

# Optimización del Transporte en la Logística de Reforestación

Reporte Final



José de Jesús Ramírez Mendieta  
Ricardo Kaleb Flores Alfonso  
Juan José de Jesús Hernández Beltrán  
Kevin Jesús Martínez Trinidad

A00835680  
A00836747  
A01198716  
A00834493

# 1 Introducción

Según datos de la Comisión Nacional Forestal, usando una metodología aprobada internacionalmente con el Sistema Satelital de Monitoreo Forestal (SAMOF), el cual fotointerpreta imágenes satelitales, en promedio, México ha demostrado una tasa anual de deforestación de aproximadamente 208,850 hectáreas por año durante el periodo 2001-2019, lo que representa un 0.31% de la superficie forestal arbolada a nivel nacional. En 2020 se concluyó que la principal fuente de deforestación es la tala de árboles para la agricultura y la tala ilegal, sumando un 80% de la deforestación tropical (Comisión Nacional Forestal, 2022).

La deforestación afecta directamente el ciclo del agua al reducir la capacidad de los bosques para capturar y regular el flujo hídrico, lo que agrava las sequías y la escasez de agua para consumo humano, industria y agricultura. Además, los bosques son importantes para la recarga de acuíferos y la prevención de inundaciones.

La biodiversidad también se ve afectada al destruir hábitats naturales y liberar grandes cantidades de carbono almacenado en los árboles. Además, se pierde la regulación climática y la prevención de la erosión. Se ha llegado a estimar que la deforestación es responsable de aproximadamente el 10% de las emisiones globales de gases de efecto invernadero.

En respuesta, el Gobierno de México junto con la SEMARNAT y sus organismos han iniciado estrategias que giran en torno a la prevención, la inspección y verificación, inteligencia, judicialización de casos, acompañamiento social y revisión del marco jurídico para combatir la deforestación y tala ilegal. Asimismo, para contribuir a la restauración de los ecosistemas, la elección de especies para reforestación se basa en criterios como adaptabilidad al clima local, diversidad y beneficios ecológicos. La distribución de las especies se planifica considerando las condiciones del sitio y los objetivos de restauración.

## 1.1 Justificación del Problema

El cambio climático y las sequías están siendo problemas cada vez mayores, no solamente para mantener los ecosistemas en armonía, sino también para las actividades humanas.

La deforestación es una de las principales causas de que existan sequías, además de que acelera el cambio climático. En nuestro país, la deforestación es un problema mucho mayor de lo que se cree, pues muchas veces este se realiza de forma ilegal o irresponsable.

Los programas de reforestación permiten reducir los efectos negativos de la deforestación. Organizaciones como la CONAFOR reciben una limitada cantidad de recursos y tienen un plazo máximo de unos cuantos meses al año para realizar la reforestación.

Por consiguiente, un modelo que indique cuáles son las rutas óptimas y el momento adecuado para realizarlas, reduciendo así el tiempo y gasto económico de la actividad, es necesario para aumentar lo más posible las probabilidades de finalizar exitosamente la empresa, utilizando la menor cantidad de recursos posibles.

## 1.2 Objetivo

El objetivo es minimizar tanto la distancia, y consecuentemente el tiempo, consumido por los camiones que transportan las plantas al lugar de sembrado. Para ello se plantea construir un plan de trabajo que contenga rutas ordenadas que les permitan hacer todas las entregas en menor tiempo posible.

## 1.3 Trabajo Relacionado

### 1.3.1 La Flora en el Altiplano Mexicano

El altiplano mexicano es un término que se refiere a la zona que se extiende desde la frontera de México con Estados Unidos hasta la latitud aproximada de la Ciudad de México, abarcando estados como Chihuahua, Coahuila, Nuevo León, Durango, San Luis Potosí, Jalisco, Puebla, entre otros. (Lifeder, 2020) En la zona sur del altiplano mexicano es común encontrarse con bosques de coníferos, en los cuales es posible encontrarse con especies como los pinos, la ceiba, el oyamel y la encina, así como las gramíneas ocasionalmente. (Lifeder, 2020)

- Pinos: Árboles característicos del bosque de hojas perenne, los cuales miden generalmente de 15 a 45m. (Masats, 2022)
- Ceiba: Son árboles muy grandes, de hojas caducifolias, con valor cultural e histórico. Suelen medir de 20 a 40 metros. (Gobierno de Yucatán, n.d.)
- Oyamel: Es un árbol de tamaño grande perennifolio, de 25 a 30m de altura y de 70 a 90cm de diámetros. Especie considerada en peligro de extinción principalmente por las actividades humanas, ya que es usada como combustible y leña. (SEMARNAT, 2018)
- Encina: Cuenta con una gran copa, con forma redondeada, y hojas frondosas, que lo hacen bueno para dar sombra. Llega a medir hasta 25m de altura y posee un tronco ancho y grueso. (Aqua, 2023)

Por otro lado, en la zona seca del altiplano las especies que predominan son las siguientes: (Lifeder, 2020)

- Cactus: El habitante característico del desierto, se caracterizan por almacenar grandes cantidades de agua que les permite sobrevivir a las grandes sequías. (Hernández, 2011)
- Maguey: Contribuye a la conservación y retención del suelo, al mismo tiempo que su jugo permite elaborar bebidas alcohólicas y es utilizado con fines medicinales. (Hernández, 2011)
- Ocotillo: Similar al cactus, es un árbol espinoso que requiere de poca agua para sobrevivir. (Hernández, 2011)
- Mezquite: Son árboles caducifolios que llegan a medir de 6 a 9 metros, se caracterizan por las propiedades de su madera, la cual es bastante utilizada para cocinar alimentos. (Múltiples autores, 2024)

### 1.3.2 Marco de Plantación Al Tresbolillo

En agricultura, es bien sabido que todas las plantas requieren cierto espacio para poder crecer y desarrollarse apropiadamente. Esta condición da origen a los marcos de plantación: la disposición espacial y distancia que existe entre las plantas. Cada cultivo tiene un marco de plantación recomendado, pues su apropiada elección favorecerá la iluminación, la luz y los nutrientes; y reducirá el riesgo de plagas y la propagación de enfermedades (Balam, 2022). Existe gran variedad de marcos de plantación (Figura 1), como el cuadrado, rectangular, cinco de oro, y el tresbolillo, solo por mencionar algunos (Balam, 2022).

