operativo ($)
Vehículo 1	250
Vehículo 2	270
Vehículo 3	300

Cuadro 2.4: Costos de operación vehicular

Resultados de la solución

El solver tardó 0.11 segundos en encontrar la solución óptima, con una función objetivo de 1066 dólares. En las siguiente tabla se muestra el proceso de recolección de cada vehículo:

Vehículos	Rutas	Cantidad de producto (P1,P2,P3)
Vehículo 1	O - 1	(0,0,0)
	1 - 4	(35,30,0)
	4 - 2	(105,30,0)
	2 - D	(125,40,35)
Vehículo 2	0 - 3	(0,0,0)
	3 - 4	(45,20,25)
	4 - D	(45,70,85)

Cuadro 2.5: Ejemplo 1: rutas y cantidad de producto transportado

Para analizar mejor los resultados, se puede observar en la Figura 2.1 la oferta de cada producto en cada nodo, así como la cantidad de producto recogida por cada vehículo. Además, se observa que la cantidad recogida por cada vehículo no sobrepasa su capacidad y la oferta de cada nodo. Finalmente, los vehículos llegan al lugar de destino y se satisface la demanda. También podemos notar que se cumple la restricción de que un nodo puede ser visitado por más de un vehículo a la vez, como sucede con el nodo 4 de la Figura 2.1.

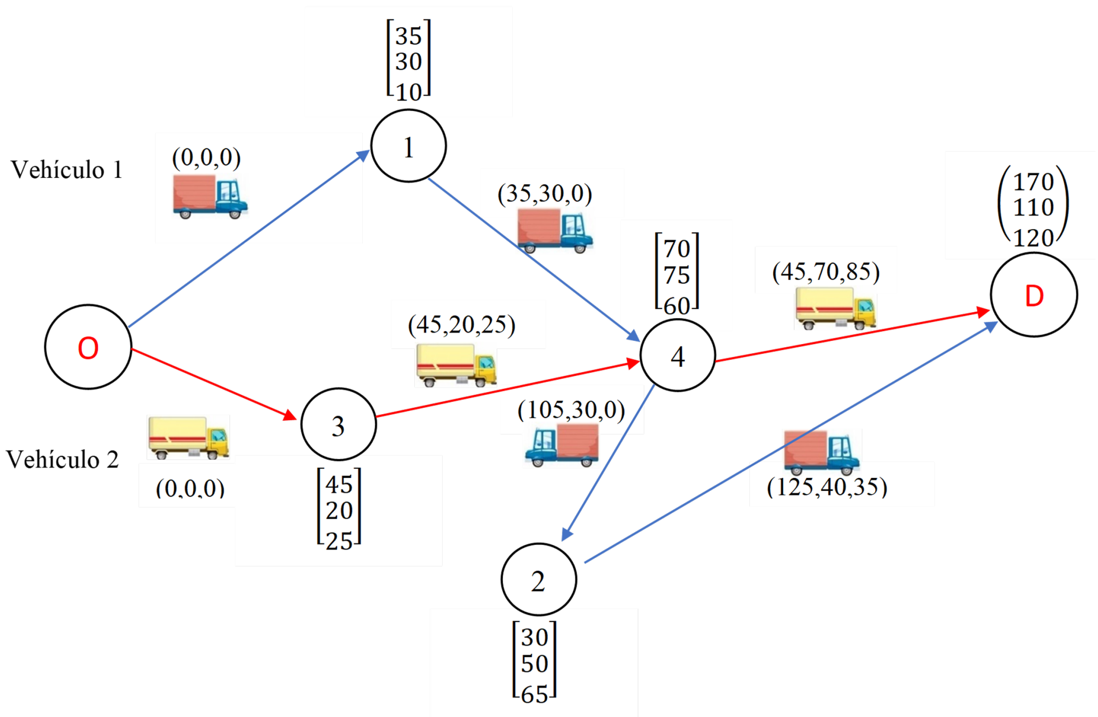

Figura 2.1: Solución instancia 1

Ahora, veamos un nuevo ejemplo para una instancia de 8 nodos, 26 aristas, 6 vehículos y 3 productos. De igual manera vamos a suponer que los vehículos tienen una capacidad Q=200. Consideremos los siguientes datos:

	Producto 1	Producto 2	Producto 3
Nodo 1	86	99	86
Nodo 2	84	93	98
Nodo 3	73	87	89
Nodo 4	76	66	70
Nodo 5	72	81	93
Nodo 6	80	82	96
Nodo 7	99	81	87
Nodo 8	84	96	82

