Instituto Tecnológico y de Estudios Superiores de Monterrey

MA2001B: Optimización Determinista (Gpo 301)

Reto: Optimización del Transporte en la Logística de reforestación

Integrantes:

Andrea Renata Garfias Núñez A01369860

Génesis Pereyra Camacho A01734276

Maritza Barrios Macías A00836821

Monterrey, N.L. a 9 de junio de 2024

Índice

1. Introducción

La deforestación es la pérdida permanente de la vegetación forestal por causas inducidas o naturales, implicando el cambio de uso de tierra forestal a cualquier otro uso de la tierra, como el agrícola, praderas, asentamientos humanos, humedales u otras tierras. En México, la deforestación ha aumentado significativamente en la última década, colocándose como el quinto país con mayor pérdida de bosques a nivel mundial según la FAO (2020), con más de 1.25 millones de hectáreas perdidas entre 2010 y 2020. Solo en 2020 se perdieron 127,770 hectáreas, equivalente a la extensión de la Sierra Madre Occidental. En 2022, se reportaron 206,564 hectáreas deforestadas según la Secretaría de Medio Ambiente y Recursos Naturales.

Este fenómeno tiene un impacto significativo en la crisis del agua y el cambio climático. Los bosques retienen humedad vital para los suelos, y los árboles son esenciales en el ciclo hidrológico devolviendo vapor de agua a la atmósfera. Sin ellos, los ecosistemas pueden transformarse en desiertos áridos, reduciendo la disponibilidad de agua y aumentando las emisiones de gases de efecto invernadero, acelerando el cambio climático (National Geographic, 2024).

La deforestacion afecta de manera económica debido a que debilita economías locales, amenaza medios de vida y culturas de poblaciones dependientes de los bosques, y causa conflictos por recursos naturales. Socialmente, afecta la supervivencia de comunidades, incluidos los pueblos indígenas, aumenta la vulnerabilidad ante desastres naturales y causa pérdida de biodiversidad, alteración de ciclos del agua, erosión del suelo y desertificación (Ruiz Villar, M, 2018).

Para revertir estos problemas, México ha implementado una estrategia coordinada por la Semarnat en colaboración con diversas dependencias y órdenes de gobierno, identificando 122 zonas críticas forestales en 20 entidades y diseñando ordenamientos territoriales para la conservación de los entornos naturales (Secretaría de Medio Ambiente y Recursos Naturales, 2021).

Para la reforestación, se seleccionan especies nativas adaptadas al clima local y con múltiples usos como néctar, polen, frutos, forraje, madera y sombra. Se prioriza la calidad de las plantas y semillas, verificando su estado libre de plagas y hongos, con pruebas de viabilidad y trazabilidad adecuadas. Además, se recomienda plantar en la estación adecuada y proporcionar cuidados iniciales como riego y protección contra condiciones adversas (Red de Viveros de Biodiversidad, s.f.).

Para distribuir las especies en áreas de reforestación, se identifican sitios con similaridad ambiental mediante recorridos en áreas cercanas con condiciones similares, considerando altitud, tipo de suelo, orientación del terreno, patrones de precipitación y temperatura, y conocimiento local de los pobladores. Se priorizan especies adaptadas al rango de altitud y tipo de suelo del área, y aquellas que se desarrollan bien en vegetación secundaria, evitando especies de vegetación primaria para sitios degradados, a menos que demuestren una amplia plasticidad genética (Criterios para elegir las especies adecuadas, s.f.).

Este texto proporciona una visión integral de la deforestación en México y las estrategias para su mitigación y restauración forestal, destacando la importancia de acciones coordinadas y criterios técnicos para la selección y distribución de especies en proyectos de reforestación.

2. Justificación

La reforestación se presenta como una estrategia esencial para mitigar estos efectos adversos. La planificación eficiente de proyectos de reforestación requiere la implementación de modelos matemáticos que optimicen la asignación y distribución de plantas.

El diseño de este modelo no solo busca optimizar la logística de reforestación, sino también contribuir a la recuperación de los ecosistemas afectados, promoviendo la regeneración natural y la sostenibilidad ambiental. La implementación efectiva de este modelo podría ser un paso crucial para revertir los efectos del cambio climático y preservar la biodiversidad en México, asegurando un futuro más sostenible y resiliente para las generaciones venideras.

Minimizar las distancias recorridas y los costos operativos es crucial para garantizar la viabilidad y sostenibilidad de un proyecto de reforestación. Al reducir las distancias, se disminuyen los costos de combustible y mantenimiento de los vehículos, además de reducir las emisiones de gases de efecto invernadero, lo que contribuye significativamente a la lucha contra el cambio climático. Adicionalmente, una planificación eficiente de las rutas de transporte asegura que las plantas se entreguen de manera oportuna y dentro de los límites operativos, optimizando el uso de los recursos disponibles y mejorando la eficiencia general del proyecto.

3. Objetivo

El objetivo del presente trabajo es desarrollar un modelo de optimización que facilite la planificación de la reforestación en el altiplano mexicano. Se espera que este modelo sea capaz de determinar adecuadamente las especies de plantas a las unidades de transportes y crear rutas de distribución que minimicen los tiempos de entrega y los costos opoerativos, asegurando una adecuada distribución de la flora de la zona.

Se trabajará con la Unidad de Servivios Profesionales Altamente Especializados (USPAE-

INECOL) y con datos de la Comisión Nacional Forestal (CONAFOR), las cuales se encargan de desarrollar planes que se dirigan al desarrollo forestal sustentable en México. Se colaborará con el fin de encontrar un modelo, con variables cuantificables, que se adecúe a sus necesidades, y de esta forma, asegurar que los recursos disponibles estén correctamente distribuidos.

4. Análisis Literatura

El altiplano mexicano, una amplia meseta caracterizada por su variedad de paisajes y climas, comprende los estados de Chihuahua, Coahuila de Zaragoza, Nuevo León, Durango, Zacatecas, San Luis Potosí, Aguascalientes, Guanajuato, Querétaro de Arteaga, Hidalgo, México, Tlaxcala, Puebla, Jalisco, Michoacán, el Distrito Federal y Zacatecas (Hernández, 2011).

El presente trabajo se enfoca en reforestar el parque Eólico Dominica LT de Enel Green Power dentro del altiplano mexicano, ubicado en el estado de San Luis Potosí.

