ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE CIENCIAS

PROBLEMA DE RUTEO DE VEHÍCULOS CON UN SOLO PRODUCTO, MÚLTIPLES FUENTES Y DESTINOS, Y CAPACIDAD DE CARGA LIMITADA.

TRABAJO DE INTEGRACIÓN CURRICULAR PRESENTADO COMO REQUISITO PARA LA OBTENCIÓN DEL TÍTULO DE INGENIERO MATEMÁTICO

HENRY FERNANDO ECHEVERRÍA GONZÁLEZ

henry.echeverria01@epn.edu.ec

DIRECTOR: DIEGO FERNANDO RECALDE CALAHORRANO

diego.recalde@epn.edu.ec

DMQ, AGOSTO 2022

CERTIFICACIONES

Yo, HENRY FERNANDO ECHEVERRÍA GONZÁLEZ, 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.

Henry Fernando Echeverría González

Certifico que el presente trabajo de integración curricular fue desarrollado por Henry Fernando Echeverría González, 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.

Henry Fernando Echeverría González

Diego Fernando Recalde Calahorrano

AGRADECIMIENTO

Le doy gracias a mis padres, Henry y María, por su gran amor, paciencia y cariño durante toda mi vida estudiantil. A mis abuelos maternos, Kenton y Mariana, que ha pesar de la distancia siempre estuvieron presentes y pendientes de mi, y de mis estudios. A mis tíos maternos, Gento, Cristina y Verónica, por tantos momentos de alegría y felicidad.

Le doy gracias al Dr. Diego Recalde por su confianza, ayuda y paciencia al otorgarme este trabajo de integración curricular. Fue un privilegio haber trabajado y aprendido con usted.

Le doy gracias a Franco, David, Carlos, Andreo, Danny, Michael, Mauricio y Yomar por sus amistad durante todos estos años, que sin ella no hubiera sido nada igual.

Le doy gracias a todos mis amigos y compañeros por tantos momentos especiales y por haber formado parte de mi vida estudiantil.

Finalmente, me doy gracias a mí por no haber tirado nunca la toalla en los momentos más difíciles de la carrera.

DEDICATORIA

A mis padres y sobre todo a mí.

RESUMEN

En el presente trabajo se propone formular y resolver un problema de enrutamiento de vehículos usando programación lineal entera. Primero, se introduce de manera general el problema de enrutamiento de vehículos y algunas heurísticas. Posteriormente, se da a conocer el problema que se desea modelar, implementar y resolver. Después, se presenta un modelo de programación lineal entera del problema de manera formal. Luego, con base en la literatura se presenta otro modelo de programación lineal entera pero más compacto. Más adelante, se realiza la verificación de los modelos con una instancia artificial y se verifica las soluciones. Por otro lado, se realiza pruebas computacionales en 52 instancias en donde se registra la función objetivo, el GAP de dualidad y el tiempo de cómputo con un límite de 3600 segundos. Seguidamente, se presenta una heurística a dos fases y se resuelve las 52 instancias registrando la función objetivo y el tiempo de cómputo. Finalmente, se usa las soluciones heurísticas como soluciones factibles de inicio para el método exacto (Modelos de programación lineal entera) a fin de obtener mejores resultados.

Palabras clave: VRP, Split, Open, Pickup and Delivery, Capacited, Heterogeneous Fleet y MultiDeport.

ABSTRACT

In this study we propose to formulate and solve a variant of the Vehicle Routing Problem (VRP) using Integer Lineal Programming. First, we begin with a literature review of VRP and some heuristics. After that, we present the main problem in order to model, implement and solve it. Afterwards, two Integer Lineal Programming models are presented, the second one being more compact and concise than the first one. Subsequently, in order to check the validity of the models, they are tested over a toy-like instance. Then, we carry out numerical experiments with 52 instances of the problem, where we retain the function objective value and duality GAP with a limit time of 3600 seconds. Next, a two-phase heuristic is developed, and with it, the 52 instances are performed. We retain the objective value and duality GAP. Finally, we report the results of initializing the second model with the solutions obtained by the heuristic in order to get better results.

Keywords: VRP, Split, Open, Pickup and Delivery, Capacited, Heterogeneous Fleet and MultiDeport.

Índice general

Índice de figuras

Capítulo 1

Introducción

1.1. Objetivo general

Abordar un problema de Ruteo de Vehículos considerando un producto, varias fuentes para la salida de los vehículos, vehículos con una capacidad máxima, puntos de abastecimiento con oferta de producto y puntos de entrega con demanda.

1.2. Objetivos específicos

1. Modelar el problema planteado con Programación Lineal Entera.
2. Implementar computacionalmente el modelo del problema de Ruteo de Vehículos planteado con la ayuda del lenguaje de programación Python.
3. Generar instancias para realizar experimentación computacional y resolverlas con el solver Gurobi.
4. Estudiar e implementar algoritmos heurísticos para el problema de Ruteo de Vehículos planteado.
5. Comparar mediante diferentes indicadores la eficiencia de los algoritmos heurísticos con el método exacto.

1.3. Alcance

El alcance del componente se basa en la siguiente metodología:

- a) Formulación del problema de Ruteo de Vehículos. Posteriormente hacer una revisión de la literatura acerca del problema planteado para conocer y entender nuevas ideas para abordarlo.
- b) Una vez hecha la revisión bibliográfica, se adaptará las ideas desarrolladas en el problema para formularlo como un problema de Programación Lineal Entera.
- c) Para poder resolver el problema se requiere de un *Framework* y un Solver, para ello es necesario hacer un breve repaso del lenguaje de programación Python y el solver Gurobi que se usaran como *Framework* y Solver, respectivamente.
- d) Una vez hecho el repaso, se procede a la implementación del modelo de Programación Lineal Entera.
- e) Validar el modelo de programación lineal entera mediante el uso de instancias factibles pequeñas que sean resueltas en un tiempo computacional bajo o moderadamente bajo (menos de un minuto).
- f) Se procede a la creación de diferentes instancias más complejas para hacer pruebas computacionales y evaluar si es posible obtener soluciones exacta y va hacer reportado el GAP alcanzado fijando un tiempo.
- g) Para aplicar una técnica alternativa de resolución se plantean algoritmos heurísticos. Así, se revisará bibliografía sobre heurísticas relacionadas al problema para posteriormente hacer pruebas computacionales con las mismas instancias generadas e incluso más grandes.
- h) Comparar los resultados mediante el valor de la función objetivo y el tiempo de resolución.

1.4. Estado del arte

El problema de enrutamiento vehicular o como se conoce en la literatura *VRP* (*Vehicle Routing Problem*) se lo describe como el recorrido de uno o varios vehículos por una red o un (di)grafo.

Por otro lado, Sitek y Wikarek en [13] al *VRP* lo definen como un problema que responde a la pregunta "*¿Cuál es el conjunto de rutas óptimas para un grupo de vehículos/flotas para viajar con el fin de llegar a un conjunto de clientes?*" y también enumeran los objetivos más comunes que se maneja en el problema *VRP* como:

1. minimizar los costos de transporte, así como la distancia recorrida
2. minimizar los número de vehículos necesarios para cubrir todos los puntos de entrega.
3. si los tiempos estimados de viaje y la carga del vehículo ha sido excedido, se busca minimizarlos.

Vogiatzis y Pardalos en [15] se describe el problema de manera simple pero es complicado en resolverlo pues es una generalización del *TSP* (*Traveling Salesman Problem*) y como consecuencia de ello el problema pertenece a la clase *NP-Hard*, como menciona Toth y Vigo en [14].

Desde la presentación del *VRP*, el cual fue introducido por Dantzig y Ramser en [2] en el año 1959, hasta el día de hoy, han surgido diferentes variantes del mismo. Marinakis et al. en [12] realizan un resumen de las variantes del *VRP* más comunes en la literatura. Algunas de ellas las enumeramos a continuación:

1. Capacitated Vehicle Routing Problem (CVRP)
2. Variantes básicas del Problema de Enrutamiento de Vehículos
    a) Open Vehicle Routing Problem
    b) Vehicle Routing Problem with Time Windows
    c) Multi-Depot Vehicle Routing Problem
3. Pickup and Delivery

   a) One Commodity Pickup and Delivery Problem (1-PDVRP)
   b) Vehicle Routing Problem with Simultaneous Pickup and Delivery (VRP-SPD)
   c) Vehicle Routing Problem with Backhauls

4. Otras variantes del Problema de Enrutamiento de Vehículos
   a) Split Delivery Vehicle Routing Problem
   b) Heterogeneous Fleet Vehicle Routing Problem
   c) Green Vehicle Routing Problem
   d) School Bus Routing and Scheduling Problem
   e) Vehicle Routing Problem with Profits
5. Stochastic and Dynamic Vehicle Routing Problems
6. Arc Routing Problems

El objetivo de enumerar las variantes del *VRP*, es poder clasificar nuestro problema. Para ello presentamos una descripción del problema.

1.4.1. Descripción del problema

En el presente trabajo de integración curricular se aborda el siguiente problema de enrutamiento vehicular: Se desea recoger un producto desde de varios puntos de abastecimiento. Para ello se dispone de varios depósitos y cada uno de estos posee una cantidad fija de vehículos con capacidad limitada para cargar el producto. Cada punto de abastecimiento posee una oferta del producto. Después de recoger el producto, los vehículos deben entregar los mismos a diferentes lugares de entrega; cada punto de entrega posee una demanda que se debe satisfacer de manera obligatoria. Para cumplir con lo expuesto, se necesita saber cuántos vehículos deben usarse y qué ruta deben seguir para recoger los productos para cumplir con la demanda mientras se minimiza el costo operativo total por el uso de los vehículos y el costo de la ruta que debe seguir cada uno de ellos.

Ahora, enumeramos las condiciones que debe cumplir el problema descrito:

1. No existe un camino desde los depósitos hasta los lugares de entrega, ni caminos entre de los depósitos, ni entre los lugares de entrega.
2. Consideramos que existe una distancia desde los depósitos hasta los nodos destino y también entre ellos mismo.
3. Varios vehículos tienen permitido visitar un mismo nodo de abastecimiento.
4. Un vehículo finalizará su ruta en un único lugares de entrega.

Clasificación del problema

Una vez descrito el problema, se procede a clasificarlo. Todas las variantes del *VRP* que se enumeran a continuación se encuentran descritas de manera más detallada por en Marinakis et al. en [12].

1. **Capacitated VRP**: Los vehículos tienen una capacidad limitada.
2. **Heterogeneous Fleet VRP**: La capacidades de los vehículos son diferentes al igual que el costo de uso de los mismos.
3. **Pickup and Delivery VRP**: El objetivo general del problema es recoger producto y entregarlos en varios lugares.
4. **Multi-Deport VRP**: Existen varios depósitos donde se encuentran los vehículos.
5. **Split Pickup and Delivery VRP** : Se permite visitar un lugar de abastecimiento para recoger producto con diferentes vehículos. Posteriormente varios vehículos pueden entregar a un mismo lugar de destino. Es decir, los proveedores están dispuestos a cargar su producto en varios vehículos y los clientes están dispuestos a recibir esa carga en múltiples entregas.
6. **Open VRP**: Los vehículos no vuelven al mismo lugar de partida.

Revisión breve de un problema similar

Hasani Goodarzi y Tavakkoli Moghaddam en [8] consideran un split vehicle routing problem, con restricción de capacidad para cross-docks de múltiples productos. El problema se formula como un modelo de programación lineal entera mixta (MILP) para determinar las mejores rutas de vehículos y el número óptimo de vehículos utilizados.