Cuadro 2.6: Oferta del producto p en el nodo i

	Demanda
Producto 1	295
Producto 2	258
Producto 3	287

Cuadro 2.7: Demanda de los productos

arco	costo	arco	costo	arco	costo	arco	costo
O-3	129	2-4	123	4-2	123	8-1	135
O-4	120	2-6	120	5-3	139	8-2	117
O-7	126	2-8	117	5-6	97	8-5	95
1-3	96	3-1	96	5-8	95	8-6	110
1-8	135	3-2	101	6-2	120	8-D	94
1-D	89	3-5	139	6-5	97		
2-3	101	3-D	91	6-8	110		

Cuadro 2.8: Costos de transporte

	Costo Operativo ($)
Vehículo 1	472
Vehículo 2	360
Vehículo 3	345
Vehículo 4	282
Vehículo 5	369
Vehículo 6	266

Cuadro 2.9: Costos de operación vehicular

Resultados de la solución

El solver tardó 34.96 segundos en encontrar la solución óptima, con una función objetivo de 3246 dólares. En las siguiente tabla se muestra el proceso de recolección de cada vehículo:

Vehículos	Rutas	(P1,P2,P3)
Vehículo 2	O-4	(0,0,0)
	4-2	(53,8,70)
	2-3	(54,8,138)
	3-D	(54,8,138)
Vehículo 3	O-3	(0,0,0)
	3-D	(0,40,0)
Vehículo 4	O-3	(0,0,0)
	3-D	(73,38,89)
Vehículo 5	O-4	(0,0,0)
	4-2	(23,1,0)
	2-3	(106,94,0)
	3-D	(106,94,0)
Vehículo 6	O-3	(0,0,0)
	3-1	(0,8,0)
	1-D	(62,78,60)
TOTAL		(295,258,287)

De la misma manera podemos observar en la tabla anterior que la cantidad recogida por cada vehículo no sobrepasa la oferta de cada nodo y su capacidad. Además, los vehículos llegan al lugar de destino y se satisface la demanda. Con estos dos ejemplos podemos ver que el modelo arroja soluciones que cumplen con todas las restricciones planteadas en nuestro problema.

2.0.3. Heurísticas

Debido al costo computacional alto que por lo general se presentan en los Problemas de Ruteo de Vehículos, conforme se incrementa el tamaño de las instancias, se necesita métodos de solución eficientes. Es por ello que en esta sección abordamos un método heurístico y posteriormente se presentan los resultados de su ejecución.

La idea de este algoritmo es encontrar un conjunto de caminos más cortos que vayan de O a D, que cumpla la condición de que la oferta total del camino es suficiente para llenar el vehículo o satisfacer la demanda total. Una vez encontrado este conjunto de caminos, enrutamos los vehículos necesarios para transportar toda la oferta posible de cada nodo hasta cumplir con la demanda de cada producto. Para encontrar el conjunto de caminos más cortos utilizamos el algoritmo de $k - caminos$ más cortos propuesto por Jin Y, en [11].

Algorithm 1 Heurística de k-caminos

```
costo=0
costo_Op=0
visited_v=[]
used_paths=[]
function try_demanda(path_list,v)
    for path in path_list do
        oferta = oferta_path(path)
        if demanda < Q then
            if demanda ≤ oferta then
                visited_v=path
            else
                buscar otro camino
        else
            if Q ≤ sum(oferta) then
                visited_v= path
            else
                buscar otro camino
        extraer oferta de path hasta llenar el vehículo o satisfacer la
        demanda
        costo += costo[path]
        return True
    return False
for v in vehículos_seleccionados do
    if sum(demanda)=0 then
        break
    while not Try_demanda(used_paths,v) do
        buscar el siguiente camino más corto de 'O' a 'D' y añadirlo a
        used_paths
for i in vehículos_seleccionados do
    costo_Op+=costo_operativo
costo_total=costo+costo_Op
```

Capítulo 3

Resultados, conclusiones y recomendaciones

3.1. Resultados

3.1.1. Pruebas Computacionales

En esta sección se presentan los resultados de las pruebas computacionales que se realizaron usando el solver Gurobi 9.1.2 con la interfaz de Python GurobiPy, y en una computadora portátil Windows 10 Home 64-bi, con un procesador AMD Ryzen 3 2300U Cores 4 y 16 GB de RAM. Además, para los datos de entrada se utilizaron números aleatorios en los siguientes intervalos: $tc_{ij} \in [80, 150], C_m \in [250, 500], o_{ir} \in [65, 100], d_r \in [250, 300]$ y $Q = 200$, pues como se mencionó anteriormente, se consideran vehículos con la misma capacidad de carga.