Flora en el altiplano mexicano

Dentro de las zonas desérticas y cálidas del altiplano mexicano, se encuentran plantas como:

- **Agave lechuguilla:** Mejor conocida como "Lechugilla" se usa para obtener fibras (ixtle) y saponinas.
- **Agave salmiana:** Mejor conocido como "Maguey Verde" es valorado por su bebida fresca (aguamiel) y fermentada (mezcal).
- **Agave Agave scabra:** Conocido como "Maguey aspero, Azul" sirve como cerca viva.

- **Agave Agave striata:** Conocida como “Guapilla” es ornamental.
- **Agave Cylindropuntia imbricata:** se utiliza para utensilios, artesanías y forraje y es mejor conocida como “Cardenche”.
- **Agave Opuntia cantabrigiensis:** El “Cuijo” es utilizado para crear una cerca viva y forraje
- **Agave Opuntia engelmannii:** “Rastrero” es utiizado principalmente para forraje.
- **Agave Opuntia leucotricha:** “Duraznillo” cerca viva y frutal.
- **Agave Opuntia robusta:** “Tapón” sirven como forraje y frutal.
- **Agave Opuntia streptacantha:** “Cardón” tiene usos en cerca viva, frutal y para forrajes.
- **Agave Prosopis laevigata:** “Mezquite” tiene usos en cerca viva, frutal y para forrajes.
- **Agave Yucca filifera:** “Palma China” es usada en cerca viva, es comestible, medicinal, y una fuente de fibras y sapogeninas, además de ser ornamental.

Diseño de plantación 3 bolillos

Uno de los métodos implementados para la siembra es el tresbolillo o de triángulo, el cual consiste en que cada una de las plantas ocupan en el terreno cada uno de los vértices de un triángulo equilátero, conservando siempre la misma distancia entre plantas que entre filas (Carbo y Vidal, s.f.).

Para determinar el número de plantas que caben en una determinada superficie a cultivar, se considera la siguiente fórmula:

$$num_{plantas} = \frac{S}{d^2 cos30} \quad (1)$$

donde S corresponde a la superficie a cultivar (metros cuadrados) y d la distancia entre plantas (metros).

Este sistema resulta particularmente útil, pues permite que cada planta pueda tener las horas de luz requeridas para su óptimo crecimiento, favoreciendo un uso óptimo del terreno cultivable y la generación de un microclima, que evita que se escape la humedad del terreno, disminuyendo la evaporación y la erosión. (Iglesias, 2021)

A continuación, la figura 1 ilustra el método previamente descrito.

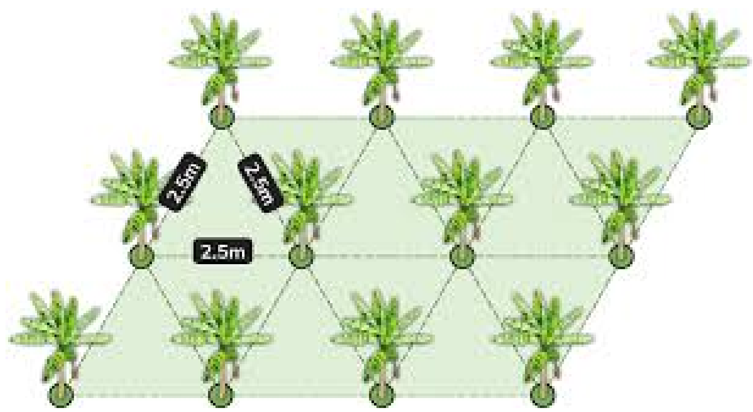

Figura 1: Ilustración gráfica del sistema tresbolillos

Modelos de solución para ruteo de vehículos

El problema de ruteo de vehículos (VRP) se ha convertido en uno de los problemas de optimización combinatoria más estudiados en los últimos años. Este problema implica encontrar un conjunto de rutas para una flota de vehículos que salen de uno o más depósitos con el fin de satisfacer la demanda de clientes ubicados en diferentes lugares. El enfoque de estudio ha sido generalmente enfocarse en optimizar un solo objetivo, sin embargo, lo ideal para su aplicación real sería optimizar múltiples objetivos (Sarmiento, 2014). Una representación gráfica de la solución del ruteo de vehículos se presenta en la figura 2.

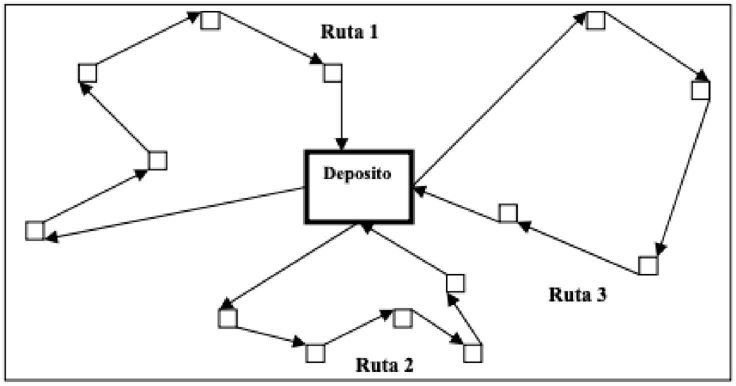

Figura 2: Ejemplo de solución al VRP

A continuación, se enlistan algunas de las técnicas y modelos implementados para la solución de este tipo de problemas:

- **Sistema de hormigas:** Las hormigas se comunican mediante feromonas que les permiten identificar los caminos más cortos entre su nido y una fuente de alimentos. Esta particularidad se ha aprovechado para resolver problemas de optimización que requieren una mejora significativa en los tiempos de cómputo para aplicaciones específicas. La optimización por colonia de hormigas (OCH) es una metaheurística basada en el comportamiento natural de estos insectos (Robles, 2010).

- **Método de composición musical:** Técnica que se inspira en la analogía del proceso creativo de la composición musical en un entorno sociocultural. Consiste en crear una sociedad de compositores que define una red de interacción entre ellos. Inicialmente, se genera un conjunto aleatorio de soluciones para cada compositor y posteriormente, los compositores interactúan entre sí, analizando la información recibida (Montes, Mora, Obregón, De Los Cobos, Rincón, Gutiérrez y Lara, 2020).

- **Método de ramificación y acotación (Branch and bound):** Consiste en una enumeración en árbol en el cual el espacio de las variables enteras se divide de forma sucesiva dando lugar a problemas lineales que se resuelven en cada nodo del árbol. Estos problemas lineales se obtienen relajando las restricciones de integralidad y añadiendo restricciones adicionales (Martínez, s.f.).