De una lectura del artículo mencionado podemos hacer símil con nuestro problema y enumerar las semejanzas. Se tiene que: 1) El objetivo es minimizar los costos de las rutas y del costo de los vehículos; 2) La característica de Split Pickup and Delivery; 3) Varios vehículos y 4) Demandas y ofertas en lugares de destino y lugares de abastecimiento, respectivamente.

1.4.2. Heurísticas

Una **heurística** se define como un método de solución (o un proceso) para encontrar una solución aceptable de un problema. En general, las **heurísticas** suelen aplicarse a problemas difíciles. Estos problemas se caracterizan principalmente por lo complicado que es obtener una solución óptima (incluso una solución factible).

Existen varias razones para considerar usar **heurísticas** en un problema y a continuación enumeramos algunas descritas por Laguna y Maren en [9].

1. Los métodos de solución exacta conocidos son computacionalmente costosos y, por lo tanto, solo pueden resolver pequeñas instancias del problema.
2. Se utiliza un método heurístico para guiar al método exacto con el objetivo de encontrar mejores soluciones.
3. Un método heurístico también es usado para problemas donde no exista un método para encontrar una solución exacta.

Aunque los métodos heurísticos están construidos para problemas específicos se puede clasificar en cinco generales categorías.

1. ***Decomposition***: Se descompone el problema es subproblemas, esto haciendo honor a "Divide y Vencerás".
2. ***Induction***: Las estrategias que se aplica en instancias pequeñas del problema original se las aplica en las instancias grandes.
3. ***Reduction***: El objetivo es reducir el espacio de soluciones y simplificar el problema original.
4. ***Construction***: Se obtiene una solución siguiendo una secuencia de pasos.
5. ***Local Search***: Trabaja en una solución existente y trata de mejorar.

Para mayor entendimiento e información sobre **Heurística** visitar [9].

Heurísticas en el *VRP*

Una solución óptima puede ser obtenida en instancias pequeñas del *VRP* en un tiempo razonable con los métodos exactos. Pero es bien estudiado que en instancias grandes se consume mucho tiempo para llegar a la optimalidad. Por ello se propone usar diferentes técnicas de resolución como las heurísticas.

Toth y Vigo en [14] describen que las heurísticas del *VRP* son tan viejas como el mismo *VRP* y actualmente su campo de aplicación es tan grande y rico que sería muy difícil recopilar todas. Además, Laporte en [10] enfatiza que los métodos más comunes para la resolución de un problema *VRP* son los algoritmos heurísticos.

Para el trabajo se analiza solamente un método heurístico: la heurística de dos fases (*two-phase heuristics*). Este método está relacionado con las heurísticas de descomposición. Laporte y Semet en [11] la definen como una descomposición del problema en dos componentes. Dentro de las heurísticas de dos fases se encuentra el método *cluster-first, route-second* en donde en general la primera fase es agrupar y la segunda es enrutar. Los algoritmos más comunes dentro de este método son *Fisher and Kaikumar*, *The Petal Algorithm*, *The Sweep algorithm* y *Taillard*.

Otras aplicaciones de heurísticas en dos fases con el método *cluster-first, route-second* para el *VRP* son:

1. Garside y Laili en [6] discuten el problema de enrutaminto pero con diferentes viajes (*Multi-Trip Periodic Vehicles*). Para resolverlo se usa una heurística *cluster-first, route-second method*. En la fase 1 se divide en grupos a los clientes usando un modelo de programación lineal entera y en la fase 2 se determina la ruta de los grupos y el tiempo total de viaje no exceda las horas de trabajo del vehículo.

2. Cömert et al. en [4] realizan un caso de estudio de un *Capacited VRP* y lo resuelven con una heurística *cluster-first, route-second method*. Primero se agrupa los clientes con tres diferentes métodos *K-means*, *K-medoids* y *random clustering* considerando la capacidad de vehículos. Segundo, los problemas de enrutamiento para cada grupos de clientes se resuelve con *Branch and Bound*. El caso de estudio se emplea en una cadena de supermercados.

Capítulo 2

Metodología

La metodología que se usa en el presente trabajo es la formulación de un de un problema de optimización como un ***Programa Lineal Entero***. El método clásico de resolución de estos problemas es el método *Branch and Bound*. La herramienta utilizada para aplicar el *Branch and Bound* es un solvers de Optimización. En particular, en el presente trabajo se utiliza el solver *Gurobi* [7].

2.1. Definiciones importantes

Para poder formular un programa lineal entero es necesario definir un **problema de optimización**. De forma general un **problema de optimización** tiene la siguiente forma:

$$\begin{aligned} &\text{Mín} \quad f_0(x) \\ &\text{s.a.} \quad f_i(x) \leq b_i, \quad \forall i = 1, \ldots, m. \end{aligned} \tag{1}$$

Donde $x = (x_1, ..., x_n) \in \mathbb{R}^n$ es la variable a optimizar; la función $f_0 : \mathbb{R}^n \longrightarrow \mathbb{R}$ es la función objetivo y $f_i : \mathbb{R}^n \longrightarrow \mathbb{R}$, $i = 1, \ldots, m$, son las restricciones; las constantes $b_i \in \mathbb{R}$ son las cotas superiores de las restricciones. El Problema 1 se dice un **Programa Lineal Entero** si y solamente si la

función objetivo f_0 y las restricciones f_i son lineales y que las variables a optimizar $x = (x_1, ..., x_n) \in \mathbb{Z}^n$.

Es necesario definir algunos conjuntos y estructuras para la presentación formal de una instancia del problema.

Primero, un grafo dirigido o **digrafo** es una tripleta $G = (V, E, \phi)$ donde V y E son conjuntos finitos y ϕ es una función con dominio E y codominio $V \times V$. Llamamos E el conjunto de aristas del **digrafo** G y a V al conjunto de vértices de G. Además, decimos que el arco (x, y) tiene costo c_{xy} si existe una función c de $\phi(E)$ hasta $\mathbb{R}^+$.

El digrafo G se dice **Digrafo Bipartito** cuando V se puede particionar en dos subconjuntos $X, Y \subset V$ de tal forma de que no existan arcos entre vértices de X y que no existan arcos entre vértices de Y.

Por otro lado, el movimiento de un vértice a otro en el digrafo G se lo define como un **camino**, sea $e_1, e_2, \ldots, e_{n-1}$ una sucesión de elementos de E para la cual existe una sucesión $a_1, a_2, \ldots, a_n$ elementos de V tales que $\phi(e_i) = (a_i, a_{i+1})$. La sucesión $e_1, e_2, \ldots, e_{n-1}$ es llamada un **camino** en G.

Finalmente, se define los cortes en un grafo. El **Corte de Entrada de** X es el conjunto de arcos (y, x) tal que $y \notin X$ y $x \in X$ y lo denotamos $\delta^-(X)$ y el **Corte de Salida de** X es el conjunto de arcos (x, y) tal que $x \in X$ y $y \notin X$ y lo denotamos $\delta^+(X)$.

2.2. Modelización Matemática

En esta sección se definirá de manera formal el problema y sus parámetros. Posteriormente se formulan dos modelo de programación lineal, un ejemplo pequeño donde se muestra las soluciones dadas por los modelos, una heurística, y finalmente algunas observaciones importantes.

2.2.1. Instancia del problema

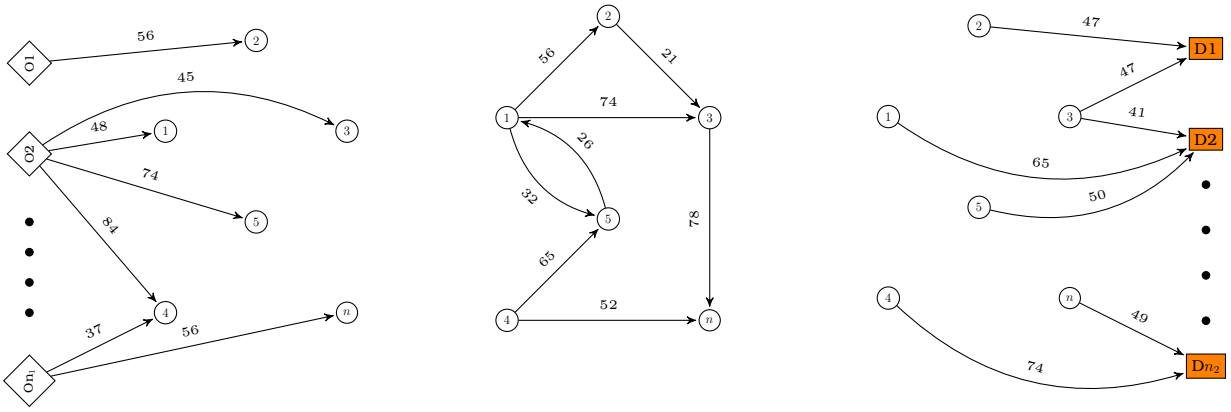
Empecemos por definir de manera formal el digrafo donde se trabaja; el digrafo será la unión de 3 digrafos diferentes: dos bipartitos y otro digrafo.

Digrafo de abastecimiento es un digrafo $G_1 = (V, A, \phi_1)$ donde V son vértices y los llamaremos nodos de abastecimiento con $|V| = n \in \mathbb{Z}^+$ y A los arcos entre los nodos de abastecimiento, no necesariamente todos los nodos deben estar conectados, y c_1 una función de costo para A.

Digrafo con fuentes es un digrafo $G_2 = (\mathcal{O} \cup V, \mathcal{A}_\mathcal{O}, \phi_2)$ donde V son los nodos de abastecimiento, y $\mathcal{O}$ conjunto de vértices con $|\mathcal{O}| = n_1 \in \mathbb{Z}^+$. Denominados nodos fuente; $\mathcal{A}_\mathcal{O}$ es el conjunto de arcos cuyos elementos tiene la forma (x, y) con $x \in \mathcal{O}$ y $y \in V$. Además, sea c_2 una función de costo para $\mathcal{A}_\mathcal{O}$. Notemos que el digrafo es bipartito pues cumple con la definición establecida anteriormente.

Digrafo con destinos es un digrafo $G_3 = (\mathcal{D} \cup V, \mathcal{A}_\mathcal{D}, \phi_3)$ donde V son los nodos de abastecimiento y $\mathcal{D}$ conjunto de vértices con $|\mathcal{D}| = n_2 \in \mathbb{Z}^+$, denominados nodos destino; $\mathcal{A}_\mathcal{D}$ es el conjunto de arcos cuyos elementos tiene la forma (x, y) con $x \in V$ y $y \in V$. Adicionalmente, sea c_3 una función de costo para $\mathcal{A}_\mathcal{D}$; este digrafo también es bipartito.

Entonces el digrafo en donde se trabaja se define como la unión de estos tres digrafos descritos. Es decir, $G = (V \cup \mathcal{O} \cup \mathcal{D}, A \cup \mathcal{A}_\mathcal{O} \cup \mathcal{A}_\mathcal{D}, \phi)$. El costo de ir de un vértice i a un vértice j en el digrafo G es $T(i, j) = t_{ij}$. En la Figura 2.1 se muestran los digrafos por separado, mientras que en la Figura 2.2 se muestra la unión de los digrafos.



(a) G_2: Fuentes (b) G_1:Abastecimiento (c) G_3:Destinos

Figura 2.1: Digrafos

Ahora describiremos los parámetros que se tiene para el problema como la oferta, los vehículos y sus capacidades y la demanda de los destinos.