Tres bolillos o tresbolillo es uno de los sistemas de plantación más utilizados en la actualidad. Este se basa en la plantación en forma de triángulos equiláteros, en donde cada vértice representa el lugar en donde una planta será sembrada, de esta forma nos aseguramos que cada planta tenga la misma separación con todas sus vecinas. La razón por la cual este sistema es tan utilizado se debe a que permite un mejor aprovechamiento del espacio de sembrado, permitiendo sembrar 17%

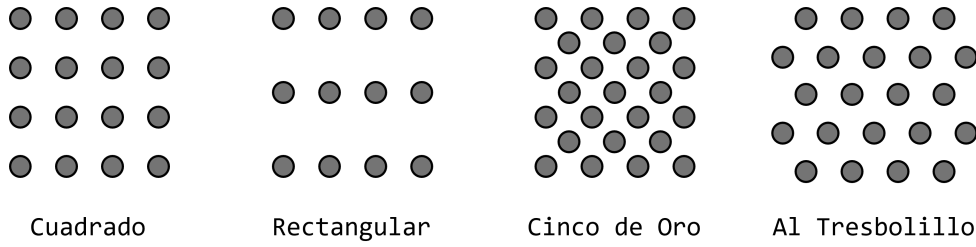


Figure 1: Principales marcos de plantación.

(Tips y Temas Agronómicos, 2016) más plantas comparado con el método de plantación rectangular; también permite un mejor control de la erosión y de la captación de luz, ya que la sombra de un árbol no se proyecta directamente sobre otros. (Iglesias, 2021) Para utilizar este sistema es necesario primero decidir cuánta distancia habrá entre cada planta, la cual depende de las características de esta y del entorno. Posteriormente, se calcula la distancia entre cada surco (filas), la cual es igual a la altura del triángulo equilátero (Guerrero, 2018).

$$h = d \cos 30 \quad (1)$$

donde  $h$  es la distancia entre surcos y  $d$  la distancia entre cada árbol. Por otro lado, es posible conocer la cantidad de plantas que caben en un espacio determinado utilizando trigonometría de la siguiente manera (Carbo, 1978):

$$n = \frac{A}{d^2 \cos 30} \quad (2)$$

donde  $n$  es el número de plantas,  $A$  es el área del terreno disponible y  $d$  es la distancia entre cada planta.

### 1.3.3 Problema de Ruteo de Vehículos (RVP)

El problema de Ruteo de Vehículos es uno de los problemas de optimización combinatoria más estudiados en las últimas décadas, debido principalmente a su relevancia para la industria. Este consiste en determinar un conjunto de rutas para una flota de vehículos que parte de uno o más depósitos para satisfacer la demanda de clientes dispersos geográficamente (Sarmiento, 2014). Por supuesto, existen un gran número de restricciones que pueden considerarse además de satisfacer la demanda (Toth & Vigo, 2002):

- Cada vehículo tiene una capacidad limitada (VRP capacitado).
- Cada cliente debe ser visitado en una determinada franja horaria (VRP con ventanas de tiempo).
- Varios puntos de suministro (VRP con múltiples depósitos).
- Los clientes pueden ser atendidos por varios vehículos (VRP con suministro dividido).
- Algunas variables del problema son aleatorias (VRP estocástico).
- Las entregas se deben realizar en determinados días (VRP periódico).

Típicamente, este problema se aborda concentrándose en un solo objetivo, comúnmente minimizar la distancia recorrida o *costo* de las rutas, minimizar la cantidad de vehículos utilizados para satisfacer a todos los clientes, o minimizar el tiempo total de transportación (García, 2010; Toth & Vigo, 2002).

## Ideas para Formulación del Problema de Ruteo de Vehículos

La red vial se describe generalmente con un grafo, los arcos representan secciones o tramos viales y los vértices corresponden a los clientes. Cada arco tiene asociado un costo que representa la longitud, tiempo de viaje, o alguna función de estos (García, 2010).

Para resolver este problema de forma exacta existen, de forma general, tres enfoques:

- *Formulación con flujo de vehículos*: Utiliza variables enteras asociadas con cada arco que cuentan el número de veces que un vehículo lo atraviesa. Generalmente, se utiliza para VRP básicos. Esto es muy funcional para los casos en que el costo de la solución se puede expresar como la suma de los costos asociados con los arcos (Toth & Vigo, 2002).
- *Formulación con flujo de mercancía*: Utiliza variables enteras adicionales asociadas con los arcos o aristas que representan el flujo de mercancías a lo largo de los caminos recorridos por los vehículos (Toth & Vigo, 2002).
- *Formulación como problema de partición*: Estos tienen un número exponencial de variables binarias, cada una de las cuales está asociada al uso (o no uso) de un recorrido factible diferente. Así, el VRP se formula como un problema de partición de conjuntos que busca cuál es el conjunto de recorridos con costo mínimo que satisfacen las restricciones del VRP (Toth & Vigo, 2002).

### 1.3.4 Modelos para Resolver el Problema de Ruteo de Vehículos (RVP)

A continuación se presentan las formulaciones de algunos modelos utilizados para estudiar este tipo de problemas.

#### Modelo con Cota Inferior de Vehículos (Larraín, 2021; Toth & Vigo, 2002)

Basado en la formulación del Problema del Agente Viajero de Dantzig, Fulkerson y Johnson. Sea  $G = (V, A)$  un grafo dirigido y completo, con costos  $c_a$ . El conjunto  $N = 1, 2, \dots, n$  representa a los clientes, de demanda  $q_i$ . El nodo cero (no incluido en  $N$ ), representa el depósito, donde se cuenta con  $K$  vehículos de capacidad  $Q$  para visitar a los clientes en  $N$ . Se define  $V := N \cup \{0\}$  y  $A$  el conjunto de arcos que unen los elementos en  $V$ . Finalmente, se definen las variables binarias  $x_a$  que indican si se utiliza el arco  $a$  en la solución. La función objetivo a minimizar sería la sumatoria de los productos del costo de los arcos y la binaria que determina si se utilizan o no. Planteado como un problema de programación entera:

$$\begin{aligned} & \text{Minimizar} \sum_{(i,j) \in A} c_{ij} x_{ij} & (3) \\ & \text{s.a.} \sum_{j \in \delta^+(i)} x_{ij} = 1, \quad \forall i \in N \\ & \sum_{i \in \delta^-(j)} x_{ij} = 1, \quad \forall j \in N \\ & \sum_{j \in \delta^+(0)} x_{0j} = K \\ & \sum_{(i,j) \in \delta^+(S)} x_{ij} \geq r(S), \quad \forall S \subseteq N, S \neq \emptyset \\ & x_{ij} \in \{0, 1\}, \quad \forall (i, j) \in A \end{aligned}$$

Donde  $\delta^+(i)$  representa los arcos que salen del nodo  $i$  en  $G$ ; y análogamente  $\delta^-(i)$  representa los arcos que se dirigen hacia el nodo  $i$ .  $r(S)$  es una función que devuelve una cota inferior correspondiente a la cantidad mínima de vehículos necesarios para satisfacer la demanda de los nodos en el subconjunto  $S$ . Por ejemplo, una cota aceptable sería:

$$r(S) = \frac{\lceil \sum_{i \in S} q_i \rceil}{Q}$$

Esta cota es bastante intuitiva: techo del cociente entre la suma de la demanda en un subconjunto de nodos  $S$  y la capacidad de los vehículos, esto nos da una mínima (aunque seguramente no viable) para la cantidad de vehículos necesarios para satisfacer la demanda del subconjunto  $S$ .