- **Algoritmo genético híbrido generacional:** Esta técnica utiliza un algoritmo genético, el cual realiza una búsqueda local para el proceso de cruza a partir de las soluciones generadas hasta el momento por el algoritmo, en otras palabras, la búsqueda local se aplica con las soluciones que se tienen al momento y se lleva a cabo utilizando distintos tipos de vecindarios para llegar a óptimos locales, los cuales son regresados como individuos para la siguiente generación. (Montes, 2017, p. 40)

- **Búsqueda Tabú:** Metaheurística que orienta un proceso heurístico de búsqueda local hacia la optimización global. Su enfoque se fundamenta en desarrollar y utilizar una serie de estrategias inteligentes para resolver problemas, empleando métodos tanto implícitos como explícitos de aprendizaje. (Batista y Glover, s.f.)

5. Definición del problema

Este problema consiste en optimizar la distribución de plantas desde un depósito central hacia varios polígonos de siembra dispersos. Si bien existen múltiples enfoques para esta situación, en el presente trabajo se busca minimizar la distancia recorrida por los vehículos de transporte, diseñando rutas eficientes que cubran la demanda de cada polígono, considerando capacidades de carga, demandas específicas y suministros disponibles en el depósito.

Esto garantiza una operación más económica, eficiente y sostenible, facilitando la consecución de los objetivos de reforestación de manera efectiva.

A continuación, se definen las variables, parámetros, función objetivo y la restricciones necesarias para encontrar la solución óptima en términos de distancia recorrida por las unidades de transporte.

Identificación de variables

- **Variable binaria que indica si el vehículo realiza una entrega al polígono de siembra:** Esta variable binaria es importante para determinar si cada vehículo realiza o no una entrega en cada polígono de siembra, lo que permite planificar las rutas de distribución de plantas de manera óptima, minimizando la distancia total recorrida.

- **Cantidad de plantas:** Cantidad de plantas de la especie asignadas a la unidad de transporte.Esta variable determina cuántas plantas de cada especie se asignan a cada unidad de transporte, asegurando que se cumpla la demanda en cada ubicación objetivo sin exceder la capacidad de los vehículos.

- **Variables de Ruta:** Ruta asignada a la unidad de transporte que incluye las entregas a las ubicaciones de los polígonos. Esta variable servirá para modelar las rutas que cada vehículo tomará para entregar las plantas, minimizando la distancia total recorrida y optimizando el tiempo de entrega.

Parámetros

- Distancia entre el punto de carga y el punto de entrega para la especie y la unidad de transporte.

- Capacidad máxima de carga de la unidad de transporte

- Tiempo de carga y descarga de las plantas del camión.

- Jornada laboral del personal que estará encargado de las actividades

Restricciones

- **Distribución homogénea de plantas:** Reforestar mínimo 194.868 hectáreas, se debe reforestar por lo menos 3 veces lo que ha sido afectado por el Parque Eólico Dominica. Asegurar que las plantas se distribuyan de manera uniforme y eficiente en todas las ubicaciones objetivo.

- **Disponibilidad de plantas por hectárea:** Al tener un máximo de cada tipo de planta por hectárea, se concluyé que el máximo de plantas por hectárea debe ser 524.

- **Restricciones de Capacidad:** Capacidad máxima de carga de cada unidad de transporte

- **Restricción de Variable Binaria:** Variable binaria que indica si la unidad de transporte realiza una entrega en el polígono de siembra.

- **Restricción de Jornada Laboral:** Parámetro que indica el máximo de horas diarias que los trabajadores pueden dedicarle a las actividades (transporte, carga y descarga).

6. Modelo Matemático

Definición de conjuntos

- **Vértices:**

$$V = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13\}, \quad v \in V$$

Donde los vértices simbolizan el número del polígono: {1: Polígono 18, 2: Polígono 1, 3: Polígono 3, 4: Polígono 4, 5: Polígono 5, 6: Polígono 20, 7: Polígono 23, 8: Polígono

24, 9: Polígono 17, 10: Polígono 16, 11: Polígono 19, 12: Polígono 25, 13: Polígono 26}.

Es importante mencionar que cada uno de los vértices anteriores se pueden convertir en origen o destino en algún punto de la ruta, por lo que el conjunto posible para estos son:

- **Nodos origen:**

$$I = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13\}, \quad i \in I$$

- **Nodos destino:**

$$J = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13\}, \quad j \in J$$

No obstante, si se considera la ruta completa, tanto el origen como el destino serán la bodega (Vértice 1 - Polígono 18), pues para este problema es necesario que el vehículo regrese a la bodega para cargar más plantas y poder cumplir con la demanda de cada polígono.

- **Especies de plantas:**

$$K = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}, \quad k \in K$$

Donde {1: Agave lechuguilla, 2: Agave salmiana, 3: Agave scabra, 4: Agava striata, 5: Opuntia cantabrigiensis, 6: Opuntia engelmani, 7: Opuntia robusta, 8: Opuntia streptacanta, 9: Prosopis laevigata, 10: Yucca filifera}

- **Número de subruta:**

$$M = \{1, 2, \ldots, n\}, \quad m \in M$$

Donde cada subruta comienza y termina en la bodega (Polígono 18).

- **Arcos:** $A\ (i,\, j) \in V$, donde $i = \{\text{nodo origen}\}$ y $j = \{\text{nodo destino}\}$

Definición de parámetros

C = Matriz que contiene cada uno de los costos (c_{ij}) de ir de i a j (distancia en metros)

d_{kj} = Demanda de plantas tipo k para el nodo v

t = Tiempo de carga y descarga requeridas en cada jornada laboral (1 hora)

l = Jornada laboral máxima del personal encargado de las actividades (8 horas)

v_t = Velocidad promedio del vehículo en terraceria (20 km/h)

r_{ij} = Tiempo que se toma en ir del nodo i al nodo j. Para el cáculo de este parámetro se utiliza la fórmula mostrada a continuación (ecuación 2):

$$r_{ij} = \frac{c_{ij}}{v_t} \tag{2}$$

Q = Volumen (en cm^3) del vehículo (capacidad)

p_k= Volumen (en cm^3) que ocupa cada unidad de planta k

Para la estimación de los volúmenes de las plantas, se utilizaron las dimensiones proporcionadas y se aplicaron. Específicamente, la altura de las plantas se determinó a partir del rango de altura promedio dado en la tabla correspondiente. El diámetro de las plantas se asumió como una décima parte de la altura promedio, lo que simplificó el proceso sin comprometer significativamente la precisión.

El volumen de cada planta se calculó utilizando la fórmula del volumen de un cilindro, la cual se muestra a continuación (ecuación 3):

$$V = \pi \times \left(\frac{d}{2}\right)^2 \times h \tag{3}$$

donde:

- V es el volumen de la planta.
- d es el diámetro de la planta.
- h es la altura de la planta.
- $\pi \approx 3{,}14159$.