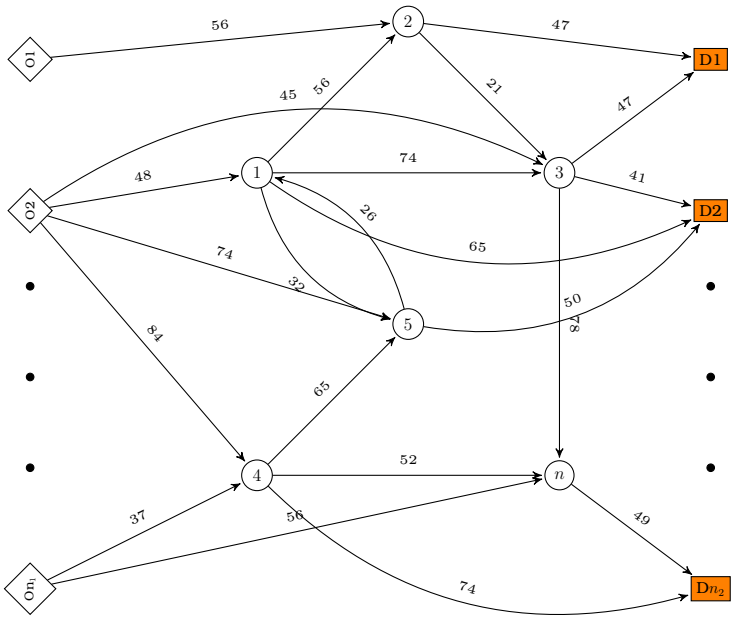


Figura 2.2: Digrafo Final G

Vehículos en las fuentes: Para cada fuente $k \in \mathcal{O}$, existe un conjunto $\mathcal{M}^k$ de vehículos; cada vehículo $m \in \mathcal{M}^k$ cuenta con sus capacidad y su costo operativo que los notaremos como $q^{km} \in \mathbb{Z}^+$ y $C^{km} \in \mathbb{R}^+$ respectivamente.

Oferta en los abastecimientos: Para cada lugar de abastecimiento $i \in V$, existe una oferta de producto notada por $o_i \in \mathbb{Z}^+$; así los vehículos visitan los nodos de abastecimiento y recogen el producto.

Demanda en los destinos: Cada destino $d \in \mathcal{D}$ posee una demanda de producto notada por $d_p \in \mathbb{Z}^+$; los vehículos llegarán a un solo destino para entregar el producto y cumplir con las demandas.

Finalmente, en la Figura 2.3 se muestra más detalladamente como se ve una instancia del problema a resolver.

2.2.2. Condiciones del problema

Describamos de manera formal las condiciones que se impusieron en el problema.

1. No existen arcos desde los nodos fuente hasta los nodos destino y viceversa. Tampoco existen arcos entre los nodos fuente, ni entre

de los nodos de destino. Es decir, que los conjuntos: $C_i = \{c_{fd} : f \in A_{\mathcal{O}}, d \in A_{\mathcal{D}}\}$, $C_v = \{c_{df} : f \in A_{\mathcal{O}}, d \in A_{\mathcal{D}}\}$, $C_o = \{c_{o\hat{o}} : o \in A_{\mathcal{O}}, \hat{o} \in A_{\mathcal{O}}\}$, $C_d = \{c_{d\hat{d}} : d \in A_{\mathcal{D}}, \hat{d} \in A_{\mathcal{D}}\}$, no se considerán en la modelización.

2. Existe una distancia desde los nodos fuente hasta los nodos destino y también entre ellos mismo. Es decir que, para todo $i, j \in A_{\mathcal{O}} \cup A_{\mathcal{D}}$ con $i \neq j$ existe una distancia $d_{i,j} > 0$

3. Varios vehículos tienen permitido visitar un mismo nodo de abastecimiento. Supongamos P_1 y P_2 dos caminos de dos vehículos diferentes. Entonces existe $a \in V$ tal que $a \in P_1$ y $a \in P_2$. Es decir, $P_1 \cap P_2 \neq \emptyset$.

4. Un vehículo finalizará su ruta en un único nodo de demanda. Es decir, si P_1 es el camino de un vehículo que pertenece a la fuente O_n y tiene $O_n, e_1, \ldots e_{m-1}, e_m$ como secuencia de nodos de abastecimiento, entonces $e_m \in \mathcal{D}$ y para todo $i \in \{1, 2, \ldots, m-1\}$ se cumple que $e_i \notin \mathcal{O} \cup \mathcal{D}$.

Factibilidad de una instancia del problema

Con lo descrito en la Sección 1.4.1 y de manera matemática en la Sección 2.2.2 se puede enumerar ciertas condiciones necesarias para que una instancia del problema sea factible.

1. **Con respecto a la oferta:** Es necesario que la oferta del producto sea mayor o igual a la demanda. Es decir, $\sum_{i \in V} o_i \geq \sum_{j \in \mathcal{D}} d_j$, donde o_i es la oferta en el nodo de abastecimiento i y d_j la demanda en el nodo destino j.

2. **Con respecto a los vehículos:** La cantidad de vehículos debe ser al menos igual a la cantidad de nodos destinos. Esto último porque un vehículo no puede satisfacer varios nodos de destinos. Así, sea M_k la cantidad de vehículos en la fuente k, entonces $\sum_{k \in \mathcal{O}} M_k \geq n_2$.

2.2.3. Modelos de programación lineal

Antes de describir las variables de decisión es importante definir el conjunto $\mathcal{K} = \mathcal{K}_1 \cup \mathcal{K}_2$, donde $\mathcal{K}_1 = \{(k, m, i, j) : k \in [n_1],\ m \in \mathcal{M}^k,\ (i, j) \in$

$\mathcal{A}_{\mathcal{O}}, i = O_k\}$ y $\mathcal{K}_2 = \{(k, m, i, j) : k \in [n_1],\ m \in \mathcal{M}^k,\ (i, j) \in \mathcal{A} \cup \mathcal{A}_{\mathcal{D}}\}$ y también notar $\{1, 2, \cdots, n\}$ como $[n]$.

Las variables de decisión se definen de la siguiente manera:

X_{ij}^{km} es una variable binaria que toma el valor de uno si el vehículo $m \in \mathcal{M}^k$ de la fuente $k \in [n_1]$ va desde el nodo $i \in V$ hasta al nodo $j \in V$ con $(k, m, i, j) \in \mathcal{K}$ y vale cero en caso contrario.

y_i^{km} es una variable binaria que toma el valor de uno si el vehículo $m \in \mathcal{M}^k$, $k \in [n_1]$, visita el nodo $i \in V \cup \mathcal{D}$ y vale cero de lo contrario.

z^{km} es una variable binaria que toma el valor de uno si el vehículo $m \in \mathcal{M}^k$, $k \in [n_1]$ se usa y vale cero caso contrario.

$a_{ip}^{km} \in \mathbb{Z}^+$ es la cantidad recogida del producto en el nodo $i \in V$ por el vehículo $m \in \mathcal{M}^k$, $k \in [n_2]$, y que se entrega en el nodo de destino D_p con $p \in [n_2]$.

$u_i \in [n]$: Variable de ordenamiento con $i \in V$ para evitar soluciones que incluyan ciclos.

Modelo 1

A continuación se presenta el modelo de programación lineal entera.

$$\text{mín} \quad \sum_{(k,m,i,j)\in\mathcal{K}} t_{ij} X_{ij}^{km} + \sum_{k\in[n_1]} \sum_{m\in\mathcal{M}_k} C^{km} z^{km} \tag{1}$$

$$\text{s.a} \quad \sum_{j\in\delta^-(O_k)} X_{O_k j}^{km} = z^{km}, \qquad \forall k \in [n_1], m \in \mathcal{M}^k, \tag{2}$$

$$\sum_{p\in[n_2]} \sum_{j\in\delta^+(D_p)} X_{jD_p}^{km} = z^{km}, \qquad \forall k \in [n_1], m \in \mathcal{M}^k, \tag{3}$$

$$\sum_{j\in\delta^-(i)} X_{ji}^{km} = y_i^{km}, \qquad \forall k \in [n_1], m \in \mathcal{M}^k, i \in V, \tag{4}$$

$$\sum_{j\in\delta^-(D_p)} X_{jD_p}^{km} = y_{D_p}^{km}, \qquad \forall k \in [n_1], m \in \mathcal{M}^k, p \in [n_2], \tag{5}$$

$$\sum_{i\in\delta^-(s)} X_{is}^{km} = \sum_{j\in\delta^+(s)} X_{sj}^{km}, \qquad \forall k \in [n_1], m \in \mathcal{M}^k, s \in V, \tag{6}$$

$$u_j \geq u_i + (1+n)x_{ij} - n, \qquad \forall i,j \in V, k \in [n_1], m \in \mathcal{M}^k, \tag{7}$$

$$\sum_{i \in V} a_{ip}^{km} \leq d_p y_{D_p}^{km}, \qquad \forall k \in [n_1], k \in \mathcal{M}^k, p \in [n_2], \tag{8}$$

$$\sum_{p \in [n_2]} \sum_{k \in [n_1]} \sum_{m \in \mathcal{M}^k} a_{ip}^{km} \leq o_i, \qquad \forall i \in V, \tag{9}$$

$$\sum_{p \in [n_2]} \sum_{i \in V} a_{ip}^{km} \leq q^{km}, \qquad \forall k \in [n_1], m \in \mathcal{M}^k, \tag{10}$$

$$a_{ip}^{km} \leq o_i y_i^{km}, \qquad \forall k \in [n_1], m \in \mathcal{M}^k, i \in V, p \in [n_2], \tag{11}$$

$$\sum_{k \in [n_1]} \sum_{m \in \mathcal{M}^k} \sum_{i \in V} a_{id}^{km} = d_p, \qquad \forall p \in [n_2], \tag{12}$$

$$X_{ij}^{km} \in \{0,1\}, \qquad \forall (k,m,i,j) \in \mathcal{K}, \tag{13}$$

$$y_i^{km} \in \{0,1\}, \qquad \forall i \in V, k \in [n_1], m \in \mathcal{M}^k, \tag{14}$$

$$z^{km} \in \{0,1\}, \qquad \forall k \in [n_1], m \in \mathcal{M}^k, \tag{15}$$

$$a_{ip}^{km} \in \mathbb{Z}^+, \qquad \forall i \in V, k \in [n_1], m \in \mathcal{M}^k, p \in [n_2], \tag{16}$$

$$u_i \in [n], \qquad \forall i \in V. \tag{17}$$

(1) es la Función Objetivo minimiza el costo de las rutas y los costos operativos de los vehículos que se usan. La Restricción (2) expresa que si el vehículo m es usado entonces debe obligatoriamente salir de O_k y salir a un nodo de abastecimiento j en el corte de O_k. La Restricción (3) parecida a la anterior, si el vehículo m es usado entonces debe obligatoriamente llegar a algún nodo de destino. La Restricción (4) relaciona X_{ij}^{km} con la variable y_i^{km} e indica que si un nodo de abastecimiento fue visitado entonces debió llegar de algún otro nodo de abastecimiento. La Restricción (5) es similar a Restricción (4) pero para el caso de los nodos de destino. La Restricción (6) es la de flujo: si un vehículo entra a un nodo, entonces debe salir del mismo, excepto para los nodos fuente y nodos de destino. La Restricción (7) evita la aparición de sub-tours. La Restricción (8) es una restricción de activación y expresa que si un vehículo va a entregar (o visitar) al nodo de destino D_p entonces solo va a recoger productos para ese nodo de destino. La Restricción (9) controla la oferta de cada nodo de abastecimiento. La Restricción (10) controla la capacidad de cada vehículo. La Restricción (11) es de activación, nos controla la variable a, solo es distinta de 0 para nodos de abastecimiento que los vehículos hayan visitado. La Restricción (12) permite que se satisfaga la demanda. Final-

mente, las Restricciones (13), (14), (15), (16) y (17) definen los espacios de las variables de decisión.