Las primeras dos restricciones especifican que para cada nodo (excluyendo el depósito) debe haber exactamente un arco de entrada y un arco de salida. Es importante notar que estas restricciones implican que la demanda de un nodo no puede ser compartida por más de dos vehículos.

La tercera restricción especifica que del nodo cero (el depósito) deben salir exactamente  $K$  arcos, uno por cada vehículo. La cuarta restricción exige que los arcos que salen (dirigidos hacia un nodo externo) de cualquier subconjunto de nodos  $S \subseteq N$ , lo que es equivalente a la cantidad de vehículos que se propone utilizar para satisfacer la demanda de los nodos en  $S$ , sea mayor o igual a una cota inferior  $r(S)$ . Esta restricción se encarga de evitar los *sub-tours* (recorridos que no incluyen al depósito, por lo tanto, son inválidos). Y finalmente, la quinta restricción define la naturaleza binaria de  $x_{ij}$ .

### Modelo con Restricciones MTZ (Flórez et al., 2017; Larraín, 2021)

Esta formulación basa sus restricciones en el trabajo de Christofides, Mingozzi y Toth. El planteamiento es casi idéntico al modelo anterior, pero cambia la lógica para evitar *sub-tours*. Sea  $G = (V, A)$  un grafo dirigido y completo, con costos  $c_a$ . El conjunto  $N = 1, 2, \dots, n$  representa a los clientes, de demanda  $q_i$ . El nodo cero (no incluido en  $N$ ), representa el depósito, donde se cuenta con  $K$  vehículos de capacidad  $Q$  para visitar a los clientes en  $N$ . Se define  $V := N \cup \{0\}$  y  $A$  el conjunto de arcos que unen los elementos en  $V$ . Se añaden las variables  $q_i$ , que representan la carga total que el vehículo repartidor ha repartido hasta el momento de abandonar el nodo  $i$ . Finalmente, se definen las variables binarias  $x_a$  que indican si se utiliza el arco  $a$  en la solución. Planteado como un problema de programación entera:

$$\begin{aligned} & \text{Minimizar} \sum_{(i,j) \in A} c_{ij} x_{ij} & (4) \\ & \text{s.a.} \sum_{j \in \delta^+(i)} x_{ij} = 1, \quad \forall i \in N \\ & \sum_{i \in \delta^-(j)} x_{ij} = 1, \quad \forall j \in N \\ & \sum_{j \in \delta^+(0)} x_{0j} = K \\ & u_i - u_j + Qx_{ij} \leq Q - q_j, \quad \forall (i,j) \in A(N) \\ & q_i \leq u_i \leq Q, \quad \forall i \in N \\ & x_{ij} \in \{0, 1\}, \quad \forall (i,j) \in A \end{aligned}$$

Donde  $\delta^+(i)$  representa los arcos que salen del nodo  $i$  en  $G$ ; y análogamente  $\delta^-(i)$  representa los arcos que se dirigen hacia el nodo  $i$ .  $A(N)$  representa el conjunto de arcos que unen los nodos en  $N$ .

Casi todas las restricciones se comparten con el modelo anterior, excepto la cuarta y quinta, que se encargan de evitar *sub-tours*, y vale la pena explicar la lógica detrás. Nótese que la cuarta restricción puede subdividirse en dos casos según si  $x_{ij}$  es 0 o 1. Estos son:

$$\begin{aligned} x_{ij} = 1 & \implies u_j \geq u_i + q_j \\ x_{ij} = 0 & \implies u_i - u_j \leq Q - q_j \end{aligned}$$

En el primer caso, la restricción exige que, para un arco  $(i, j)$ , la carga repartida acumulada hasta el nodo  $j$  sea mayor o igual a la carga repartida acumulada hasta el nodo  $i$  visitado justo antes más la demanda del nodo  $j$ . En el caso contrario, la restricción se cumple siempre, volviéndose redundante (a propósito), pues  $u_i - u_j \leq 0$  y  $Q - q_j \geq 0$  (recordemos que en este modelo los vehículos no comparten nodos).

La ventaja de utilizar este modelo respecto al de cota inferior es que la restricción de *sub-tours* de este último tiene cardinalidad exponencial respecto al número de nodos; mientras que con el modelo MTZ la cardinalidad es polinomial, ahorrando tiempo de computación (García, 2010).

### Modelo de Partición de Conjuntos (García, 2010; Toth & Vigo, 2002)

Este modelo fue originalmente propuesto por Balinski y Quandt (Balinski & Quandt, 1964). Retomando parte de los planteamientos anteriores, sea  $\mathcal{R} = \{R_1, R_2, R_3, \dots, R_s\}$  el conjunto de todas las rutas o circuitos factibles en  $G$ , a cada ruta se le asigna un costo  $\gamma_j$ . Se definen variables binarias  $a_{ij}$ , que vale 1 solo si el nodo  $i$  es visitado (o cubierto) por la ruta  $R_j$ . También se definen las variables binarias  $x_j$ , que valen 1 solo si la ruta  $R_j$  se utiliza en la solución óptima. Se plantea como problema de programación entera de la siguiente forma:

$$\begin{aligned} & \text{Minimizar } \sum_{j=1}^s \gamma_j x_j \tag{5} \\ & \text{s.a. } \sum_{j=1}^s a_{ij} x_j = 1, \quad \forall i \in N \\ & \quad \sum_{j=1}^s x_j = k \\ & \quad x_j \in \{0, 1\}, \quad \forall j \in \{1, 2, \dots, s\} \\ & \quad a_{ij} \in \{0, 1\}, \quad \forall i \in V, \quad j \in \{1, 2, \dots, s\} \end{aligned}$$

La primera restricción exige que cada cliente sea visitado exactamente por uno de los circuitos de  $\mathcal{R}$ . La segunda restricción exige que se elijan exactamente  $k$  circuitos de  $\mathcal{R}$ . Esta formulación es muy general, pero fácilmente moldeable, aunque tiene la desventaja de que requiere brindar de antemano el conjunto de crecimiento exponencial  $\mathcal{R}$ .

## 2 Definición del Problema

Se busca, a través de un modelo matemático y/o un algoritmo heurístico, proveer al personal de la Comisión Nacional Forestal (CONAFOR) un método para planificar de manera óptima las rutas necesarias para cumplir con la asignación de especies y la cantidad de individuos que será necesaria para cada sección de las hectáreas (Figura 2). Dado el tipo de problema es posible crear este plan de a través de programación lineal, planteándolo como un *problema de ruteo de vehículos*, representando los polígonos y la base de operaciones como nodos unidos por arcos con un costo determinado.