Aplicando esta fórmula, se obtuvieron las estimaciones de volumen necesarias para el análisis (Consulte tabla 7 en Anexos). A continuación, la tabla 1 muestra el volumen correspondiente de cada tipo de planta.

Cuadro 1: Tabla de volúmenes para cada tipo de planta k (cm^3)

Especie	**Volumen** (p_k)
Agave lechuguilla	502.65
Agave salmiana	284.83
Agave scabra	62.83
Agave striata	122.72
Opuntia cantabrigiensis	122.72
Opuntia engelmani	122.72
Opuntia robusta	122.72
Opuntia streptacanta	212.06
Prosopis laevigata	212.06
Yucca filifera	122.72

Esta metodología permite una aproximación práctica y eficaz para la estimación del volumen en estudios forestales y de vegetación. El enfoque utilizado en este estudio está inspirado en trabajos previos, como el de Zhang et al. (2015), quienes desarrollaron ecuaciones generalizadas de biomasa para Pinus massoniana. En su investigación, Zhang et al. utilizaron modelos similares de simplificación para estimar la biomasa y los volúmenes de las plantas, demostrando la validez y aplicabilidad de este método en diferentes contextos ecológicos.

g_k= Cantidad de plantas tipo k que satisfacen la demanda de una hectárea.

o_{kv} = Oferta de la planta tipo k del nodo v.

q_k = Capacidad máxima de plantas tipo k en el vehículo.

Definición de variables

Sea x_{ij} una variable binaria tal que:

$$x_{ij} = \begin{cases} 1 & \text{si se realiza el viaje del nodo } i \text{ al nodo } j \\ 0 & \text{en caso contrario} \end{cases}$$

Sea y_{mk} una variable binaria tal que:

$$y_{mk} = \begin{cases} 1 & \text{si en el viaje } m \text{ se lleva la planta tipo } k \\ 0 & \text{en caso contrario} \end{cases}$$

w_{kj} = Cantidad de plantas de tipo k a descargar en el nodo j

Función Objetivo

A partir del planteamiento anterior, se ha definido como función objetivo el minimizar la distancia total recorrida por todas rutas realizadas por el vehículo para satisfacer la demanda de cada tipo de plantas demandadas en cada polígono (Ecuación 4).

$$\text{mín } z = \sum_{i=1}^{I} \sum_{j=1}^{J} c_{ij} x_{ij} \quad \forall\ j \tag{4}$$

Restricciones

Las primeras restricciones a considerar para este modelo son en relación al origen y destino final de cada una de las rutas. Por un lado, la ecuación 5 asegura que cada vehículo comience su viaje saliendo de la bodega.

$$\sum_{j \in V} x_{1j} = 1 \quad \forall\ j \tag{5}$$

Por otro lado, la segunda restricción garantiza que el destino final de la ruta de cada vehículo sea la bodega, es decir, que se regrese a la bodega al finalizar el recorrido. La ecuación 6 muestra esta restricción:

$$\sum_{j \in V} x_{j1} = 1 \quad \forall\ j \tag{6}$$

Para considerar las restricciones de oferta y demanda, se tiene por una parte, la restricción 3 (Ecuación 7) expresa que la suma de todas las plantas tipo k que sean descargadas en el nodo j deben satisfacer la demanda de plantas tipo k del mismo nodo.

$$\sum_{j \in J} w_{kj} = d_{kj} \quad \forall\ k, j \tag{7}$$

Sumado a ello, se añade la restricción relacionada con la oferta del nodo origen y la cantidad de plantas que se dejan de un tipo específico en el nodo destino, pues esta última tiene que ser menor a la oferta que recibe por parte de todos los nodos posibles de origen. Cabe mencionar que esta oferta se multiplica por una variable binaria, pues solamente si se decide tomar ese arco, la nueva oferta se formará por la suma de las ofertas provenientes de los distintos nodos origen (Ecuación 8).

$$\sum_{i=1}^{K} \sum_{j=1}^{J} x_{ij} o_{kj} - w_{kj} \geq 0 \quad \forall k \tag{8}$$

Asimismo, se considera la variable binaria que determina si se llevarán plantas de un tipo específico en determinado número de viaje y se relaciona con la cantidad de plantas que se descarga en cada nodo, pues solamente si se decide llevar de esa planta, habrá plantas que dar en los nodos que se visiten (Ecuación 9).

$$w_{kj} \leq M * y_{mk} \quad \forall k, m, j \tag{9}$$

Donde M es una constante lo suficientemente grande que puede ser igual a la oferta máxima de plantas tipo k para cualquier nodo v.

En relación a esto, aparece también la restricción de que la suma de todas las cantidades que se dejen en cada nodo de un tipo de planta específico tiene que ser menor o igual a la oferta del nodo, y esto se relaciona con la variable binaria y_{mk}, pues solamente si esta se activa, se podrán descargar plantas. Esto se refleja en la siguiente ecuación 10.

$$\sum_{j \in J} w_{kj} \leq o_{kv} * y_{mk} \quad \forall k, m, v \tag{10}$$

La capacidad se ve reflejada en la restricción mostrada en la ecuación 11, la cual nos indica que el transporte solo puede llevar una capacidad máxima de plantas por ruta(q_k).

$$\sum_{k=1}^{K} \sum_{j=1}^{J} w_{kj} \leq q_k \tag{11}$$

Para definir cuántas plantas de cada tipo abastecen en su totalidad el vehículo, se consideró el volumen de cada planta en cm^3 (p_k)y la cantidad de plantas que se necesitan para una hectárea (g_k). Posteriormente, se sumaron todos los resultados de estas multiplicaciones. Esto se expresa en la ecuación 12.

$$Q = \sum_{k=1}^{K} p_k g_k \tag{12}$$

Después, se dividió esa capacidad total del vehículo entre p_k para obtener la capacidad máxima del camión para cada tipo de cada planta, definida como q_k (Ecuación 13).

$$q_k = \frac{Q}{p_k} \tag{13}$$

La tabla 2 muestra los resultados de utilizar la ecuación anterior.