Modelo Compacto

Hasani Goodarzi y Tavakkoli Moghaddam en [8] no consideran la variable z^{km}. Así, se puede compactar el **Modelo 1** donde la función objetivo tiene la forma:

$$\sum_{k=1}^{n_1}\sum_{m=1}^{M_k} C^{km} z^{km} \Longrightarrow \sum_{(k,m,.,j)\in\mathcal{K}} C^{km} X_{O_k j}^{km}$$

Las siguientes restricciones también cambiarán a:

$$\sum_{j\in\delta^-(O_k)} X_{O_k j}^{km} = z^{km} \Longrightarrow \sum_{j\in\delta^-(O_k)} X_{O_k j}^{km} \leq 1, \qquad \forall k \in [n_1], m \in \mathcal{M}^k,$$

$$\sum_{p\in[n_2]}\sum_{j\in\delta^+(D_p)} X_{jD_p}^{km} = z^{km} \Longrightarrow \sum_{p\in[n_2]}\sum_{j\in\delta^+(D_p)} X_{jD_p}^{km} \leq 1, \qquad \forall k \in [n_1], m \in \mathcal{M}^k.$$

Las demás restricciones se mantienen. Llamemos a este modelo **Modelo 2**.

2.3. Verificación de la validez de los modelos

Para mostrar la validez de los modelos se propone el siguiente ejemplo:

Con un costo de 3405 unidades monetarias, la solución del Ejemplo 1 se presenta en la Figura 2.4.

Además, el tiempo de cómputo de la solución óptima fue de 0.1818s y 0.8164s para el Modelo 1 y Modelo 2, respectivamente. En la Figura 2.5 se presenta la recolección de los productos después de visitar un nodo de abastecimiento.

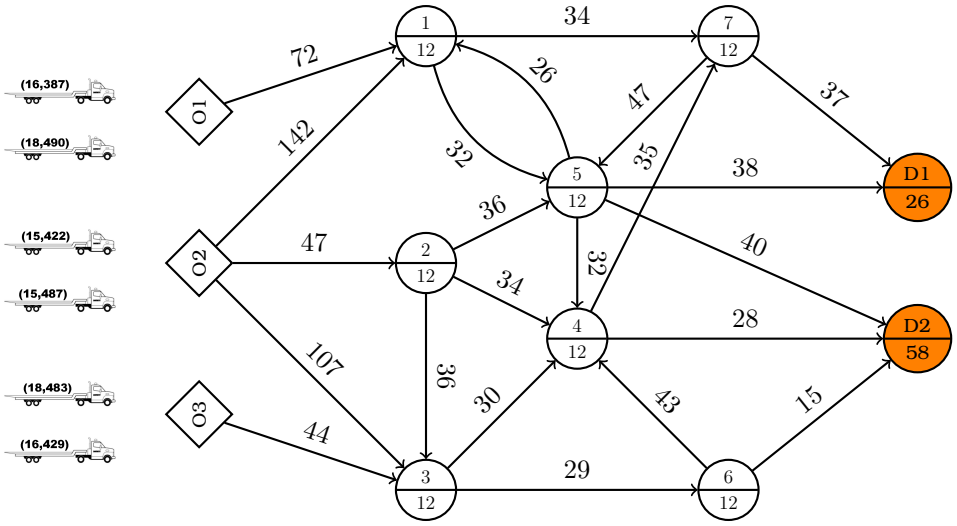
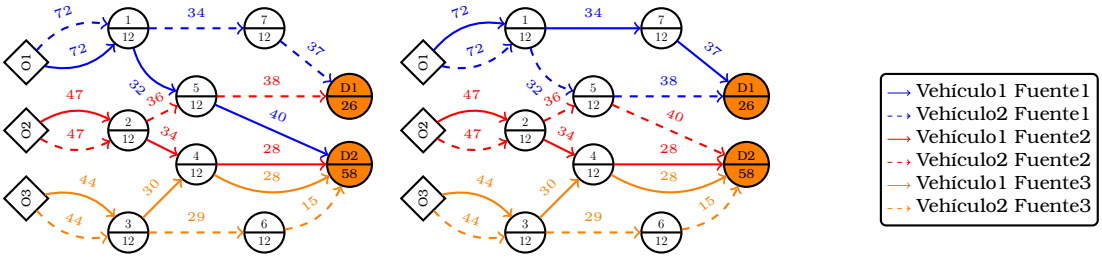


Figura 2.3: Ejemplo 1



(a) Solución Modelo 1

(b) Solución Modelo 2

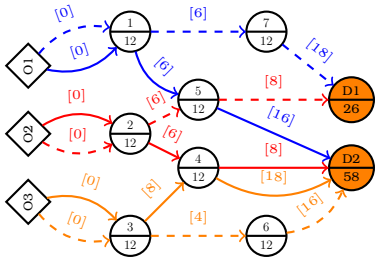
Figura 2.4: Caminos óptimos

2.4. Heurística

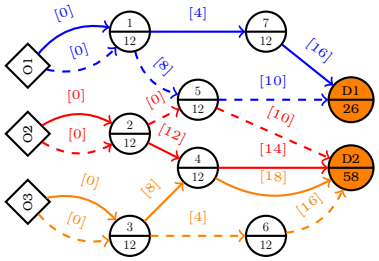
La heurística propuesta en este trabajo de integración curricular se basa en el método *cluster-first, route-second*. Para la primera fase, se escoge los vehículos. Y la segunda fase es enrutar los vehículos seleccionados en la primera fase.

2.4.1. Primera Fase

Consideremos la instancia presentada en la Sección 2.3 y para enfocarnos más en la primera fase, la representaremos gráficamente como se puede ver en la Figura 2.6a y Figura 2.6b. El objetivo, en la primera fase, es escoger arcos de forma que el costo de escoger los mismos sea



(a) Recolección con el Modelo 1



(b) Recolección con el Modelo 2

Figura 2.5: Recolección del Producto

mínimo. Notemos que escoger un arco en el Digrafo 2.6b es equivalente a escoger un auto que va a llevar producto a un destino; la cantidad de producto que vaya a llevar también es determinada por el arco escogido pues cada vehículo tiene su capacidad. Por ejemplo, si se escoge el arco $(O1-1, D2)$esto implica que el vehículo 1 de la fuente 1 se dirige al destino 2; además, notemos que la capacidad de $O1-1$ es de 16, por lo que el vehículo 1 de la fuente 1 irá al destino 2 entregando a lo más 16 unidades de producto. El problema descrito anteriormente se puede considerar un Problema de Transporte; Appa en [1] hace un breve resumen del problema problema del transporte junto a sus variantes. Para resolver este problema se utiliza Programación Lineal Entera tomando en cuenta las condiciones del problema inicial que fueron descritas en la Sección 2.2.2.

Programa Lineal para la Fase 1

Consideremos C^{km} el costo de usar el vehículo $m \in \mathcal{M}^k$ del depósito $k \in \mathcal{O}$. Definamos el conjunto $L = \{(k, m, p) : k \in \mathcal{O}, m \in \mathcal{M}^k, p \in \mathcal{D}\}$, representa a todas las tripletas ordenadas posibles. Sea

x_p^{km} una variable entera positiva que representa la cantidad de producto que va a transportar el vehículo $m \in \mathcal{M}^k$ del depósito $k \in O$ al destino $p \in \mathcal{D}$

z_p^{km} una variable binaria que toma el valor de 1 si el vehículo $m \in \mathcal{M}^k$ del depósito $k \in O$ transporta al destino $p \in \mathcal{D}$ y 0 en caso contrario.

Modelo:

$$\text{mín} \quad \sum_{(k,m,p)\in L} C^{km} z_p^{km}, \tag{1}$$

$$\text{s.a} \quad \sum_{(k,m,*)\in L} x_p^{km} = d_p, \qquad \forall p \in \mathcal{D}, \tag{2}$$

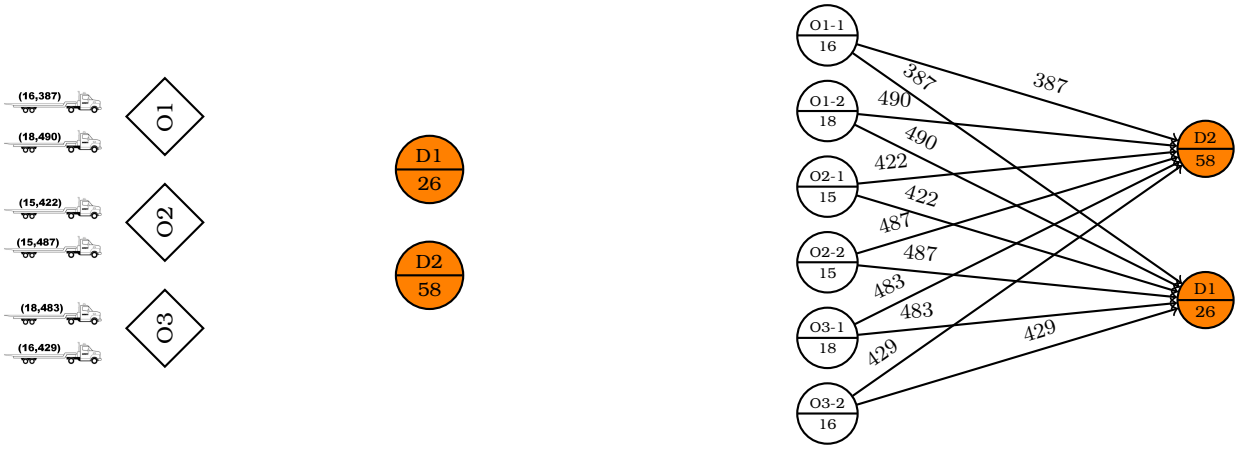
$$\sum_{(*,*,p)\in L} z_p^{km} \leq 1, \qquad \forall k \in O, m \in \mathcal{M}^k, \tag{3}$$

$$x_p^{km} \leq z_p^{km} q^{km}, \qquad \forall (k,m,p) \in L, \tag{4}$$

$$x_p^{km} \in \mathbb{Z}, \qquad \forall (k,m,p) \in L, \tag{5}$$

$$z_p^{km} \in \{0,1\}, \qquad \forall (k,m,p) \in L. \tag{6}$$

La función a minimizar (1) representa el costo operativo de los vehículos; la Restricción (2) permite satisfacer la demanda; como un vehículo puede entregar producto solamente a un lugar es necesaria la Restricción (3), la Restricción (4) es de activación: si se usa un vehículo, entonces este podrá transportar cierta cantidad de producto. Finalmente, Las restricciones (5) y (6) muestran los espacios de las variables de decisión.