## 2.1 Conjuntos

- $N$  es el conjunto que representa a los polígonos (nodos demandantes).
- $V$  es el conjunto  $N \cup 0$  donde el elemento 0 representa el deposito o base.
- $A$  es el conjunto de arcos que unen a los elementos en  $V$ .
- $K$  es el conjunto de vehículos (o viajes que pueden hacerse).

## 2.2 Parámetros

- Se tiene un horario de trabajo de 8 horas por día.
- La camioneta tiene capacidad para 524 plantas, divididas en determinadas proporciones (véase Figura 2). Esta medida es equivalente a 1 *hectárea de plantas*.
- El tiempo que toma cargar y descargar completamente a una camioneta es de 30 minutos.
- Se tienen 30 polígonos demandantes con una superficie de 182.99 hectáreas en total.
- Se considera una velocidad para las camionetas de 20 kilómetros por hora.
- El gasto por kilómetro de gasolina es de 0.1 litros de gasolina.
- $c_{ij}$  es la distancia en km entre el nodo  $i$  y el nodo  $j$ .
- $Q$  la capacidad de la camioneta.
- $d_{ij}$  la distancia entre el polígono  $i$  y el  $j$ .

## 2.3 Variables

- Sea  $x_{ijk}$  una binaria que solo toma el valor 1 si el vehículo  $k$  viaja a través del arco  $(i, j)$ .
- Sea  $w_{ijk}$  la cantidad de recurso transportado por el vehículo  $k$  a través del arco  $(i, j)$ .
- Sea  $u_{ik}$  un número ordinal correspondiente a la posición cronológica que toma el nodo  $i$  en el recorrido del vehículo  $k$ , tomando el valor 0 si no utiliza dicho nodo.

No.	Especie	% de Reforestación	No. De Plantas por Ha.
<b>Especies Rosetófilas Crasifolias (48.85%)</b>			
1	Agave lechuguilla	6.267709924	33
2	Agave salmiana	29.96183206	157
3	Agave scabra	6.297709924	33
4	Agave striata	6.297709924	33
<b>Especies Crasicaules (34%)</b>			
5	Cylindropuntia imbricata	3.816793893	20
6	Opuntia cantabrigiensis	4.198473282	22
7	Opuntia engelmannii	3.816793893	20
8	Opuntia leucotricha	4.961832061	26
9	Opuntia robusta	8.58778626	45
10	Opuntia streptacantha	8.58778626	45
<b>Especies Leñosas (13.16)</b>			
11	Prosopis laevigata	13.16793893	69
<b>Especies Rosetófilas Arborescentes (4%)</b>			
12	Yucca filifera	4.007633588	21
Total		100	524

Figure 2: Tabla de Plantas requeridas por hectárea.



### 3 Metodología

Es posible modelar el proceso de reforestación mediante la implementación de un grafo  $G = (V, A)$  dirigido conformado por vértices y aristas, las vértices representando los puntos de descarga y las aristas representando la distancia entre cada uno de ellos.

Tras un breve análisis, se puede observar que el proyecto tiene muchas similitudes con los modelos VRP (Vehicule Routing Problem), específicamente VRP con entregas divididas. Estos problemas se caracterizan por tener un nodo origen, el cual suministra los recursos a todos los demás nodos, los cuales pueden ser satisfechos en varias entregas. Desde el nodo de origen parten  $k$  vehículos, los cuales pueden tener capacidades de carga idénticas o diferentes (Dror et al., 1994).

Para este modelo, dada la escasa información sobre los caminos con los que se cuenta, se decidió implementar la distancia euclidiana como el costo de tomar cada arista. Estos costos son representados por el parámetro  $c(i,j)$ . Por otro lado, como no se tiene conocimiento de la cantidad de vehículos con los que se cuenta, se decidió fijar el número de vehículos como la mínima cantidad de ciclos que se necesitan para suplir todas las demandas, definiendo a un ciclo como aquel que empieza saliendo del nodo origen y termina llegando a este mismo.

Para obtener las distancias entre los nodos se hizo uso de la imagen de los polígonos suministrado por el socio formador, se utilizó la escala en metros que se presenta en la imagen y se implementó un código en Python que nos permitió obtener la distancia entre cada nodo tan solo ingresando la coordenada aproximada del centro de cada uno.

Durante el proceso de modelación, nos dimos cuenta de que, si asumimos que todos los camiones utilizados tienen la misma capacidad, muchos de los nodos necesitan de uno o más camiones completamente llenos para satisfacerse, es decir, es necesario hacer múltiples rutas en donde el camión sale del nodo de origen, llega al nodo a suministrar, descarga toda la carga que lleva y regresa al nodo de origen. Tras esto, nos percatamos que la mayoría de los ciclos de entrega pueden ser fácilmente determinados tan solo redondeando la demanda al entero menor inmediato. Por ejemplo, si un nodo tiene una demanda de 6.28 y la capacidad del camión es 1, el modelo llevará siempre 6 camiones completamente llenos, dejando al nodo con únicamente la parte decimal por suplir. El orden en el que se hacen estas rutas no cambia el costo y calcular este último no requiere de gran capacidad computacional.

Es posible observar que el problema se reduce entonces a determinar las rutas a seguir para suplir las partes decimales de las demandas, reduciendo así la complejidad del problema. Para este modelo simplificado se puede observar que el número de nodos se reduce a 26, pues hay 5 que tienen demandas enteras.

Dado que lo que se pretende optimizar es el tiempo que tardan los vehículos en suministrar las demandas solicitadas, es necesario considerar la velocidad a la que estos camiones se mueven. Al ser camiones de carga pesada, estos tienden a tener una aceleración y velocidad baja, principalmente cuando se trata de caminos no pavimentados, como es en este caso de estudio. Consideramos una velocidad promedio de 20km/h, pues no existen caminos establecidos en la mayoría de los trayectos y la zona de reforestación tiene una altitud uniforme.

Para obtener una solución óptima se propone primeramente un modelo matemático de programación lineal mixta (Ver Sección 4). Este modelo fue ejecutado en el software de modelación GAMS y en Python con ayuda de la librería PuLP. Sin embargo, el modelo propuesto presenta una complejidad excesiva para la licencia con la que contamos en GAMS, mientras que en Python, el modelo supera el límite de tiempo.

Debido a esto, se propone una solución heurística (Ver Sección 5), la cual, mediante código en Python, permite obtener una solución muy cercana a la óptima, utilizando bastante menos memoria y tiempo de ejecución.

## 4 Modelo Matemático

### 4.1 Definiciones

#### 4.1.1 Conjuntos

El conjunto  $N = \{1, 2, \dots, n\}$  representa a los clientes. Se define  $V = N \cup \{0\}$ , donde 0 representa al depósito.  $A$  es el conjunto de arcos que unen los elementos en  $V$ .  $A'$  es el conjunto de arcos que unen a los elementos de  $N$ . Con lo anterior,  $G = (V, A)$  es un grafo dirigido y completo de nodos  $V$  y arcos  $A$ . Se define  $K = \{1, 2, \dots, m\}$ , representando a los vehículos (o a los recorridos distintos que deben hacerse).