A continuación, se reporta la función objetivo, GAP de optimalidad de ejecución y tiempo (segundos). Además, cada instancia está caracterizada por (Nodos, Aristas, Vehículos, Productos) y se fija un tiempo límite de ejecución de 3600 segundos.

		Modelo 1		
Prueba	Instancia	Obj. Value	Gap	Tiempo (seg)
1	[4, 20, 4, 2]	1937	0.00	1.60
2	[4, 11, 5, 2]	2498	0.00	0.14
3	[4, 20, 5, 2]	1937	0.00	2.13
4	[5, 17, 4, 2]	1841	0.00	0.57
5	[5, 30, 4, 2]	1736	0.00	34.88
6	[5, 30, 5, 2]	1668	0.00	56.46
7	[5, 17, 5, 3]	2986	0.00	1.13
8	[5, 17, 8, 3]	2839	0.00	81.90
9	[6, 30, 5, 3]	2960	0.00	230.11
10	[6, 16, 6, 3]	3174	0.00	0.30
11	[6, 25, 6, 3]	2955	0.00	457.76
12	[7, 24, 6, 3]	3293	0.00	26.57
13	[7, 24, 7, 3]	3199	0.00	41.55
14	[7, 35, 6, 3]	3496	0.00	793.04
15	[8, 40, 6, 3]	2184	0.00	2006.92
16	[8, 26, 6, 3]	3246	0.00	34.99
17	[9, 26, 7, 4]	3419	0.00	15.14
18	[9, 37, 9, 4]	3050	10.23	3600.08
19	[9, 42, 7, 4]	3108	18.76	3600.09
20	[9, 42, 9, 4]	3027	22.53	3600.06
21	[10, 29, 7, 4]	4553	0.00	153.00
22	[10, 29, 8, 4]	4418	0.00	529.28
23	[10, 34, 8, 4]	4206	0.00	3032.98
24	[11, 40, 8, 4]	3893	22.53	3600.08
25	[11, 40, 10, 4]	3851	30.98	3600.06
26	[12, 44, 7, 4]	3464	9.09	3600.05
27	[15, 114, 11, 3]	2652	65.61	3600.09
28	[18, 167, 11, 4]	3091	73.37	3600.09
29	[19, 165, 12, 5]	3373	66.65	3600.08
30	[21, 234, 13, 5]	4052	74.43	3600.15
31	[23, 268, 14, 6]	4490	73.54	3600.16
32	[24, 281, 15, 6]	4365	74.59	3600.09

Cuadro 3.1: Instancias Modelo 1

Como podemos ver en la anterior tabla, para las primeras instancias el solver encuentra la solución en un intervalo muy reducido de tiempo, y conforme se incrementa el número de vehículos, nodos, aristas y productos, el tiempo computacional aumenta considerablemente. Para instancias relativamente pequeñas como la instancia 18,19,20 y a partir de la 24, el costo computacional es elevado, lo que motiva el uso de técnicas heurísticas para la búsqueda de soluciones casi óptimas en tiempos reducidos. Además podemos notar, por ejemplo, que en la instancia 2 y 3, se tiene el mismo número de nodos, vehículos y productos pero en la instancia 3 se incrementa el número de aristas, y por tanto, la función objetivo disminuye debido a que es posible encontrar una mejor ruta. También podemos observar las instancias 7 y 8, donde se fijan el mismo número de nodos, arcos y productos, pero para la instancia 8 se incrementa el número de vehículos, y es por eso que, la función objetivo es menor pues existe la opción de elegir vehículos que posiblemente tengan menor costo operativo. De la misma manera sucede en algunos casos cuando incrementamos el número de nodos, ya que se puede elegir visitar nodos con menor costo de transporte.

3.1.2. Resultados de la Heurística

Implementando el pseudocódigo presentado en el capítulo 2 en Lenguaje de Programación de Python se obtuvieron los siguientes resultados, donde podemos comparar las soluciones y los tiempos de ejecución de las instancias usando la heurística y el modelo entero, con sus errores relativos.