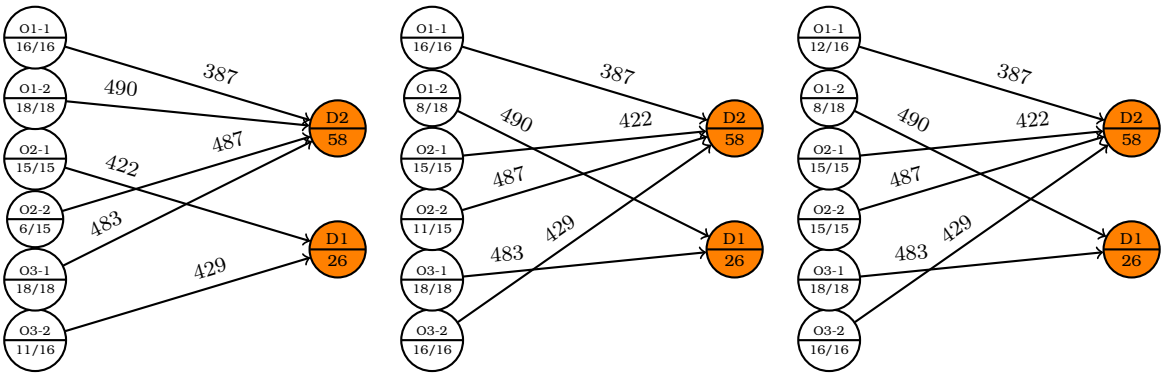


(a) (b)

Figura 2.6: Instancia para la primera fase

La solución encontrada para la instancia mostrada en la Figura 2.6b se muestra en el Figura 2.7a. Sin embargo, no es la única solución pues existen varias soluciones como se presentan en las Figuras 2.7b y 2.7c.

El resultado de la primera fase es un nuevo conjunto y dos nuevos parámetros que se los usarán posteriormente. Al conjuto lo notaremos



(a) (b) (c)

Figura 2.7: Soluciones para 2.6b

como: $A_{\mathcal{K}} = \{(k, m) \in \mathcal{O} \times \mathcal{M}^k : z_p^{km} = 1, p \in \mathcal{D}\}$. Los parámetros los notaremos como: $Au_0^{km} \in \mathcal{D}$ y $Au_1^{km} \geq 0$. El parámetro Au_0^{km} representa el destino del vehículo m que pertenece al depósito k. En cambio, el parámetro Au_1^{km} representa las unidades de producto que transporta el vehículo m que pertenece al depósito k. Los parámetros existirían si y solamente si $z_p^{km} = 1$. Observemos que los parámetros Au_0^{km} y Au_1^{km} están indexados por el conjunto $A_{\mathcal{K}}$. Para el ejemplo presentado en la Figura 2.7a los parámetros son: $Au_0^{11} = D2$, $Au_1^{11} = 16$, $Au_0^{12} = D2$, $Au_1^{12} = 18$, $Au_0^{21} = D1$, $Au_1^{21} = 15$, $Au_0^{22} = D2$, $Au_1^{22} = 6$, $Au_0^{31} = D2$, $Au_1^{31} = 18$, $Au_0^{32} = D1$ y $Au_1^{32} = 11$. Finalmente, también tendremos un costo $\mathcal{C} \geq 0$ que representa el uso de los vehículos, el cual será igual al valor de la función objetivo de la primera fase y para el ejemplo presentado en la Figura 2.7a se tiene que $C = 2698$ unidades monetarias.

2.4.2. Segunda Fase

Una vez elegidos los vehículos, actualizada su capacidad y etiquetado su destino hay que enrutarlos. Para ellos se puede aplicar varios métodos, por ejemplo Figliozzi en [5] propone un método iterativo de dos fases para la solución del *Vehicle routing problem with soft time windows*. En el trabajo se propone un Programa Lineal Entero para el enrutamiento de los vehículos.

Programa Lineal para la Fase 2

Consideremos $t_{ij} \in \mathbb{R}^+$ que representa el costo de ir entre los lugares de abastecimiento i y j. Además, en el apartado anterior se introdujo los parámetros $Au_0^{km} \in \mathcal{D}$ y $Au_1^{km} \geq 0$, y el conjunto $A_\mathcal{K}$. También se considera el costo de los vehículos elegidos notado por $\mathcal{C} > 0$ y el conjunto $\mathcal{K} = \mathcal{K}_1 \cup \mathcal{K}_2 \cup \mathcal{K}_3$, donde $\mathcal{K}_1 = \{(k,m,i,j) : (k,m) \in A_\mathcal{K}, i = \mathcal{O}_k, j \in \delta^-(\mathcal{O}_k)\}$, $\mathcal{K}_2 = \{(k,m,i,j) : (k,m) \in A_\mathcal{K}, (i,j) \in \mathcal{A}\}$ y $\mathcal{K}_3 = \{(k,m,i,j) : (k,m) \in A_\mathcal{K}, j = Au_0^{km}, i \in \delta^+(j)\}$ .Finalmente, el conjunto $\mathcal{R} = \mathcal{R}_1 \cup \mathcal{R}_2$ en donde $\mathcal{R}_1 = \{(k,m,i) : (k,m) \in A_\mathcal{K}, i = O_k\}$ y $\mathcal{R}_2 = \{(k,m,i) : (k,m) \in A_\mathcal{K}, i \in V\}$. Estos conjuntos servirán para definir el espacio de las variables de decisión. Se definen las siguientes variables de decisión:

Sea x_{ij}^{km} una variable binaria que toma el valor de 1 si el vehículo m del depósito k va de i hasta j con $(k,m,i,j) \in \mathcal{K}$, y 0 en caso contrario.

Sea y_i^{km} una variable binaria que toma el valor de 1 si el vehículo m del depósito k visita a i con $(k,m,i) \in \mathcal{R}$, y 0 en caso contrario.

Sea a_i^{km} una variable entera positiva que representa la cantidad de producto recogido en i por el vehículo m del depósito k con $(k,m,i) \in \mathcal{R}$.

Modelo:

$$\text{mín} \quad \sum_{(k,m,i,j)\in\mathcal{K}} t_{ij} X_{ij}^{km} + \mathcal{C}, \tag{1}$$

$$\text{s.a.} \quad \sum_{j\in\delta^-(O_k)} X_{O_kj}^{km} = 1, \qquad \forall (k,m) \in A_\mathcal{K}, \tag{2}$$

$$\sum_{j\in\delta^+(Au_0^{km})} X_{jAu_0^{km}}^{km} = 1, \qquad \forall (k,m) \in A_\mathcal{K}, \tag{3}$$

$$\sum_{j\in\delta^-(i)} X_{ji}^{km} = y_i^{km}, \qquad \forall (k,m) \in A_\mathcal{K}, i \in V, \tag{4}$$

$$\sum_{i\in\delta^-(s)} X_{is}^{km} = \sum_{j\in\delta^+(s)} X_{sj}^{km}, \qquad \forall (k,m) \in A_\mathcal{K}, s \in V, \tag{5}$$

$$u_j \geq u_i + (1+n)x_{ij}^{km} - n, \qquad \forall (k,m,i,j) \in \mathcal{K}, \tag{6}$$

$$\sum_{(k,m)\in A_\mathcal{K}} a_i^{km} \leq o_i, \qquad \forall i \in V, \tag{7}$$

$$\sum_{i\in V} a_i^{km} = Au_1^{km}, \qquad \forall (k,m) \in A_\mathcal{K}, \tag{8}$$

$$a_i^{km} \leq o_i y_i^{km}, \qquad \forall (k,m) \in A_\mathcal{K}, i \in V, \tag{9}$$

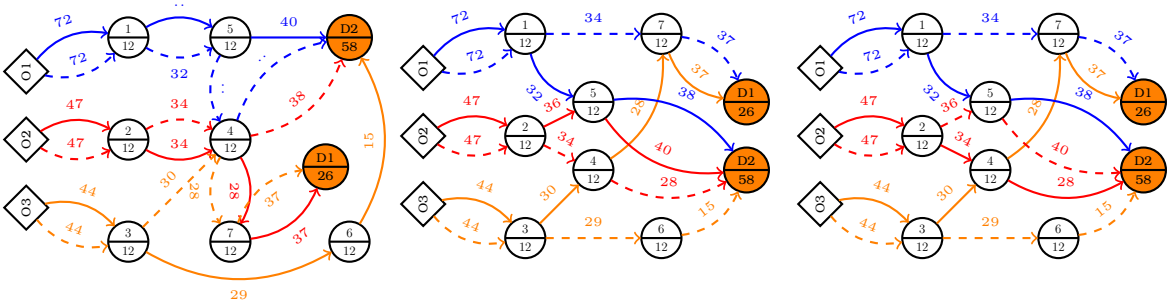
$$x_{ij}^{km} \in \{0,1\}, \qquad \forall (k,m,i,j) \in \mathcal{K}, \tag{10}$$

$$y_{i}^{km} \in \{0,1\}, \qquad \forall (k,m,i) \in \mathcal{R}, \tag{11}$$

$$a_{i}^{km} \in \mathbb{Z}^{+}, \qquad \forall (k,m,i) \in \mathcal{R}. \tag{12}$$

La función objetivo (1) minimiza los costos totales de los caminos; la Restricción (2) asegura que un vehículo debe partir desde su fuente mientras que la Restricción (3) garantiza la llegada del mismo al lugar de destino establecido en la primera fase; la Restricción (4) relaciona la variable y con la x; la Restricción (5) es una restricción de conservación de flujo; para controlar que no existan ciclos ni subtours se establece la Restricción (6); la Restricción (7) controla la oferta en cada lugar de abastecimiento; la Restricción (8) obliga al vehículo a transportar la cantidad de producto asignado en la primera fase; la Restricción (9) es de activación: si se visita el nodo de abastecimiento el vehículo puede recoger producto.

Para nuestro ejemplo, consideremos las soluciones de la primera fase presentadas en la Figura 2.7, al enrutar los vehículos las soluciones se representa en las Figuras 2.8 y 2.9.



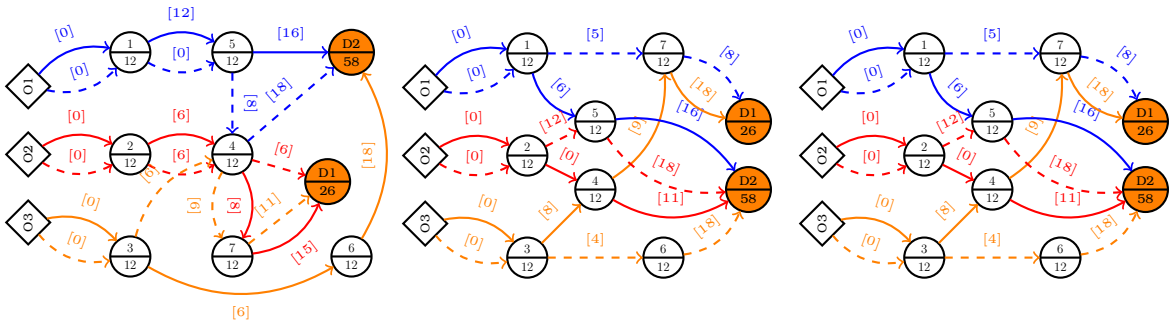
(a) Solución con 2.7a
Objetivo: 3502

(b) Solución con 2.7b
Objetivo: 3451

(c) Solución con 2.7c
Objetivo: 3451

Figura 2.8: Enrutamiento de vehículos de la primera fase

Se puede observar que existen diferentes soluciones óptimas para la primera fase de la heurística, esto implica que se obtendrá diferentes enrutamientos factibles con diferentes costos usando la segunda fase. Cabe recalcar de que se puede conjeturar que: si la primera fase logra hacerse con una combinación óptima de vehículos, entonces se obtendrá una solución óptima en la segunda fase (ver siguiente Sección). Por ejemplo, para la instancia de la Figura 2.3 una combinación óptima de vehículos se la presenta en la Figura 2.10; cabe dejar claro que no va a ser la única



(a) Solución con 2.7a (b) Solución con 2.7b (c) Solución con 2.7c

Figura 2.9: Recolección del producto

combinación óptima de vehículos En las Figuras 2.4 y 2.5 se muestra dos enrutamientos diferentes con dos combinaciones de vehículos diferentes donde estos recogen el producto de manera distinta.