#### 4.1.2 Parámetros

Sea  $n \in \mathbb{N}_0$  la cantidad de clientes, cada uno de ellos con una demanda de  $q_i : i \in N$ . Sea  $m \in \mathbb{N}$  la cantidad máxima de vehículos (o recorridos independientes) disponibles, todos ellos de capacidad  $L \in \mathbb{R}^+$ . Nótese que puede ponerse una cota inferior para  $m \geq (\sum q_i) \div Q$ . El grafo  $G$  tiene costos por unidad  $c_a : a \in A$ . Sea  $\mathcal{M}$  una constante suficientemente grande, para relacionar variables continuas y binarias en restricciones, que es al menos igual la demanda máxima. Sea  $v$  una constante correspondiente a la velocidad media de los vehículos en metros/hora, y  $t$  una constante correspondiente al tiempo de carga o descarga medido en horas.

#### 4.1.3 Variables

Sea  $x_{ijk}$  una binaria que toma valor 1 si el vehículo  $k$  utiliza el arco  $(i, j)$  en su ruta, o toma valor 0 si no lo utiliza. En la misma línea,  $w_{ijk} \geq 0$  representa la cantidad de material que transporta el vehículo  $k$ , saliendo del nodo  $i$  y a entregar en el nodo  $j$ . Finalmente  $u_{ik} \in \mathbb{Z}^+$  toma un valor correspondiente al orden en el que el nodo  $i$  es visitado por el vehículo  $k$ .

### 4.2 Problema de Programación Lineal

Planteado como un problema de programación lineal, puede escribirse de la siguiente forma:

$$\text{Minimizar} \quad v^{-1} \sum_{i \in V} \sum_{j \in V} \sum_{k \in K} c_{ij} x_{ijk} + 2t \sum_{i \in V} \sum_{j \in V} \sum_{k \in K} w_{ijk} \quad : i \neq j \quad (6)$$

$$\text{Sujeto a} \quad \sum_{j \in V} x_{ijk} - \sum_{j \in V} x_{jik} = 0 \quad : i \neq j \quad \forall i \in V, \quad k \in K$$

$$\sum_{j \in N} x_{0jk} = 1 \quad \forall k \in K$$

$$\sum_{i \in V} \sum_{j \in V} w_{ijk} \leq L \quad : i \neq j \quad \forall k \in K$$

$$\sum_{i \in V} \sum_{k \in K} w_{ijk} \geq q_j \quad : i \neq j \quad \forall j \in N$$

$$u_{ik} - u_{jk} + n x_{ijk} \leq n - 1 \quad : i \neq j \quad \forall i, j \in N, \quad k \in K$$

$$w_{ijk} \leq \mathcal{M} x_{ijk} \quad \forall i, j \in V, \quad k \in K$$

$$x_{ijk} \in \{0, 1\} \quad \forall i, j \in V, \quad k \in K$$

$$w_{ijk} \geq 0 \quad \forall i, j \in V, k \in K$$

La función objetivo a minimizar es la sumatoria del producto de los costos por arco y la binaria que indica si se utilizó o no, multiplicado por el recíproco de la velocidad, todo esto representando el tiempo invertido en recorrer distancias; más la suma de todas las cargas entregadas multiplicadas por  $2t$ , que es el tiempo de carga/descarga, que se asume varía linealmente según la cantidad de plantas a descargar. La primera restricción es para conservar el flujo de vehículos. La segunda restricción se asegura de que todos los autos pasen por el nodo cero (depósito). La tercera restricción corresponde a la carga máxima de cada vehículo. La cuarta restricción es para satisfacer las demandas de cada cliente. La siguiente restricción se utiliza para evitar subtours, asignando a cada nodo un entero positivo correspondiente al orden en el que son visitados por cada vehículo. Las últimas tres restricciones son sobre la naturaleza de las variables:  $x_{ijk}$  debe valer cero solo si  $w_{ijk}$  es igual a 0, y debe valer 1 en cualquier otro caso.  $x_{ijk}$  es una variable binaria. Por su parte,  $w_{ijk}$  debe ser mayor o igual que cero para evitar transportar una cantidad negativa de material (plantas).

## 5 Método Heurístico

Para el método heurístico se optó por utilizar un algoritmo heurístico greedy, el cual empieza supliendo completamente la demanda del nodo más lejano al depósito, para después descargar el recurso restante en el nodo más cercano al visitado anteriormente. Esto hace que las entregas sean más eficientes, pues al haber visitado y suplido completamente el nodo más lejano, sin importar hacia donde te muevas, siempre te acercas más a la base. Debido a que los clientes más alejados son los que generan más costos, este algoritmo nos permite reducir las veces que es necesario viajar a un nodo lejano.

### 5.1 Resultados Preliminares

Como se explicó anteriormente, el problema reducido consiste en satisfacer aquellos polígonos donde la demanda es no entera, por lo que la solución implementada con el método heurístico se centró únicamente en estos nodos, en total 26, contando al nodo que representa a la base (demanda 0).

En total, obtuvimos 15 rutas diferentes necesarias para satisfacer la demanda de la parte decimal de los nodos, mostradas en la Tabla 4. Asimismo, se muestra de manera gráfica dos de las rutas del modelo planteado (ver Figura 3). El resto de las rutas pueden ser observadas en los anexos.

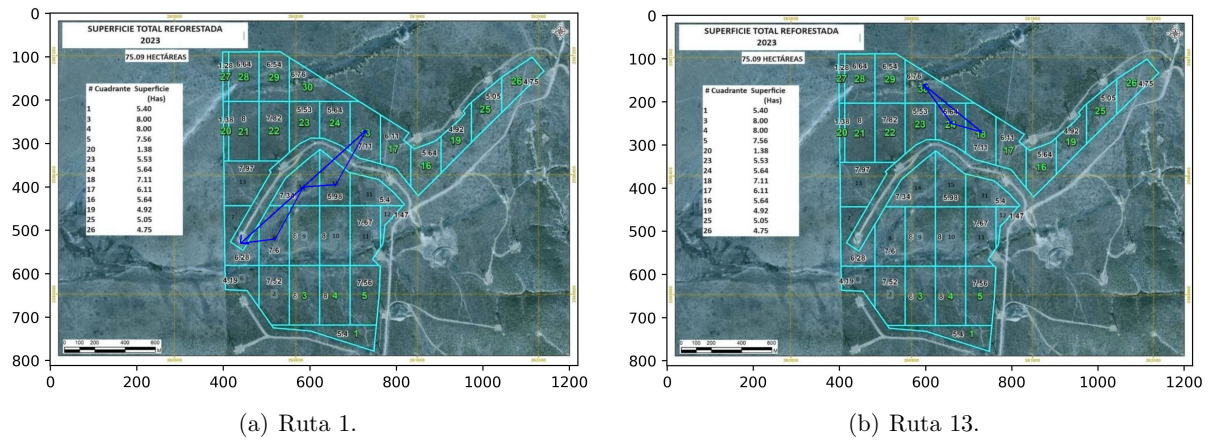


Figure 3: Algunas rutas que forman parte de la solución heurística.