Cuadro 2: Tabla de capacidadades máximas del camión para cada tipo de planta

Especie	**Capacidad máxima (q_k)**
Agave lechuguilla	220.91
Agave salmiana	389.84
Agave scabra	1767.29
Agave striata	904.81
Opuntia cantabrigiensis	904.81
Opuntia engelmani	904.81
Opuntia robusta	904.81
Opuntia streptacanta	523.62
Prosopis laevigata	523.62
Yucca filifera	904.81

La siguiente restricción (ecuación 14) hace referencia al límite de tiempo con el que se cuenta por día, ya que una jornada laboral máxima es de 8 horas. A partir de ello, se consideran los distintos trayectos que se pueden realizar, y si alguno de estos es activado (debido a que es una variable binaria), será multiplicado por el tiempo que se toma en realizar dicho trayecto. Además, se le suma una constante de 1 hora, puesto que es el tiempo total que se consume en todas las cargas y descargas realizadas durante el día.

$$\sum_{i=1}^{I} \sum_{j=1}^{J} x_{ij} r_{ij} + t \leq l \tag{14}$$

Otra restricción importante a considerar es la ecuación de flujo, la cual establece que una vez llegado al nodo destino, solo se puede partir de ahí para el nodo siguiente (ecuación 15).

$$\sum_{j \in V} x_{ij} - \sum_{k \in V} x_{ki} = b_i \begin{cases} 1 & i = \text{origen - bodega} \\ -1 & i = \text{destino - bodega} \\ 0 & \text{otro caso} \end{cases} \tag{15}$$

Recordando que la ruta completa tiene el mismo origen y destino, es decir, la bodega.

Finalmente, es importante establecer que la cantidad que se dejan de cada tipo de planta tiene que ser igual o mayor a 0, puesto que no es posible dejar un valor negativo (ecuación 16).

$$w_{kj} \geq 0 \tag{16}$$

7. Método heurístico

Un método heurístico que podría resultar de utilidad para la solución de un problema de este tipo es el del vecino más cercano, el cual consiste en generar rutas uniendo vértices que se encuentren a la menor distancia. Esta secuencia de inserción de puntos comienza en el depósito e incorpora el punto más cercano de todos sus posibles caminos. Después de ser incluido, se inicia la búsqueda del siguiente punto a ser añadido, el cual también se decide con la distancia, que ahora es del nuevo nodo en el que se encuentra el vehículo a todos los posibles nodos a los que podría ir. (Ramírez, 2016, p. 18)

La inserción de puntos puede verse restringida por diveros parámetros. Para el enfoque de este proyecto, se consideraron las restricciones relacionadas con la capacidad del vehículo y el tiempo disponible de acuerdo a la jornada laboral. Con estas consideraciones en mente, la inserción de puntos se realiza de manera secuencial hasta que se hayan incorporado todos los puntos. Por su parte, los recorridos se realizan en la matriz de distancias, que inician seleccionando la menor distancia desde el depósito hacia los demás clientes. En caso de surgir empates, la heurística los rompe arbitrariamente, para esto es necesario que las distancias del depósito hacia los demás clientes se encuentren en la primera fila de la matriz.

El algoritmo desarrollado en Python (consulte código en anexos) se centra en satisfacer por completo la demanda de una planta específica antes de pasar al siguiente tipo de planta.

Se considera una ruta al conjunto de subrutas necesarias para satisfacer la demanda total de la planta tipo k $\forall K$. Para decidir el próximo nodo a visitar en cada subruta, se busca el nodo que se encuentra a una menor distancia del nodo en que el vehículo se encuentra actualmente y se descarga su respectiva demanda del tipo de planta que se transporta. Después, pasa al siguiente polígono más cercano, y así sucesivamente hasta que no tenga más plantas que descargar o se hayan cumplido las 8 horas de jornada laboral.

En caso de que el vehículo todavía tenga plantas disponibles pero el nodo más cercano requiera de una demanda mayor, el algoritmo toma la decisión de visitar ese nodo, descargar lo que le queda y posteriormente regresar a la bodega para terminar su subruta. Tras volver a cargar el vehículo y comenzar una nueva subruta, vuelve a considerar las distancias a todos los nodos que no han sido satisfechos en su totalidad y visita el que se encuentra más cerca de este.

8. Experimentación y resultados

Características de la PC y software

A continuación el Cuadro 3 muestra los detalles de la PC utilizada para realizar el modelo.

Características	**Detalles**
Modelo de la PC	huawei matebook 14s
Sistema Operativo	Windows 10
Capacidad de disco duro	2 TB
Memoria RAM y capacidad	16.0 GB (15.8 GB utilizable)
Tipo de procesador	11th Gen Intel(R) Core(TM) i7-11370H @ 3.30GHz 3.30 GHz
Número de núcleos	Núcleos: 4, Procesadores lógicos: 8
Software utilizado y versión	Microsoft Windows 10

Cuadro 3: Características de la PC y software

Tamaño del problema

De acuerdo a la manera en la que se planteó el problema, se determina que es de un tamaño mediano, pues solamente se decidió trabajar con 13 polígonos, los cuales cubrían un total de 75.09 hectáreas. Al ser 13 nodos en el grafo, la cantidad de arcos disponibles fue de 169.

En relación a las variables consideradas, se encuentra x_{ij}, una variable binaria que decide si realiza o no el viaje del nodo i al j, considera 169 variables. Además, y_{mk}, una variable binaria que decide si en el viaje m se lleva la planta tipo k, considera alrededor de 1680 variables, pues en cada subruta tomaba esta decisión, y en total se hacen 168 viajes para satisfacer la demanda total, y esto es multiplicado por los tipos de plantas disponibles, es decir, 10. Finalmente, una tercera variable fue introducida, la cual nos decía la cantidad de plantas tipo k a descargar en el nodo j (w_{kj}), y debido a que es un valor específico de acuerdo al arco tomado, se establece que son un total de 169. Por ello, se concluye que el problema cuenta con un total de 2018 variables.

Finalmente, el modelo cuenta con un total de 632 parámetros, los cuales se desglosan de la siguiente manera:

- Costos (c_{ij}) de ir de i a j (13^2 =169 parámetros)
- Demanda de plantas para cada nodo (d_{kj}) ($10 * 13 = 130$ parámetros)
- Tiempo de carga y descarga (t) (1 parámetro)
- Jornada laboral máxima (l) (1 parámetro)
- Velocidad promedio del vehículo (v_t) (1 parámetro)
- Tiempo que se toma en ir del nodo i al nodo j (r_{ij}) (169 parámetros)
- Volumen del vehículo (Q) (1 parámetro)
- Volumen que ocupa cada planta k (p_k) (10 parámetros)
- Cantidad de plantas tipo k que satisfacen la demanda de una hectárea (g_k) (10 parámetros)
- Oferta de la planta tipo k del nodo v (o_{kv}) ($10 * 13 = 130$ parámetros)
- Capacidad máxima de plantas tipo k en el vehículo (q_k) (10 parámetros)

Resultados

El cuadro 4 presenta los datos obtenidos del proceso de optimización de rutas de distribución de plantas, considerando diferentes combinaciones de polígonos a visitar. Se realizaron múltiples ejecuciones para evaluar la calidad de la solución y el tiempo de procesamiento explorando la influencia de la dimensión del problema.