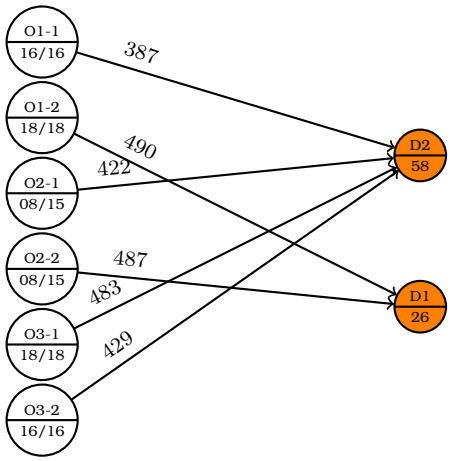
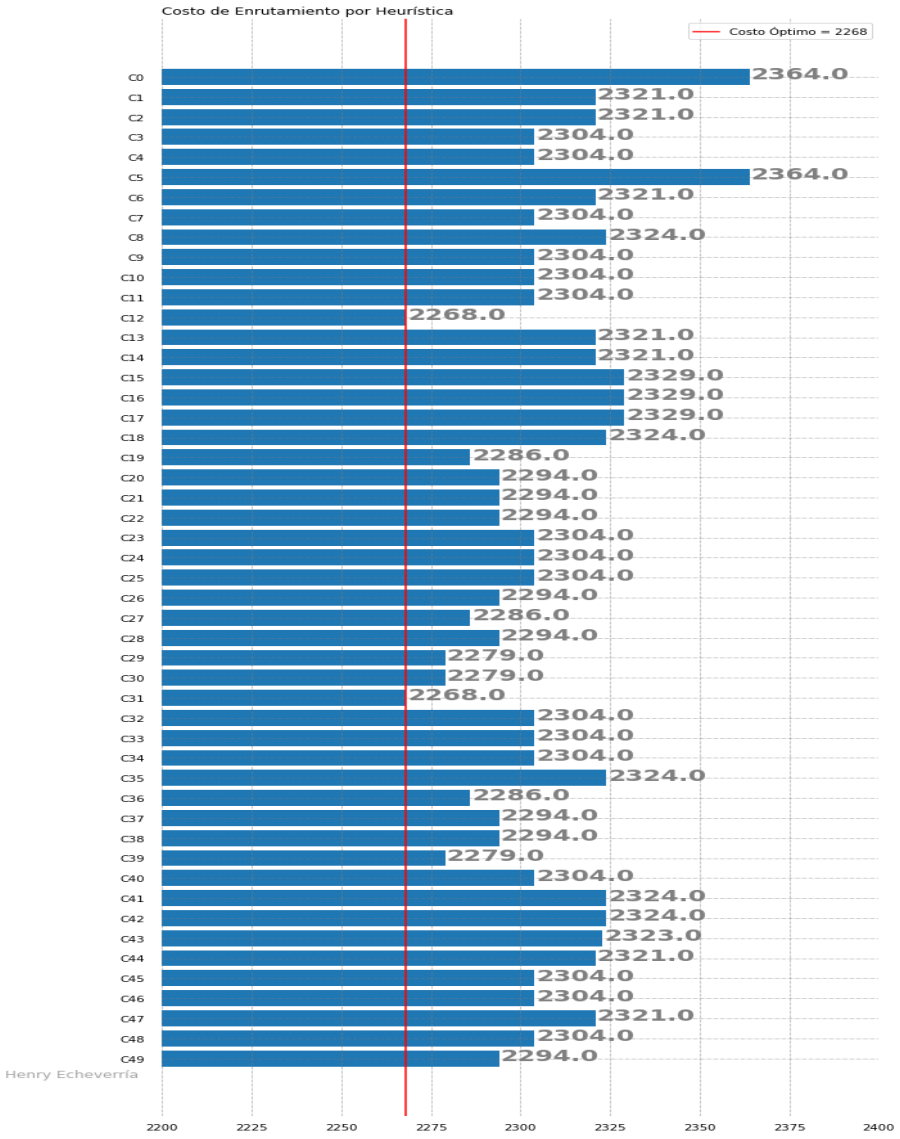


Figura 2.10: Combinación de Vehículos que da una solución óptima

Conjetura de la Heurística

Como se mencionó en la Sección anterior, se puede conjeturar que: si la primera fase logra hacerse con una combinación óptima de vehículos, entonces se obtiene un enrutamiento óptimo en la segunda fase. Para identificar de mejor manera que es una combinación óptima de vehículos se crea una instancia fácil de resolver con el objetivo de llegar al valor óptimo de la misma. Posteriormente, se aplica la primera fase en esta instancia para obtener 50 diferentes combinaciones de vehículos, y con estas combinaciones se enrutan los vehículos en la segunda fase. La Figura 2.4.2 muestra los resultados para cada combinación de vehículos, y

vemos que con la combinación 12 ($C12$) y combinación 31 ($C31$) se obtiene la solución óptima, por lo tanto esas combinaciones de vehículos serán combinaciones óptimas de vehículos.

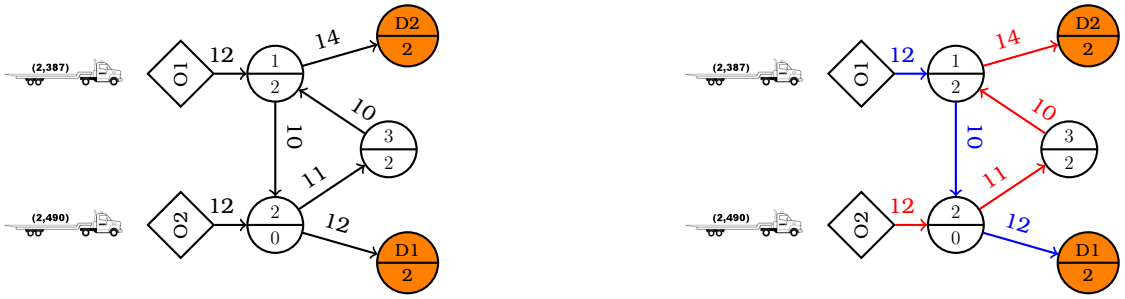


La idea para demostrar la propiedad anterior es saber diferenciar las combinaciones óptima de vehículos entre todas las combinaciones de vehículos posibles; pues en general no se sabe cual es el valor óptimo por la dificultad computacional del problema. Es válido preguntarse *"¿Qué relación hay entre $C12$ y $C31$? y ¿Qué diferencia hay entre $C12$, $C31$ y las demás combinaciones?"*, el objetivo de responder las preguntas es mejorar el espacio de búsqueda para las soluciones de la primera fase y obtener mejores resultados en la segunda fase. Finalmente, dado al alcance del trabajo de integración, no se estudiará a fondo las posibles propiedades

que se generan a partir de las heurísticas (una de ellas es la mencionada anteriormente).

2.5. Observaciones a los Modelos 1 y 2

Consideremos una instancia muy simple para realizar algunas observaciones a los modelos. La instancia se presenta en la Figura 2.11a y es fácil ver que una **solución factible** es presentada en la Figura 2.11b. Por otro lado, si se desea saber si esta solución es óptima o si existe una mejor es necesario usar el Solver *Gurobi*. Cuando se implementa la instancia y se compilan los modelos, *Gurobi* no da una solución o en otras palabras es infactible, pero esto no cuadra con el resultado presentado en la Figura 2.11b. A consecuencia de ello surge la pregunta *¿Por qué Gurobi (o cualquier otro Solver) no proyecta una solución factible?*. Para responder la pregunta es necesario hacer un análisis más profundo a la instancia presentada, a la solución en la Figura 2.11b y obviamente a los modelos de programación lineal entera (Modelo 1 y Modelo2).



(a) (b)

Figura 2.11: Instancia y Enrutamiento para Observación

Primero en los modelos **Modelo 1** y **Modelo 2** tenemos la restricción para evitar los ciclos: $u_j \geq u_i + (1+n)x_{ij}^{km} - n$. Observemos que las variables de ordenamiento u_i no se encuentras indexadas por los vehículos. A consecuencia de ello, el Modelos 1 y el Modelo 2 buscarán soluciones que no posea ciclos en ninguna parte del Digrafo Solución, gracias a esto se tiene el siguiente resultado:

Resultado. *Para toda instancia del problema que sea factible sobre el* ***Modelo 1*** *y* ***Modelo 2*** *se define* $G' = (V \cup \mathcal{O} \cup \mathcal{D}, A')$ *el Digrafo de caminos óptimos donde* A' *es el conjunto de arcos de la solución óptima, entonces el Digrafo* G' *no presenta ningún ciclo.*

Una consecuencia directa del resultado anterior es que los caminos presentados en la Figura 2.12 no se van a dar (considerando que cada color representa un vehículo), y como podemos observar la solución mostrada en la Figura 2.11b presenta un ciclo y es: $(1) \longrightarrow (2) \longrightarrow (3) \longrightarrow (1)$.

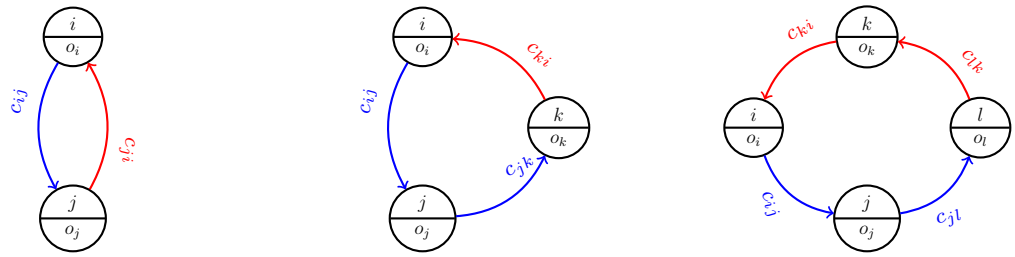


Figura 2.12: Caminos no posibles en las soluciones

Por ello para encontrar soluciones que presenten ciclos entre los vehículos, se redefine la restricción $u_j \geq u_i + (1+n)x_{ij}^{km} - n$ por $u_j^{km} \geq u_i^{km} + (1+n)x_{ij}^{km} - n$, así obtenemos el **Modelo 3** y **Modelo 4** que serían el **Modelo 1** y **Modelo 2**, respectivamente pero con la restricción que evita ciclos modificada.

Capítulo 3

Resultados Computacionales

3.1. Pruebas Computacionales

Existen diferentes solvers para resolver el programa lineal entero; para realizar las pruebas computacionales usamos el solver *Gurobi* el cual es un solver comercial que se auto denomina como el más rápido y más potente solver en todo el mundo para ciertos problemas, en particular para los programas lineales enteros.

Se presentan a continuación 52 instancias las cuales han sido implementadas en *Python 3* y sus parámetros (ofertas, demandas, vehículos, costos, etc.) han sido generados a partir de números (pseudo) aleatorios con una función del paquete *NumPy* la cual devuelve enteros de una distribución uniforme discreta. Todas las instancias fueron ejecutadas en *Gurobi 9.5* en una computadora portátil con las siguientes características: Intel Core i7 8va Gen,16 GB Memoria RAM y Windows 11 64-bit.