El costo total de las rutas de la parte decimal es de 15.001 horas. Mientras que el costo total en horas de la parte entera de las rutas es de 182 horas. Sumando así un total de 197 horas, o 24.7 jornadas laborales.

A partir de las rutas en la tabla 1 se realizó una distribución de rutas, la cual considera que una vez visitado un nodo, este tiene que suplir su parte entera lo más pronto posibles, así como no se debe pasar encima de nodos que se hayan suplido totalmente. Estas rutas fueron planeadas con una jornada laboral de 8 horas, así como el uso solamente de una camioneta. Una vez ordenadas, el algoritmo tomó 26 jornadas laborales (Tabla 5), una más de las mínimas, tomando como referencia una cota inferior.

## 6 Experimentación y Resultados

La experimentación tanto para el modelo matemático como para el metodo heuristico se llevó a cabo en un computadora con las siguientes características:

- **Modelo de la PC:** Dell G15 5520
- **Sistema operativo:** Windows 11.
- **Capacidad de disco duro:** 512GB
- **Memoria Ram:** 16GB.
- **Tipo de procesador:** 12th Gen Intel(R) Core(TM)
- **Numero de nucleos:** 7
- **Software utilizado y versión:** Para el modelo matemático se usó la librería PuLP 2.8.0 de Python en combinación con la licencia de CPLEX IBM. Para el metodo heuristico se hizo uso de Pandas 2.2 en Python.

### 6.1 Tamaño del problema

#### 6.1.1 Nodos y Arcos

El problema original, en su parte decimal, cuenta con 26 nodos (excluyendo 5 nodos cuya parte decimal es cero), obteniendo a su vez 676 arcos.

#### 6.1.2 Variables

El modelo suma un total de 20670 variables de decisión, tomando en cuenta tres grupos de variables:

- $x_{ijk}$  una variable binaria que toma el valor de 1 si el vehículo  $k$  utiliza el arco  $(i, j)$  en su ruta y 0 si no lo utiliza. Ya que se consideran variables distintas para cada uno de los 15 vehículos a utilizar, existen 10140 variables de este tipo.
- $w_{ijk}$  representa la cantidad de material que transporta el vehículo  $k$  para el arco  $(i, j)$ . Ya que se consideran variables distintas para cada uno de los 15 vehículos a utilizar, existen 10140 variables de este tipo.
- $u_{ik} \in Z^+$  toma un valor correspondiente al orden en el que el nodo  $i$  es visitado por el vehículo  $k$ . Ya que existen 26 nodos y 15 vehículos a utilizar, existen 390 variables de este tipo.

### 6.1.3 Parámetros

El modelo cuenta con un total de 707 parámetros, que son parte de la siguiente lista:

- $c_{ij}$  siendo el costo de viajar del nodo  $i$  al nodo  $j$ , resulta en 676 parámetros, uno por arco.
- $q_i$  como la demanda del nodo  $i$ , resulta en 26 parámetros.
- $m$  como la mínima cantidad de rutas requeridas.
- $L$  siendo la capacidad de los vehículos.
- $t$  el tiempo que toma cargar o descargar.
- $M$  una constante de gran tamaño.
- $V$  la velocidad promedio de los vehículos.

## 6.2 Resultados

### 6.2.1 Problemas Reducidos

Para comparar los resultados entre el modelo matemático y el método heurístico con el fin de determinar si la cercanía de la solución heurística a la solución óptima es aceptable (menor al 1%), se usaron tres tamaños de problema que el modelo matemático es capaz de resolver en tiempos razonables, probando en grupos de cinco muestras tomadas aleatoriamente: con 5 nodos (Tabla 1), otro con 10 nodos (Tabla 2), y un ultimo con 12 nodos (Tabla 3). Para garantizar la igualdad de condiciones entre las muestras, se asegura que no contienen otros nodos de demanda cero además de la base. Asimismo, se incluyen gráfica que permiten visualizar la diferencia de los tiempos de procesamiento de los 3 tamaños tanto para el modelo matemático como para el heurístico (Figura 4).

#	Muestra de Nodos	Modelo Matemático		Método Heurístico		Parte Entera	Distancia Porcentual
		Solución Óptima	Tiempo de Computo	Solución	Tiempo de Computo		
1	{1, 18, 20, 23, 26}	2.33	0.05 segundos	2.483	0.004 segundos	16.433	0.81%
2	{1, 7, 11, 18, 20}	1.97	0.5 segundos	2.112	0.007 segundos	21.245	0.61%
3	{1, 2, 8, 16, 18}	2.40	0.6 segundos	2.547	0.003 segundos	26.481	0.50%
4	{5, 17, 18, 19, 29}	2.33	0.1 segundos	2.352	0.003 segundos	24.83	0.08%
5	{6, 7, 13, 18, 30}	1.41	0.1 segundos	2.414	0.006 segundos	25.094	3.78%

Table 1: Resultados para casos de prueba con cinco nodos.

### 6.2.2 Problema Completo

Dado el tamaño de este problema, no fue posible resolverlo con el modelo matemático en un tiempo razonable (menor que 48 horas), por ello presentamos la solución obtenida mediante el método

#	Muestra de Nodos	Modelo Matemático		Método Heurístico		Parte Entera	Distancia Porcentual
		Solución Óptima	Tiempo de Computo	Solución	Tiempo de Computo		
1	{2, 8, 13, 16, 17, 18, 20, 22, 28, 29}	5.67	3 minutos 53 segundos	5.73	0.032 segundos	56.021	0.09%
2	{5, 8, 12, 16, 18, 19, 20, 22, 28, 29}	6	15 segundos	6.049	0.005 segundos	47.488	0.09%
3	{2, 5, 6, 7, 11, 17, 18, 20, 22, 30}	4.69	5.8 segundos	4.734	0.036 segundos	55.326	0.07%
4	{2, 8, 11, 18, 19, 20, 23, 26, 29, 30}	6.14	3.9 segundos	6.203	0.031 segundos	50.758	0.11%
5	{2, 7, 11, 18, 19, 20, 23, 26, 29, 30}	5.81	4.7 segundos	5.864	0.012 segundos	49.773	0.09%

Table 2: Resultados para casos de prueba con diez nodos.