Cuadro 4: Resultados de plantación de polígonos

Grupo	**Núm. Polígonos**	**Calidad de la solución (m)**	**Tiempo de proc. (s)**
	5	142,880.00	0.0125
	5	140,920.00	0.0133
Pequeño	5	108,720.00	0.0119
	5	112,760.00	0.0124
	5	167,720.00	0.0181
	10	260,920.00	0.0240
	10	217,020.00	0.0210
Mediano	10	247,300.00	0.0221
	10	268,480.00	0.0239
	10	230,440.00	0.0216
	12	279,700.00	0.0327
	12	279,700.00	0.0327
Mediano	12	279,700.00	0.0327
	12	279,700.00	0.0327
	12	279,700.00	0.0327
Mediano (Problema)	13	352100.00	0.0407

A partir de la tabla anterior, se identifica la variación presentada en cada uno de los grupos de prueba. En el caso del grupo pequeño, los valores de distancia oscilan entre los 108,000 y 143,000 metros, presentando un promedio de 134,600 metros recorridos. Este grupo tuvo un tiempo de procesamiento promedio de 0.01364 segundos. Con respecto al grupo mediano de 10, se obtuvo un valor óptimo promedio de 244, 832 metros y un tiempo de 0.02252 segundos. Finalmente, el tercer grupo de prueba, con un tamaño de 12, presentó la misma distancia total para todas las pruebas, así como el mismo tiempo de 0.0327 segundos. Analizando este caso, se puede inferir que estos no cambian debido a que solamente se excluía un polígono de los totales, por lo que probablemente estos se repitieron.

Cabe mencionar que las distancias totales también presentan un incremento en cada conjunto de prueba, lo cual se fundamenta en que cada vez requieren cubrir más polígonos y más demanda, y por tanto, el vehículo necesita llevar a cabo más vueltas a la bodega para reabastecerse.

Los resultados obtenidos en la tabla anterior se muestran de manera gráfica en la figura 3. En esta se puede observa el promedio de cada uno de los conjuntos de prueba y cómo los tiempos tienden a incrementar conforme aumenta el tamaño del problema, es deicr, se aumenta la cantidad de nodos.

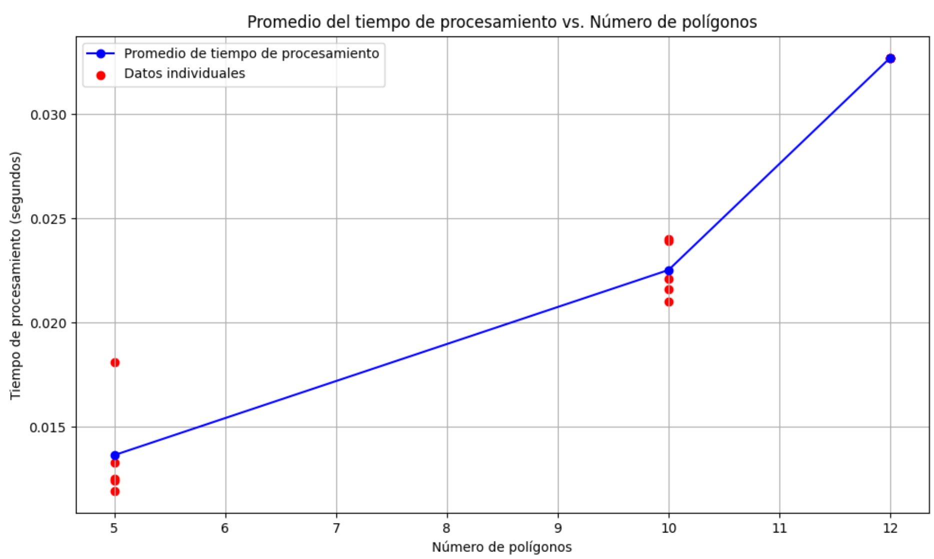

Figura 3: Gráfica de tiempos de ejecución

A pesar de no haber presentado una diferencia tan significativa entre los tiempos de cada grupo, se puede concluir que el tiempo de ejecución aumenta conforme incrementa el tamaño del conjunto a solucionar, y por ello, puede llegar a ser muy tardado cuando se trabaje con más variables y parámetros.

9. Conclusiones

La reforestación es una estrategia de vital importancia para solucionar los problemas generados por el cambio climático, así como conservación de la biodiversidad e incrementar participación y la cohesión social.

La realización de este proyecto permitió la aplicación de conceptos de álgebra lineal para el desarrollo de un modelo de optimización que se enfocara en este aspecto social, resultando de gran relevancia para la mejora de la comunidad y ahorro de recursos.

A pesar de no haber obtenido una solución por medio del modelo matemático desarrollado, se implementó un método heurístico a través del uso de Python, el cual logró resolver el problema en cuestión de segundos.

Tras el análisis de resultados, se reconoce la importancia de desarrollar modelos y códigos optimizados que permitan solucionar problemas de gran magnitud sin requerir de tanto tiempo y recursos, es decir, implementar soluciones que sean fácilmente escalables y viables.

A partir de lo anterior, se concluye que el uso de algoritmos para la solución de problemas reales de gran tamaño puede resultar muy complejo y requerir de un dispositivo con gran capacidad de procesamiento, a diferencia de los métodos heurísticos, los cuales logran encontrar soluciones muy cercanas a las óptimas en cuestión de segundos.