Se reporta el valor de la Función Objetivo, GAP de dualidad y el tiempo de cómputo para comparar el **Modelo 1**, **Modelo 2**, **Modelo 3** y **Modelo 4**.Para más información de los parámetros, en el repositorio de *Git-Hub* [3] se encuentra detallado.

Tabla 3.1: Soluciones Instancias: Modelo 1, Modelo 2 y Heurística

Parámetros				MODELO 1			MODELO 2			HEURISTICA	
Abastecimientos	**Fuentes**	**Destinos**	**Arcos**	**OBJ**	**GAP[%]**	**Tiempo[s]**	**OBJ**	**GAP[%]**	**Tiempo[s]**	**OBJ**	**Tiempo[s]**
5	**11**	**17**	132	18252	3.10	3600.241	18252	10.92	3600.107	18906	0.215515
	14	**17**	146	18053	16.91	3600.111	18053	12.01	3600.165	18906	0.221201
	17	**2**	95	6903	0.00	2131.61	6903	0.00	218.7547	7004	0.171654
	18	**6**	113	9480	2.95	3600.094	9478	2.82	3600.111	9728	0.173773
10	**11**	**10**	224	9094	0.00	2042.168	9094	0.00	237.8497	9531	0.73715
	15	**4**	209	8335	18.25	3600.107	8334	0.00	716.5548	8573	1.859671
	17	**9**	269	9123	17.71	3600.084	9123	12.28	3600.068	9462	1.094561
	21	**11**	314	10775	10.20	3600.108	10796	8.68	3600.133	11101	1.213198
15	**6**	**5**	256	5419	0.94	3600.115	5419	0.85	3600.063	5693	8.730991
	7	**6**	278	5043	2.89	3600.057	5073	20.37	3600.065	5247	5.478151
	18	**7**	425	10454	21.63	3600.095	10443	10.06	3600.164	10814	35.93274
	22	**9**	500	13701	24.15	3600.121	13708	8.52	3600.161	14220	7.161972
20	**12**	**2**	428	5278	2.38	3600.087	5287	3.20	3600.08	5295	14.21498
	13	**8**	543	8266	21.16	3600.193	8229	22.78	3600.158	8515	68.24565
	19	**10**	678	9992	19.08	3600.118	9997	16.17	3600.126	10450	5.298611
	24	**4**	665	10457	3.93	3600.092	10427	4.15	3600.131	10345	143.1142
25	**11**	**12**	784	12040	15.71	3600.112	12039	0.34	3600.091	12867	144.2387
	16	**9**	830	9515	25.99	3600.106	9508	23.94	3600.097	9820	600.1147
	17	**17**	1023	16763	15.45	3600.207	16834	15.80	3600.308	17453	600.249
	24	**17**	1173	17185	17.33	3600.228	17183	4.83	3600.419	18171	600.1604
30	**10**	**2**	745	4603	1.94	3600.099	4625	3.01	3600.145	4593	444.0031
	15	**3**	899	8197	8.35	3600.096	8161	3.13	3600.059	8228	600.1092
	20	**12**	1251	12637	24.75	3600.475	12666	25.08	3600.155	13031	600.1918
	24	**16**	1459	15537	37.45	3600.184	15491	0.55	3601.641	16438	600.3077
35	**6**	**15**	1215	12577	34.79	3600.189	12606	34.95	3600.294	12801	900.1635
	15	**2**	1102	6663	1.34	3600.319	6665	4.66	3600.759	6719	215.8501
	19	**5**	1316	10546	16.31	3600.139	10554	16.85	3600.171	10907	600.1111
	23	**19**	1851	15028	34.95	3600.223	15082	35.20	3600.216	15662	600.2082
40	**6**	**4**	1106	5411	0.40	3600.102	5428	0.68	3600.195	5622	600.1903
	7	**8**	1270	9402	25.48	3600.099	9553	26.64	3600.093	9528	600.1311
	13	**8**	1473	9343	24.31	3600.11	9267	29.68	3600.143	9454	600.1168
	15	**15**	1788	inf		3600.14	inf		3600.193	16513	600.2734
45	**14**	**13**	2005	inf		3600.213	inf		3600.149	13728	600.2163
	19	**9**	2045	9229	9.25	3600.26	9219	10.66	3602.039	9704	600.1865
	23	**14**	2395	13946	26.96	3600.181	13967	27.06	3600.203	14254	600.1587
	25	**15**	2508	inf		3600.153	inf		3600.258	15809	600.4357
50	**13**	**13**	2319	inf		3600.119	inf		3600.14	14654	600.1772
	14	**14**	2403	inf		3600.107	inf		3600.097	15716	600.2504
	16	**13**	2446	inf		3600.125	inf		3600.129	15453	900.2689
	21	**10**	2524	inf		3600.193	inf		3600.255	10724	600.3153
55	**9**	**6**	2160	7501	16.36	3600.117	7587	2.89	3600.079	7787	600.1516
	12	**9**	2447	10072	37.35	3600.22	10109	37.57	3600.281	10217	600.4973
	23	**9**	2960	inf		3600.165	11706	19.49	3600.701	11190	600.3594
	24	**15**	3296	inf		3600.086	inf		3600.105	16761	900.3558
60	**10**	**6**	2553	7493	14.65	3600.162	7521	20.22	3600.175	7453	600.183
	17	**20**	3638	inf		3600.032	inf		3600.076	21544	600.4622
	18	**2**	2762	8511	9.80	3600.153	8394	3.36	3600.192	8352	600.2261
	23	**18**	3837	inf		3600.038	inf		3600.116	20554	2000.452
65	**15**	**9**	3369	inf		3600.268	11072	26.88	3600.171	11085	2000.312
	19	**12**	3759	inf		3600.131	inf		3600.118	14133	600.2887
	22	**15**	4088	inf		3600.089	inf		3600.076	17255	600.2391
	25	**19**	4465	inf		3600.048	inf		3600.055	20658	4200.502

Tabla 3.2: Soluciones Instancias: Modelo 3 y Modelo 4

Parámetros				MODELO 3			MODELO 4		
Abastecimiento	**Fuentes**	**Destinos**	**Arcos**	**OBJ**	**GAP[%]**	**Tiempo[s]**	**OBJ**	**GAP[%]**	**Tiempo[s]**
5	**11**	**17**	132	18252	3.96	3600.15	18252	0.00	2719.5
	14	**17**	146	18053	0.00	3405.01	18053	23.21	3600.17
	17	**2**	95	6903	0.00	1253.44	6903	0.00	189.185
	18	**6**	113	9478	4.50	3600.12	9478	3.69	3600.11
10	**11**	**10**	224	9094	0.00	339.985	9094	0.00	1312.1
	15	**4**	209	8334	0.00	3251.09	8334	0.00	608.142
	17	**9**	269	9123	17.01	3600.09	9123	10.17	3600.12
	21	**11**	314	10768	10.26	3600.61	10773	9.27	3600.4
15	**6**	**5**	256	5427	24.61	3600.07	5419	0.82	3600.09
	7	**6**	278	5048	2.97	3600.04	5048	2.71	3600.06
	18	**7**	425	10445	20.44	3600.09	10473	10.02	3600.13
	22	**9**	500	13701	24.13	3600.1	13701	12.44	3600.09
20	**12**	**2**	428	5276	1.05	3600.11	5287	1.63	3600.06
	13	**8**	543	8211	20.59	3600.14	8247	25.17	3601.47
	19	**10**	678	9999	19.80	3600.12	9992	18.97	3600.17
	24	**4**	665	10452	4.10	3600.08	10433	4.14	3600.1
25	**11**	**12**	784	12066	17.87	3600.11	12040	3.53	3600.09
	16	**9**	830	9564	15.96	3600.95	9541	20.69	3600.15
	17	**17**	1023	16762	15.62	3600.15	16772	15.45	3600.23
	24	**17**	1173	17323	35.60	3600.14	17180	6.97	3600.13
30	**10**	**2**	745	4623	2.53	3600.09	4596	1.85	3600.17
	15	**3**	899	8168	6.07	3602.83	8161	2.99	3602.03
	20	**12**	1251	12799	24.72	3600.11	12643	24.81	3600.17
	24	**16**	1459	15559	37.46	3600.18	15504	0.63	3600.14
35	**6**	**15**	1215	12529	34.54	3600.17	12631	35.10	3600.14
	15	**2**	1102	6682	4.95	3600.24	6654	1.25	3600.19
	19	**5**	1316	10571	18.57	3600.11	10682	20.32	3600.19
	23	**19**	1851	29252	66.55	3600.21	15163	35.52	3600.23
40	**6**	**4**	1106	5414	0.40	3600.17	5428	0.73	3600.1
	7	**8**	1270	9526	26.42	3600.1	9491	26.13	3600.12
	13	**8**	1473	9350	26.67	3600.12	9293	3.54	3600.12
	15	**15**	1788	16367	14.67	3600.17	16358	14.61	3600.17
45	**14**	**13**	2005	inf		3600.25	inf		3600.14
	19	**9**	2045	9218	10.55	3600.43	9381	12.10	3600.2
	23	**14**	2395	inf		3600.14	13911	26.69	3600.17
	25	**15**	2508	inf		3600.3	inf		3600.13
50	**13**	**13**	2319	14861	13.36	3600.18	14972	14.01	3600.25
	14	**14**	2403	inf		3600.13	15905	15.63	3600.15
	16	**13**	2446	inf		3600.13	inf		3600.15
	21	**10**	2524	10467	11.92	3600.41	10598	12.98	3600.26
55	**9**	**6**	2160	7558	12.17	3600.45	7586	15.74	3600.15
	12	**9**	2447	10092	33.88	3600.14	10105	37.50	3600.23
	23	**9**	2960	11888	20.72	3600.45	11874	20.65	3600.38
	24	**15**	3296	inf		3600.1	inf		3600.1
60	**10**	**6**	2553	7571	15.32	3600.27	7456	19.21	3600.19
	17	**20**	3638	inf		3600.04	inf		3600.04
	18	**2**	2762	8384	2.94	3600.13	8395	3.18	3600.27
	23	**18**	3837	inf		3600.11	inf		3600.14
65	**15**	**9**	3369	10964	26.16	3600.14	10914	25.96	3600.14
	19	**12**	3759	14562	30.08	3600.35	inf		3600.08
	22	**15**	4088	inf		3600.1	inf		3601.37
	25	**19**	4465	inf		3600.05	inf		3600.06