		Modelo 1			Heurística 1		Error relativo
No.	Instancia	Obj.	Gap	Tiempo	Obj.	Tiempo	(%)
1	[4, 20, 4, 2]	1937	0.00	1.60	2139	0.40	10.43
2	[4, 11, 5, 2]	2498	0.00	0.14	2498	0.22	0.00
3	[4, 20, 5, 2]	1937	0.00	2.13	2139	0.38	10.43
4	[5, 17, 4, 2]	1841	0.00	0.57	1949	0.39	5.87
5	[5, 30, 4, 2]	1736	0.00	34.88	1827	0.44	5.24
6	[5, 30, 5, 2]	1668	0.00	56.46	1759	0.45	5.46
7	[5, 17, 5, 3]	2986	0.00	1.13	3015	0.29	0.97
8	[5, 17, 8, 3]	2839	0.00	81.90	2839	0.32	0.00
9	[6, 30, 5, 3]	2960	0.00	230.11	2960	0.37	0.00
10	[6, 16, 6, 3]	3174	0.00	0.30	3279	0.34	3.31
11	[6, 25, 6, 3]	2955	0.00	457.76	2981	0.39	0.88
12	[7, 24, 6, 3]	3293	0.00	26.57	3373	0.36	2.43
13	[7, 24, 7, 3]	3199	0.00	41.55	3279	0.40	2.50
14	[7, 35, 6, 3]	3496	0.00	793.04	3578	0.51	2.35
15	[8, 40, 6, 3]	2184	0.00	2006.92	2184	0.45	0.00
16	[8, 26, 6, 3]	3246	0.00	34.99	3340	0.42	2.90
17	[9, 26, 7, 4]	3419	0.00	15.14	3436	0.45	0.50
18	[9, 37, 9, 4]	3050	10.23	3600.08	3142	0.50	3.02
19	[9, 42, 7, 4]	3108	18.76	3600.09	3113	0.57	0.16
20	[9, 42, 9, 4]	3027	22.53	3600.06	3180	0.62	5.05
21	[10, 29, 7, 4]	4553	0.00	153.00	4553	0.45	0.00
22	[10, 29, 8, 4]	4418	0.00	529.28	4418	0.41	0.00
23	[10, 34, 8, 4]	4206	0.00	3032.98	4246	0.47	0.95
24	[11, 40, 8, 4]	3893	22.53	3600.08	3924	0.72	0.80
25	[11, 40, 10, 4]	3851	30.98	3600.06	3851	0.55	0.00
26	[12, 44, 7, 4]	3464	9.09	3600.05	3565	0.57	2.92
27	[15, 114, 11, 3]	2652	65.61	3600.09	2652	1.09	0.00
28	[18, 167, 11, 4]	3091	73.37	3600.09	3191	1.46	3.24
29	[19, 165, 12, 5]	3373	66.65	3600.08	3410	1.27	1.10
30	[21, 234, 13, 5]	4052	74.43	3600.15	4183	1.65	3.23
31	[23, 268, 14, 6]	4490	73.54	3600.16	4585	2.12	2.12
32	[24, 281, 15, 6]	4365	74.59	3600.09	4499	3.66	3.07

Cuadro 3.2: Comparación Modelo 1 y Heurística 1

Podemos ver que usando el método heurístico hemos encontrado soluciones muy cercanas al óptimo y óptimas en algunos casos, por ejemplo, en las instancias 2, 8, 9, 15, 21, 22. También se puede observar en las instancias 25 y 27 que la heurística en aproximadamente un segundo, alcanza las mismas soluciones que el solver alcanza en 3600 segundos. En general los tiempos computacionales son bastante reducidos en comparación con los del solver aplicado al modelo entero. En la última columna observamos los errores relativos, que nos indican la calidad de la mayoría de las soluciones que se han presentado usando el método heurístico.