#	Muestra de Nodos	Modelo Matemático		Método Heurístico		Parte Entera	Distancia Porcentual
		Solución Óptima	Tiempo de Computo	Solución	Tiempo de Computo		
1	{1, 2, 6, 8, 11, 15, 18, 19, 20, 22, 23, 29}	7.07	1 minutos 3 segundos	7.237	0.027 segundos	62.879	0.23%
2	{1, 2, 5, 6, 8, 15, 18, 20, 22, 25, 27, 30}	6.07	7 minutos 5 segundos	6.763	0.01 segundos	60.104	1.04%
3	{1, 5, 6, 7, 8, 11, 14, 17, 18, 19, 22, 29}	6.04	17 minutos 15 segundos	6.135	0.036 segundos	71.469	0.11%
4	{2, 3, 7, 9, 11, 13, 18, 22, 25, 29, 30}	6.01	26 segundos	6.107	0.01 segundos	72.862	0.12%
5	{2, 5, 6, 7, 11, 14, 15, 18, 24, 29, 30}	6.54	61 minutos	6.591	0.013 segundos	72.448	0.06%

Table 3: Resultados para casos de prueba con doce nodos.

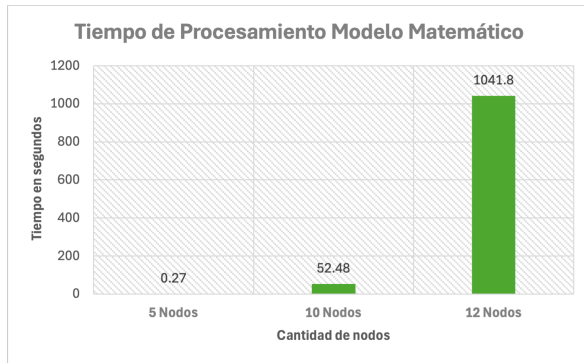
heurístico. Para la parte entera se tiene que serán necesarias 182.68 horas, mientras que para la parte decimal se requieren 15.18 horas. Por lo que el resultado de la función objetivo es:

$$z^* = 197.86 \text{ horas}$$

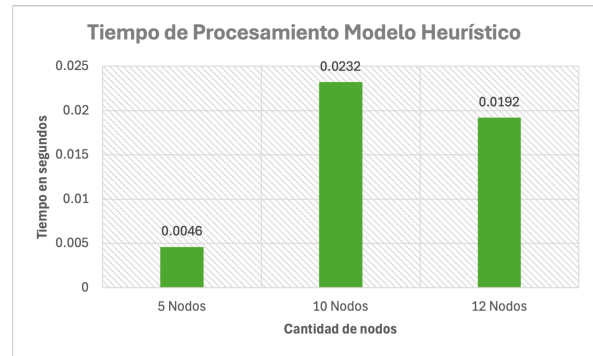
Equivalente a 25 jornadas laborales de 8 horas, si se considera un único vehículo, pero este tiempo puede reducirse si varios vehículos operan de forma paralela. Esta solución requiere 183 viajes, justamente los mínimos estimados a través de la cota inferior  $\lceil \sum_i q_i \div L \rceil = 183$ , prueba de que esta solución es factible.

## 7 Conclusiones

Dados estos resultados, se puede observar que el modelo matemático proporciona mejores resultados, pero el porcentaje de mejoría no es suficientemente alto como para asumir el costo computacional que requiere en problemas grandes. Es por esto que el resultado del problema completo solo se pudo



(a) Modelo matemático.



(b) Metodo heurístico

Figure 4: Tiempos de procesamiento para distintos tamaños de problema.

llevar a cabo mediante el método heurístico. De igual manera, el bajo tiempo computacional que requiere este método, permite ajustes rápidamente en los parámetros y función objetivo, pudiendo obtener soluciones a instancias distintas del problema sin requerir un alto tiempo de espera para obtener una respuesta.

Table 4: Plan de Trabajo para la Parte Decimal de la Demanda

#	Ruta
1	base, 1, 5, 11, base
2	base, 6, 2, 8, base
3	base, 7, 8, 14, 15 base
4	base, 26, 25, 19, base
5	base, 27, 28, 29, base
6	base, 20, 22, base
7	base, 13, 22, base
8	base, 29, 30, base
9	base, 11, 12, base
10	base, 19, 16, base
11	base, 22, 23, 24, base
12	base, 12, 31, 15, base
13	base, 30, 24, base
14	base, 16, 17, base
15	base, 15, base

El costo total de las rutas de la parte decimal es de 15.001 horas. Mientras que el costo total en horas de la parte entera de las rutas es de 182 horas. Sumando así un total de 197 horas, o 24.7 jornadas laborales.

A partir de las rutas mostradas en la Tabla 4 se realizó una distribución de rutas, la cual considera que una vez visitado un nodo, este tiene que suplir su parte entera lo más pronto posible, así como no se debe pasar encima (atravesar geográficamente) de nodos que se hayan suplido totalmente, pues estos podrían estar ya plantados. Estas rutas fueron planeadas con una jornada laboral de 8 horas, así como el uso solamente de una camioneta. Una vez ordenadas, el algoritmo provee una solución que toma 26 jornadas laborales (Vea Anexos Tabla 5) para satisfacer a los 30 polígonos demandantes.