10. Referencias

Batista, B. y Glover, F. (s.f.). Introducción a la Búsqueda Tabú. https://leeds-faculty.colorado.edu/glover/fred %20pubs/329 %20- %20Introduccion %20a %20la %20Busqueda %20Tabu %20TS_Spanish %20w %20Belen(11-9-06).pdf

Carbo, A. y Vidal, O. (s.f.). Marqueo de plantaciones. Ministerio de Agricultura. https://www.mapa.gob.es/ministerio/pags/biblioteca/hojas/hd197821.pdf

Comisión Nacional Forestal, Gerencia de Sistema Nacional de Monitoreo Forestal. (2022). Deforestación. https://snmf.cnf.gob.mx/deforestacion/

Espinosa, V., & Saldívar, A. (2021). Depredadores del bosque Paisajes de la deforestación en México. https://depredadores.proceso.mx/intro.html

Hernández, F. (2011). El Altiplano Mexicano o Altiplanicie Mexicana. https://www.expresionesveterinarias.com/2011/11/el-altiplano-mexicano-o-altiplanicie.html

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/

Martínez, A. (s.f.). Programación Entera. https://www.dma.uvigo.es/ aurea/Programaci %C3 %9Bn_entera_2015.pdf

Montes, E. (2017). Metaheurísticas para el problema de ruteo de vehículos con ventanas de tiempo (VRP-TW). http://zaloamati.azc.uam.mx/bitstream/handle/11191/5699/Metaheuristicas_problema_de_ruteo_de_vehiculos_2017_Montes_MOPT.pdf?sequence=1

Montes, E., Mora, R., Obregón, B., De Los Cobos, S., Rincón, E., Gutiérrez, M. y Lara, P. (2020). Matheurísticas para resolver el problema de ruteo de vehículos con ventanas de tiempo. https://www.scielo.sa.cr/pdf/rmta/v27n2/1409-2433-rmta-27-02-305.pdf

National Geographic. (2024). El último medio siglo ha supuesto el mayor coste de deforestación de la historia de la humanidad, arrasando un 15 por ciento de la superficie mundial de vegetación, equivalente al territorio de España, Portugal y Francia. https://www.nationalgeographic.es/medio-ambiente/deforestacion

Ramírez, L. (20 de junio de 2016). UNA SOLUCIÓN AL PROBLEMA DE RUTEO DE VEHÍCULOS ABIERTO (OVRP), IMPLEMENTANDO LA HEURISTICA DEL VECINO MÁS CERCANO. Universidad Distrital Francisco José de Caldas. https://repository.udistrital.edu.co/bitstream/handle/11349/2985/RamirezRodriguezLuisErnesto.pdf;jsessionid=BD49BBF7147AE8FAAD760F4C666F0A6F?sequence=1

Red de viveros de biodiversidad. (s. f.). Criterios para elegir las especies adecuadas. Red de Viveros de Biodiversidad. https://revivemx.org/index.php/page/como_elegir

Robles, C. (2010). Optimización por colonia de hormigas: Aplicaciones y tendencias. https://revistas.ucc.edu.co.

Ruiz Villar, M. (2018). Reducción de la deforestación. FAO, Forestry Department. https://www.fao.org/sustainable-forest-management/toolbox/modules-alternative/reducing-deforestation/basic-knowledge/es/

Sarmiento, A. (2014). Estudio del problema de ruteo de vehículos con balance de carga: Aplicación de la meta-heurística Búsqueda Tabú. Universidad de La Sabana. https://intellectum.unisabana.edu.co/bitstream/handle/ 10818/9798/Ang

Secretaría de Medio Ambiente y Recursos Naturales. (2021). Informa Semarnat estrategia contra deforestación y tala ilegal. Gobierno de México. Recuperado de https://www.gob.mx/semarnat/prensa/informa-semarnat-estrategia-contra-deforestacion-y-tala-ilegal

FAO. (2022, mayo). Global deforestation slowing but tropical rainforests remain under threat, key FAO report shows. Food And Agriculture Organization Of The United Nations. https://www.fao.org/newsroom/detail/global-deforestation-slowing-but-rainforests-under-threat-fao-report-shows-030522/en

Zhang, J., Li, Z., Wang, W., Tang, L., & Li, H. (2015). Development of generalized biomass equations for Pinus massoniana across different scales. *Forest Ecology and Management, 342*, 50-60.

11. Anexos

Cuadro 5: Tabla de tiempos de transporte del polígono i al j (horas)

	1	2	3	4	5	6	7	8	9	10	11	12	13
1	-	0.0745	0.063	0.061	0.06	0.05	0.022	0.011	0.012	0.025	0.033	0.045	0.058
2	0.0745	-	0.024	0.016	0.014	0.088	0.08	0.078	0.069	0.067	0.08	0.095	0.011
3	0.063	0.024	-	0.011	0.022	0.067	0.064	0.065	0.063	0.065	0.08	0.095	0.06
4	0.061	0.016	0.011	-	0.011	0.073	0.065	0.064	0.057	0.058	0.072	0.087	0.103
5	0.06	0.014	0.022	0.011	-	0.079	0.067	0.065	0.0055	0.053	0.066	0.082	0.096
6	0.05	0.088	0.067	0.0073	0.079	-	0.029	0.04	0.061	0.073	0.083	0.095	0.107
7	0.022	0.08	0.064	0.065	0.067	0.029	-	0.011	0.034	0.047	0.056	0.066	0.079
8	0.011	0.078	0.065	0.064	0.065	0.04	0.011	-	0.024	0.037	0.045	0.055	0.068
9	0.012	0.069	0.063	0.057	0.055	0.061	0.034	0.024	-	0.013	0.023	0.036	0.046
10	0.025	0.067	0.065	0.058	0.053	0.073	0.047	0.037	0.013	-	0.015	0.03	0.045
11	0.033	0.08	0.08	0.072	0.066	0.083	0.056	0.045	0.023	0.015	-	0.016	0.021
12	0.045	0.095	0.095	0.065	0.082	0.095	0.066	0.055	0.036	0.03	0.016	-	0.015
13	0.058	0.011	0.06	0.0103	0.096	0.107	0.079	0.068	0.046	0.045	0.021	0.015	-

Cuadro 6: Tabla de distancias entre polígono i al j (metros)

	1	2	3	4	5	6	7	8	9	10	11	12	13
1	-	1490	1260	1220	1200	1000	440	220	240	500	660	900	1160
2	1490	-	480	320	280	1760	1600	1560	1380	1340	1600	1900	220
3	1260	480	-	220	440	1340	1280	1300	1260	1300	1600	1900	1200
4	1220	320	220	-	220	1460	1300	1280	1140	1160	1440	1740	2060
5	1200	280	440	220	-	1580	1340	1300	110	1060	1320	1640	1920
6	1000	1760	1340	1460	1580	-	580	800	1220	1460	1660	1900	2140
7	440	1600	1280	1300	1340	580	-	220	680	940	1120	1320	1580
8	220	1560	1300	1280	1300	800	220	-	480	740	900	1100	1360
9	240	1380	1260	1140	1100	1220	680	480	-	260	460	720	920
10	500	1340	1300	1160	1060	1460	940	740	260	-	300	600	900
11	660	1600	1600	1440	1320	1660	1120	900	460	300	-	320	420
12	900	1900	1900	1300	1640	1900	1320	1100	720	600	320	-	300
13	1160	220	1200	2060	1920	2140	1580	1360	920	900	420	300	-