3.1.3. Modelo con inicialización

Una posibilidad para acelerar la solución de modelos de programación entera computacionalmente difíciles consiste en inicializar el solver con una solución factible de buena calidad, la cual puede encontrarse por métodos heurísticos. Es así que, debido al costo computacional que representa resolver el problema planteado, pretendemos mejorar los tiempos de ejecución y para ello, a partir de los resultados obtenidos por la heurística presentada, se inicializaron las variables en el modelo y se obtuvo la siguiente tabla:

		Modelo 1			Modelo 1 con inicio		
No.	Instancia	Obj.	Gap	Tiempo	Obj.	Gap	Tiempo
1	[4, 20, 4, 2]	1937	0.00	1.60	1937	0.00	0.74
2	[4, 11, 5, 2]	2498	0.00	0.14	2498	0.00	0.15
3	[4, 20, 5, 2]	1937	0.00	2.13	1937	0.00	1.94
4	[5, 17, 4, 2]	1841	0.00	0.57	1841	0.00	0.49
5	[5, 30, 4, 2]	1736	0.00	34.88	1736	0.00	7.33
6	[5, 30, 5, 2]	1668	0.00	56.46	1668	0.00	33.13
7	[5, 17, 5, 3]	2986	0.00	1.13	2986	0.00	0.71
8	[5, 17, 8, 3]	2839	0.00	81.90	2839	0.00	35.20
9	[6, 30, 5, 3]	2960	0.00	230.11	2960	0.00	22.70
10	[6, 16, 6, 3]	3174	0.00	0.30	3174	0.00	0.41
11	[6, 25, 6, 3]	2955	0.00	457.76	2955	0.00	50.63
12	[7, 24, 6, 3]	3293	0.00	26.57	3293	0.00	16.51
13	[7, 24, 7, 3]	3199	0.00	41.55	3199	0.00	37.40
14	[7, 35, 6, 3]	3496	0.00	793.04	3496	0.00	315.68
15	[8, 40, 6, 3]	2184	0.00	2006.92	2184	0.00	966.11
16	[8, 26, 6, 3]	3246	0.00	34.99	3246	0.00	15.56
17	[9, 26, 7, 4]	3419	0.00	15.14	3419	0.00	10.19
18	[9, 37, 9, 4]	3050	10.23	3600.08	3050	0.00	3325.67
19	[9, 42, 7, 4]	3108	18.76	3600.09	3108	0.00	3431.67
20	[9, 42, 9, 4]	3027	22.53	3600.06	3027	20.58	3600.07
21	[10, 29, 7, 4]	4553	0.00	153.00	4553	0.00	36.98
22	[10, 29, 8, 4]	4418	0.00	529.28	4418	0.00	139.02
23	[10, 34, 8, 4]	4206	0.00	3032.98	4206	0.00	603.61
24	[11, 40, 8, 4]	3893	22.53	3600.08	3893	17.39	3600.06
25	[11, 40, 10, 4]	3851	30.98	3600.06	3851	26.43	3600.07
26	[12, 44, 7, 4]	3464	9.09	3600.05	3464	0.00	2396.58
27	[15, 114, 11, 3]	2652	65.61	3600.09	2652	66.37	3600.09
28	[18, 167, 11, 4]	3091	73.37	3600.09	3091	73.92	3600.11
29	[19, 165, 12, 5]	3373	66.65	3600.08	3373	66.26	3600.09
30	[21, 234, 13, 5]	4052	74.43	3600.15	4052	73.47	3600.13
31	[23, 268, 14, 6]	4490	73.54	3600.16	4490	70.60	3600.25
32	[24, 281, 15, 6]	4365	74.59	3600.09	4365	71.55	3600.25

Cuadro 3.3: Instancias Modelo 1 con inicialización

Podemos observar en el Cuadro 3.3 que al dar una solución inicial al modelo se obtienen tiempos menores en comparación con el modelo entero en el que no se dio un punto inicial; los GAPs disminuyen para algunas instancias, sin embargo, para instancias relativamente grandes, como por ejemplo, a partir de la instancia 27 los GAPs siguen siendo bastante elevados.

El código con la implementación del modelo, la heurística y las instancias pueden ser encontradas en el repositorio Códigos.

3.2. Conclusiones y recomendaciones

3.2.1. Conclusiones

- Hablar de Problemas de Ruteo de Vehículos implica un estudio bastante amplio ya que actualmente estos problemas tienen una gran aplicabilidad en el mundo real; existen muchas actividades y servicios que se pueden optimizar, como es el caso del sector logístico de una cadena de suministro, particularmente en el área de recolección que es en lo que nos hemos enfocado en este trabajo.

- Se ha abordado un problema de ruteo de vehículos con capacidad limitada para recolecciones divididas de múltiples productos que forma parte de los conocidos problemas *Split Pickup and Delivery Vehicle Routing Problems* (VRPSDP) por sus siglas en inglés. Estos problemas tienen la ventaja de que en algunos casos al dividir los servicios en servicios más pequeños ayudan a reducir costos.