## Referencias

- Aquae. (2023). *La encina, el árbol más popular del quijote*. Aquae Fundación: La Fundación de Agua. <https://www.fundacionaquae.org/wiki/encina/> Consultado el 18/5/2024.
- Balam. (2022). *El marco de plantación: Qué es y tipos*. Balam Agriculture. <https://balam.es/el-marco-de-plantacion/> Consultado el 18/5/2024.
- Balinski, M. L., & Quandt, R. E. (1964). On an integer program for a delivery problem. *Operations research*, 12(2), 300–304.
- Carbo, A. (1978). *Marqueo de plantaciones: Tresbolillo*. Hojas Divulgadoras. <https://www.mapa.gob.es/ministerio/pags/biblioteca/hojas/hd.1978.21.pdf> Consultado el 18/5/2024.
- Comisión Nacional Forestal. (2022). *Se Reduce en 26% la Tasa Anual de Deforestación*. <https://www.gob.mx/conafor/prensa/se-reduce-en-26-la-tasa-anual-de-deforestacion> Consultado el 18/5/2024.
- Dror, M., Laporte, G., & Trudeau, P. (1994). Vehicle routing with split deliveries. *Discrete Applied Mathematics*, 50(3), 239–254. [https://doi.org/https://doi.org/10.1016/0166-218X\(92\)00172-I](https://doi.org/https://doi.org/10.1016/0166-218X(92)00172-I)
- Flórez, J. A. V., Grisales, C. J. Z., & Gatica, G. (2017). Una aplicación del método mtz a la solución del problema del agente viajero. *Scientia et technica*, 22(4), 341–344. <https://www.redalyc.org/pdf/849/84955649006.pdf>
- García, I. (2010). *El problema de ruteo de vehículos*. Universidad Autónoma de Coahuila. <http://tikhonov.fcencias.unam.mx/presentaciones/2010ago05.pdf>
- Gobierno de Yucatán. (n.d.). *Ceiba*. Gobierno del Estado de Yucatán. <https://www.yucatan.gob.mx/?p=ceiba> Consultado el 18/5/2024.
- Guerrero, F. A. (2018). *Fórmula para plantaciones en tres bolillos*. Agricultura y Agronomía. <https://www.youtube.com/watch?v=NCNM0Yc7D3c&t=310s> Consultado el 18/5/2024.
- Hernández, F. (2011). *El altiplano mexicano o altiplanicie mexicana: Flora fauna del altiplano mexicano*. Expresiones Veterinarias. <https://www.expresionesveterinarias.com/2011/11/el-altiplano-mexicano-o-altiplanicie.html> Consultado el 18/5/2024.
- Iglesias, L. (2021). *Siembra a tresbolillo*. <https://matematicas11235813.luismiglesias.es/2021/06/05/siembra-a-tresbolillo-competencia-matematica-geometria-plana-aplicada-en-huertos-y-jardines-dia-mundial-del-medio-ambiente/> Consultado el 18/5/2024.
- Larraín, H. (2021). *Ruteo de vehículos 2.1: Formulación del vrp*. YouTube. <https://youtu.be/q2fUDDJMbpq?feature=shared> Consultado el 18/5/2024.
- Lifeder. (2020). *Altiplanicie mexicana: Características, flora, fauna, relieve: Flora de la altiplanicie mexicana*. <https://www.lifeder.com/altiplanicie-mexicana/> Consultado el 18/5/2024.
- Masats, J. (2022). *Características del pino*. Botanical Online. <https://www.botanical-online.com/botanica/pino-caracteristicas> Consultado el 18/5/2024.
- Múltiples autores. (2024). *Mezquite*. Wikipedia. <https://es.wikipedia.org/wiki/Mezquite> Consultado el 18/5/2024.
- Sarmiento, A. (2014). *Estudio del problema de ruteo de vehículos con balance de carga: Aplicación de la meta-heurística búsqueda tabú*. [Master's thesis, Universidad de la Sabana]. <https://intellectum.unisabana.edu.co/bitstream/handle/10818/9798/Ang%20Sarmiento%20Lepesqueur%20%28TESIS%29.pdf> Consultado el 18/5/2024.
- SEMARNAT. (2018). *Oyamel de Juárez, especie endémica y en peligro de extinción*. Gobierno de México. <https://www.gob.mx/semarnat/articulos/oyamel-de-juarez-especie-endemica-y-en-peligro-de-extincion?idiom=es> Consultado el 18/5/2024.
- Tips y Temas Agronómicos. (2016). *Siembra en triángulo o en tresbolillo*. Tips y Temas Agronómicos. <https://www.tipsytemasagronomicos.com/siembra-en-triangulo-o-en-tresbolillo/> Consultado el 18/5/2024.
- Toth, P., & Vigo, D. (2002). *The vehicle routing problem*. SIAM. ISBN: 0-89871-579-2.



## 8 Anexos

### 8.1 Plan de Trabajo Propuesto

Se incluyen detalles de cada viaje, así como el orden específico en el que debe realizarse para minimizar la cantidad de jornadas laborales requeridas y satisfacer exitosamente la demanda.

Table 5: Plan de Trabajo Completo para una Camioneta

Día	Horas Totales	Horas por Viaje	Carga para Dejar	Nodos a Visitar	Número de Viajes
1	1.000	1.000	0.40, 0.56, 0.04	1, 5, 11	1
1	5.700	1.140	1	1	5
2	7.790	1.113	1	5	7
3	7.800	1.114	1	4	7
4	1.100	1.100	1	4	1
4	6.670	1.112	1	3	6
5	2.220	1.110	1	3	2
5	5.367	1.073	1	10	5
6	3.220	1.073	1	10	3
6	4.320	1.080	1	9	4
7	4.320	1.080	1	9	4
7	1.000	1.000	0.19, 0.52, 0.29	6, 2, 8	1
7	2.254	1.127	1	2	2
8	5.636	1.127	1	2	5
8	2.186	1.093	1	8	2
9	5.464	1.093	1	8	5
9	2.260	1.130	1	6	2
10	2.260	1.130	1	6	2
10	1.000	1.000	0.28, 0.31, 0.34, 0.07	7, 8, 14, 15	1
10	4.440	1.110	1	7	4
11	2.222	1.111	1	7	2
11	5.267	1.053	1	14	5
11	2.107	1.053	1	14	2
12	1.000	1.000	0.75, 0.05, 0.2	26, 25, 19	1
12	4.436	1.109	1	26	4
12	2.169	1.084	1	25	2
13	3.253	1.084	1	25	3
13	4.250	1.063	1	19	4
14	1.000	1.000	0.72, 0.28	19, 16	1
14	5.230	1.046	1	16	5
14	1.000	1.000	0.36, 0.11	16, 17	1
15	3.060	1.020	1	17	3
15	3.060	1.020	1	17	3
15	1.000	1.000	0.63, 0.37	11, 12	1
16	4.280	1.070	1	11	4
16	3.220	1.070	1	11	3
17	1.060	1.060	1	12	1
17	1.000	1.000	0.10, 0.4, 0.5	12, 31, 15	1
17	5.215	1.043	1	31	5
18	1.000	1.000	0.28, 0.64, 0.08	27, 28, 29	1
18	1.100	1.100	1	27	1
18	5.450	1.090	1	28	5
19	1.090	1.090	1	28	1
19	6.400	1.067	1	29	6
20	1.000	1.000	0.46, 0.54	29, 30	1
20	6.290	1.048	1	30	6
21	1.000	1.000	0.22, 0.34	30, 24	1
21	5.104	1.021	1	24	5
21	1.000	1.000	0.17, 0.53, 0.3	22, 23, 24	1
22	4.163	1.041	1	23	4
22	1.041	1.041	1	23	1
22	1.000	1.000	0.38, 0.62	20, 22	1
22	1.090	1.090	1	20	1
23	5.400	1.080	1	21	5
23	2.160	1.080	1	21	2
24	7.400	1.057	1	22	7
25	7.609	1.087	1	13	7
26	1.000	1.000	0.97, 0.03	13, 22	1
26	5.204	1.041	1	15	5
26	1.000	1.000	0.41	15	1

## 8.2 Imágenes de las rutas

Se proporciona un enlace a una [carpeta de Google Drive](#). con las imágenes de las rutas mencionadas en nuestro plan de trabajo sobre el espacio geográfico, con fines ilustrativos.

## 8.3 Código de los Modelos

A continuación, se proporcionan enlaces a los códigos utilizados para la resolución del modelo matemático y la ejecución del método heurístico.

- [Modelo Matemático en GAMS](#).
- [Modelo Matemático en Python](#).
- [Modelo Heurístico en Python](#).