Tabla 3.3: Modelo 2 e Inicialización con Heuristica

Párametros				MODELO 2			MODELO 2 + Heurística		
Abastecimientos	**Fuentes**	**Destinos**	**Arcos**	**OBJ**	**GAP[%]**	**Tiempo[s]**	**OBJ**	**GAP[%]**	**Tiempo[s]**
5	**11**	**17**	132	18252	10.92	3600.11	18252	1.95	3600.12
	14	**17**	146	18053	12.01	3600.16	18053	13.39	3600.18
	17	**2**	95	6903	0.00	218.75	6903	0.00	260.43
	18	**6**	113	9478	2.82	3600.11	9478	2.64	3600.07
10	**11**	**10**	224	9094	0.00	237.85	9094	0.00	1097.88
	15	**4**	209	8334	0.00	716.55	8334	0.00	437.47
	17	**9**	269	9123	12.28	3600.07	9123	11.75	3600.06
	21	**11**	314	10796	8.68	3600.13	10780	10.11	3600.15
15	**6**	**5**	256	5419	0.85	3600.06	5419	0.71	3600.06
	7	**6**	278	5073	20.37	3600.06	5043	2.63	3600.11
	18	**7**	425	10443	10.06	3600.16	10449	13.14	3600.10
	22	**9**	500	13708	8.52	3600.16	13712	17.46	3600.10
20	**12**	**2**	428	5287	3.20	3600.08	5271	2.67	3600.09
	13	**8**	543	8229	22.78	3600.16	8213	24.96	3600.14
	19	**10**	678	9997	16.17	3600.13	9997	18.58	3600.13
	24	**4**	665	10427	4.15	3600.13	10196	1.91	3600.22
25	**11**	**12**	784	12039	0.34	3600.09	12039	1.81	3600.11
	16	**9**	830	9508	23.94	3600.10	9566	20.78	3600.10
	17	**17**	1023	16834	15.80	3600.31	16804	15.65	3600.18
	24	**17**	1173	17183	4.83	3600.42	17204	17.77	3600.15
30	**10**	**2**	745	4625	3.01	3600.15	4593	2.39	3600.11
	15	**3**	899	8161	3.13	3600.06	8194	3.58	3600.09
	20	**12**	1251	12666	25.08	3600.16	12726	25.40	3600.11
	24	**16**	1459	15491	0.55	3601.64	15490	0.54	3600.18
35	**6**	**15**	1215	12606	34.95	3600.29	12510	34.51	3600.13
	15	**2**	1102	6665	4.66	3600.76	6682	5.06	3600.16
	19	**5**	1316	10554	16.85	3600.17	10527	16.61	3600.18
	23	**19**	1851	15082	35.20	3600.22	15073	35.15	3600.22
40	**6**	**4**	1106	5428	0.68	3600.20	5414	0.44	3600.08
	7	**8**	1270	9553	26.64	3600.09	9419	22.87	3600.12
	13	**8**	1473	9267	29.68	3600.14	9336	4.15	3600.12
	15	**15**	1788	inf		3600.19	16299	14.35	3600.15
45	**14**	**13**	2005	inf		3600.15	13415	4.16	3600.15
	19	**9**	2045	9219	10.66	3602.04	9254	10.98	3600.26
	23	**14**	2395	13967	27.06	3600.20	14009	27.27	3600.20
	25	**15**	2508	inf		3600.26	15665	25.17	3600.31
50	**13**	**13**	2319	inf		3600.14	14582	11.75	3600.12
	14	**14**	2403	inf		3600.10	15526	13.56	3600.19
	16	**13**	2446	inf		3600.13	15044	43.03	3600.13
	21	**10**	2524	inf		3600.25	10486	12.08	3600.18
55	**9**	**6**	2160	7587	2.89	3600.08	7696	27.70	3600.15
	12	**9**	2447	10109	37.57	3600.28	10097	37.51	3600.11
	23	**9**	2960	11706	19.49	3600.70	11117	15.22	3600.16
	24	**15**	3296	inf		3600.10	16149	32.11	3600.16
60	**10**	**6**	2553	7521	20.22	3600.18	7381	18.58	3600.16
	17	**20**	3638	inf		3600.08	21384	31.40	3600.05
	18	**2**	2762	8394	3.36	3600.19	8341	2.76	3600.27
	23	**18**	3837	inf		3600.12	19850	45.94	3600.03
65	**15**	**9**	3369	11072	26.88	3600.17	11028	26.58	3600.22
	19	**12**	3759	inf		3600.12	14019	27.36	3600.32
	22	**15**	4088	inf		3600.08	16996	35.80	3600.04
	25	**19**	4465	inf		3600.05	20213	43.77	3600.05

3.2. Comentarios a las pruebas computacionales

En la Tabla 3.1 se puede observar los resultados de las instancias resueltas con el **Modelo 1**, el **Modelo 2** y la **Heurística**. Con respecto a los valores objetivos de los modelos, estas están muy cercanas entre si. Pero es necesario enfatizar que en las instancias $(55, 23, 9)$ y $(65, 15, 9)$ el **Modelo 2** da una solución factible mientras que el **Modelo 1** no. Por otro lado, no existe ninguna instancia en donde el **Modelo 1** de una solución factible y el **Modelo 2** no. Adicionalmente, la **Heurística** planteada en este trabajo da muy buenos resultados teniendo en cuenta al tiempo de cómputo que toma resolver las instancias con el **Modelo 1** y el **Modelo 2** y a los resultados de los valores objetivos; hay que recalcar que en ciertas instancias la **Heurística** da mejores valores objetivos que los modelos; por ejemplo, en las instancias: $(20, 24, 4)$, $(40, 7, 8)$, $(55, 23, 9)$ y $(60, 18, 2)$. No menos importante, la **Heurística** da soluciones factibles en todas las instancias.

A consecuencia de ello, usando las soluciones factibles proporcionadas por la **Heurística** y junto al **Modelo 2** en la mayoría de las instancias se obtuvo mejores resultados, y más importante es recalcar que se registran soluciones factibles en las instancias en donde inicialmente no se obtuvieron resultados dado a su complejidad, esto se puede observar en la Tabla 3.3; por ejemplo, en la instancia $(45, 14, 13)$ se obtuvo una solución factible aplicando la **Heurística** con un límite de tiempo de cómputo de 600 segundos, y cuando usamos esa solución factible en el **Modelo 2** aplicando un límite de tiempo de cómputo de 3600 segundos se obtiene un $GAP = 4.16\,\%$. Entonces el tiempo de cómputo total es de 4200 segundos. En cambio, usando el **Modelo 2** sin una solución inicial y con un límite de tiempo de cómputo de 3600 segundos, no se encuentra solución factible y probablemente en 4200 segundos tampoco se vaya a obtenerla.

Finalmente, en la Tabla 3.2 se puede observar los resultados de las instancias resueltas con el **Modelo 3** y el **Modelo 4**. En general, comparado a los otros modelos, estos resultan ser mejores en las instancias grandes; esto por que el **Modelo 3** y el **Modelo 4** dan soluciones factibles

mientras que en los otros dos modelos no; por ejemplo, en las instancias $(50, *, *)$ el **Modelo 3** registra 2 de 5 soluciones factibles y el **Modelo 4** registra 3 de 5, en cambio el **Modelo 1** y el **Modelo 2** no registran ninguna solución factible.

Capítulo 4

Conclusiones y recomendaciones

4.1. Conclusiones

En conclusión, gracias a la amplia literatura que aborda el *VRP* fue posible formular diferentes modelos de Programación Lineal Entera para el problema planteado en este trabajo de integración curricular. Los modelos han sido implementados computacionalmente con ayuda del solver *Gurobi* y el lenguaje de programación en *Python 3*. Estos modelos se pusieron a prueba con 52 instancias computacionales generadas a partir de funciones de números aleatorios de *Numpy* y en la mayoría de ellas, a pesar de que se llegó a soluciones factibles, no se alcanzó a obtener soluciones óptimas en un tiempo limite de una hora; esto se debe a la complejidad del problema pues al ser una variante del *VRP* este pertenece a la clase *NP-Hard*.

Gracias a los resultados obtenidos en las pruebas computacionales reflejadas en las Tablas 3.1, 3.2 y 3.2 se puede concluir a priori que los modelos que consideran todas las variables de ordenamiento (**Modelo 3** y **Modelo 4**) dan mejores resultados, al menos en la mayoría de las instancias, que los otros modelos (**Modelo 1** y **Modelo 2**).

Dada la complejidad del problema, fue necesario el estudio de métodos heurísticos para el problema; el método que se empleó fue *cluster-first, route-second*, el cual cuenta con dos diferentes fases y que fueron resu-

letas con Programación Lineal Entera. Es importante recalcar que en la primera fase se puede obtener diferentes soluciones óptimas por lo que, junto a la segunda fase, alguna de ellas nos puede conducir a la solución óptima (enrutamiento óptimo) en un tiempo razonable.

4.2. Recomendaciones

Los modelos obtenidos pueden ser usados en diferentes campos de la industria de recolección de productos por lo que sería conveniente aplicarlos a una o varias instancias que posean datos del mundo real y ver la diferencia de su comportamiento respecto a las instancias artificiales. Adicionalmente, se puede hacer un estudio más profundo de la generación de las instancias artificiales para que estas puedan acercarse lo más posible a una instancia que se presentaría en el mundo de la industria de recolección y entrega de productos.

Además, se puede encontrar diferentes variantes del mismo problema; por ejemplo, se puede considerar que un vehículo puede entregar en diferentes lugares. Otra opción sería estudiar un problema más extenso considerando que una vez entregado el producto este se reparte a diferentes lugares de venta del producto. El problema cobraría más realismo si se consideraran también ventanas de tiempo.

Respecto a los métodos de resolución, en vez de aplicar Programación Lineal Entera se puede abordar el problema usando métodos combinatorios. Como se mencionó en le estado del arte, existen diferentes tipos de heurística, una alternativa es implementar una heurística iterativa con ello se tendría una soluciones factibles en tiempos menores a los presentados en la tabla 3.1.

Finalmente, dado que el **Modelo 3** y el **Modelo 4** resultaron mejores, es conveniente usarlos con una solución factible proporcionada por la **Heurística** para obtener mejores resultados.

Referencias bibliográficas

[1] GM Appa. The transportation problem and its variants. *Journal of the Operational Research Society*, 24(1):79–99, 1973.

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

[3] Henry Echeverría. Instancias computacionales. https://github.com/PyHenry08090/TrabajoIntregacion-VRP, 2022.

[4] Serap Ercan Cömert, Harun Yazgan, Sena Kır, and Furkan Yener. A cluster first-route second approach for a capacitated vehicle routing problem: A case study. *International Journal of Procurement Management*, 11:399, 01 2018.

[5] Miguel Andres Figliozzi. An iterative route construction and improvement algorithm for the vehicle routing problem with soft time windows. *Transportation Research Part C: Emerging Technologies*, 18(5):668–679, 2010.

[6] Annisa Kesy Garside and Nabila Rohmatul Laili. A cluster-first route-second heuristic approach to solve periodic multi-trip vehicle routing problem.

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

[8] Asefeh Hasani-Goodarzi and Reza Tavakkoli-Moghaddam. Capacitated vehicle routing problem for multi-product cross- docking with

split deliveries and pickups. *Procedia - Social and Behavioral Sciences*, 62:1360–1365, 2012. World Conference on Business, Economics and Management (BEM-2012), May 4–6 2012, Antalya, Turkey.

[9] Manuel Laguna and Rafael Martí. *Heuristics*, pages 695–703. Springer US, Boston, MA, 2013.

[10] Gilbert Laporte. Fifty years of vehicle routing. *Transportation Science*, 43(4):408–416, 2009.

[11] Gilbert Laporte and Frédéric Semet. *5. Classical Heuristics for the Capacitated VRP*, pages 109–128. 01 2002.

[12] Yannis Marinakis, Magdalene Marinaki, and Athanasios Migdalas. *Variants and Formulations of the Vehicle Routing Problem*, pages 91–127. Springer International Publishing, Cham, 2018.

[13] Paweł Sitek and Jarosław Wikarek. Capacitated vehicle routing problem with pick-up and alternative delivery (cvrppad): model and implementation using hybrid approach. *Annals of Operations Research*, 273, 02 2019.

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

[15] Chrysafis Vogiatzis and Panos M. Pardalos. *Combinatorial Optimization in Transportation and Logistics Networks*, pages 673–722. Springer New York, New York, NY, 2013.