- Se encontró un modelo de Programación Entera para el Problema de Ruteo de Vehículos con capacidad de carga limitada para la recolección dividida de múltiples productos, sin embargo, a partir de las soluciones que se obtuvieron, podemos concluir que el modelo entero no resulta eficiente para instancias relativamente pequeñas, pues se alcanzan GAP bastante elevados. Por lo tanto, fue necesario buscar una técnica heurística eficiente que encuentre soluciones casi óptimas en tiempos reducidos.

- Se realizó la validación del modelo entero presentado mediante dos ejemplos, y se obtuvieron resultados que cumplen con todas las restricciones del problema. Además se realizaron pruebas computacionales para 32 instancias.

- Al comparar las dos técnicas de solución usadas en este trabajo, se obtuvieron los errores relativos que indican que las soluciones del método heurístico son de buena calidad, es decir hemos encontrado una heurística que encuentra soluciones casi óptimas en intervalos muy reducidos de tiempo.

- A partir de las soluciones factibles que se obtuvieron con la heurística se inicializó el modelo entero, lo que redujo en general los tiempos de ejecución, sin embargo, para algunas instancias el GAP sigue siendo elevado.

3.2.2. Recomendaciones

- Algunas variaciones interesantes del Problema de Ruteo de Vehículos con capacidad de carga limitada para la recolección dividida de múltiples productos se pueden plantear utilizando camiones con diferentes capacidades de carga que lleguen simultáneamente al destino y también considerando ventanas de tiempo de recogida o de llegada al destino.

- Debido a que el modelo entero no resulta eficiente para instancias relativamente pequeñas se recomienda usar técnicas de mejoramiento para el modelo de Programación Entera planteado.

- También se recomienda realizar mas pruebas computacionales para diferentes instancias lo que podría ayudar a analizar otros casos particulares variando el número de nodos, aristas, productos y vehículos, ya que en este trabajo solamente hemos considerado 32 instancias.

- Aunque hemos obtenido resultados buenos con la heurística presentada, siempre puede existir nuevas formas de mejorarla, por tanto, se sugiere profundizar el estudio de heurísticas para este tipo de problemas.

- Para niveles posteriores a pregrado, se recomienda realizar un estudio más amplio sobre los problemas de ruteo de vehículos para recolección divididas, ya que no es un problema tan abordado en las investigaciones actuales, pero puede resultar más ventajoso que los problemas en los cuales no se dividen las cargas o servicios.

Referencias bibliográficas

[1] C. Archetti, M.G. Speranza, and A. Hertz. A tabu search algorithm for the split delivery vehicle routing problem. *Transportation Science*, 40:64 – 73, 2 2006.

[2] Maria Battarra, Jean-François Cordeau, and Manuel Iori. Pickup-and-delivery problems for goods transportation. In *Vehicle Routing*, 2014.

[3] G. Clarke and J. W. Wright. Scheduling of Vehicles from a Central Depot to a Number of Delivery Points. *Operations Research*, 12(4):568–581, August 1964.

[4] G. B. Dantzig and J. H. Ramser. The truck dispatching problem. *Management Science*, 6(1):80–91, 1959.

[5] Moshe Dror and Pierre Trudeau. Savings by Split Delivery Routing. *Transportation Science*, 23(2):141–145, May 1989.

[6] Gurobi Optimization, LLC. Gurobi Optimizer Reference Manual, 2022.

[7] C. Miller, A.W. Tucker, and R.A. Zemlin. Integer programming formulation of traveling salesman problems. *J. ACM*, 7:326–329, 10 1960.

[8] Sophie Parragh, Karl Doerner, and Richard Hartl. A survey on pickup and delivery problems: Part i: Transportation between customers and depot. *Journal für Betriebswirtschaft*, 58:21–51, 04 2008.

[9] Paolo Toth and Daniele Vigo. *Vehicle routing: problems, methods, and applications*. SIAM, 2014.

[10] Thibaut Vidal, Teodor Gabriel Crainic, Michel Gendreau, and Christian Prins. Heuristics for multi-attribute vehicle routing problems: A survey and synthesis. *CIRRELT*, 02 2012.

[11] Jin Y Yen. Finding the k shortest loopless paths in a network. *management Science*, 17(11):712–716, 1971.