Código Python

```python
import numpy as np
import matplotlib.pyplot as plt
import matplotlib.image as mpimg
```

```python
import imageio
import os
import random

#Parametros
#Numero de plantas por cada hectarea de cada especie
HA = [33, 157, 33, 33, 39, 30, 58, 51, 69, 21]

#Volumen de cada planta
Volumenes = [502.65, 284.83, 62.83, 122.72, 122.72, 122.72, 122.72,
    212.06, 212.06, 122.72]

#Volumen por cantidad de plantas requeridas por hectarea
Volumenes_por_especie = [x * y for x, y in zip(HA, Volumenes)]

#Volumen total del vehiculo
Suma_valumenes_por_planta = sum(Volumenes_por_especie)

#Capacidad del vehiculo por especie
Capacidad_vehiculo = [Suma_valumenes_por_planta / x for x in Volumenes]
print(Capacidad_vehiculo)

Poligonos = ['p1', 'p3', 'p4', 'p5', 'p20', 'p23', 'p24', 'p17', 'p16', '
    p19', 'p25', 'p26']

Demanda_1_Agave_lechuguilla = [178.2, 264, 264, 249.48, 45.54, 182.49,
    186.12, 201.63, 186.12, 162.36, 166.65, 156.75]
Demanda_2_Agave_salmiana = [847.8, 1256, 1256, 1186.92, 216.66, 868.21,
    885.48, 959.27, 885.48, 772.44, 792.85, 745.75]
Demanda_3_Agave_scabra = [178.2, 264, 264, 249.48, 45.54, 182.49, 186.12,
    201.63, 186.12, 162.36, 166.65, 156.75]
```

```
Demanda_4_Agave_striata = [178.2, 264, 264, 249.48, 45.54, 182.49, 186.12,
     201.63, 186.12, 162.36, 166.65, 156.75]
Demanda_5_Opuntia_cantabrigiensis = [210.6, 312, 312, 294.84, 53.82,
    215.67, 219.96, 238.29, 219.96, 191.88, 196.95, 185.25]
Demanda_6_Opuntia_engelmani = [162, 240, 240, 226.8, 41.4, 165.9, 169.2,
    183.3, 169.2, 147.6, 151.5, 142.5]
Demanda_7_Opuntia_robusta = [313.2, 464, 464, 438.48, 80.04, 320.74,
    327.12, 354.38, 327.12, 285.36, 292.9, 275.5]
Demanda_8_Opuntia_streptacanta = [275.4, 408, 408, 385.56, 70.38, 282.03,
    287.64, 311.61, 287.64, 250.92, 257.55, 242.25]
Demanda_9_Prosopis_laevigata = [372.6, 552, 552, 521.64, 95.22, 381.57,
    389.16, 421.59, 389.16, 339.48, 348.45, 327.75]
Demanda_10_Yucca_filifera = [113.4, 168, 168, 158.76, 28.98, 116.13,
    118.44, 128.31, 118.44, 103.32, 106.05, 99.75]

#Lista de las demandas por especie
demandas = [Demanda_1_Agave_lechuguilla, Demanda_2_Agave_salmiana,
    Demanda_3_Agave_scabra, Demanda_4_Agave_striata,
    Demanda_5_Opuntia_cantabrigiensis,
            Demanda_6_Opuntia_engelmani, Demanda_7_Opuntia_robusta,
    Demanda_8_Opuntia_streptacanta, Demanda_9_Prosopis_laevigata,
    Demanda_10_Yucca_filifera]

#Lista de los nombres de cada demanda
nombres_demandas = ["Demanda 1 Agave lechuguilla", "Demanda 2 Agave
    salmiana", "Demanda 3 Agave scabra", "Demanda 4 Agave striata", "
    Demanda 5 Opuntia cantabrigiensis",
                    "Demanda 6 Opuntia engelmani", "Demanda 7 Opuntia
    robusta", "Demanda 8 Opuntia streptacanta", "Demanda 9 Prosopis
    laevigata", "Demanda 10 Yucca filifera"]

```

```
Distancias = [
  [0, 1490, 1260, 1220, 1200, 1000, 440, 220, 240, 500, 660, 900, 1160],
  [1490, 0, 480, 320, 280, 1760, 1600, 1560, 1380, 1340, 1600, 1900, 220],
  [1260, 480, 0, 220, 440, 1340, 1280, 1300, 1260, 1300, 1600, 1900,
   1200],
  [1220, 320, 220, 0, 220, 1460, 1300, 1280, 1140, 1160, 1440, 1740,
   2060],
  [1200, 280, 440, 220, 0, 1580, 1340, 1300, 110, 1060, 1320, 1640, 1920],
  [1000, 1760, 1340, 1460, 1580, 0, 580, 800, 1220, 1460, 1660, 1900,
   2140],
  [440, 1600, 1280, 1300, 1340, 580, 0, 220, 680, 940, 1120, 1320, 1580],
  [220, 1560, 1300, 1280, 1300, 800, 220, 0, 480, 740, 900, 1100, 1360],
  [240, 1380, 1260, 1140, 1100, 1220, 680, 480, 0, 260, 460, 720, 920],
  [500, 1340, 1300, 1160, 1060, 1460, 940, 740, 260, 0, 300, 600, 900],
  [660, 1600, 1600, 1440, 1320, 1660, 1120, 900, 460, 300, 0, 320, 420],
  [900, 1900, 1900, 1300, 1640, 1900, 1320, 1100, 720, 600, 320, 0, 300],
  [1160, 220, 1200, 2060, 1920, 2140, 1580, 1360, 920, 900, 420, 300, 0],
  ]

Tiempos = [
  [0, 0.0745, 0.063, 0.061, 0.06, 0.05, 0.022, 0.011, 0.012, 0.025, 0.033,
    0.045, 0.058],
  [0.0745, 0, 0.024, 0.016, 0.014, 0.088, 0.08, 0.078, 0.069, 0.067, 0.08,
    0.095, 0.011],
  [0.063, 0.024, 0, 0.011, 0.022, 0.067, 0.064, 0.065, 0.063, 0.065, 0.08,
    0.095, 0.06],
  [0.061, 0.016, 0.011, 0, 0.011, 0.073, 0.065, 0.064, 0.057, 0.058,
   0.072, 0.087, 0.103],
  [0.06, 0.014, 0.022, 0.011, 0, 0.079, 0.067, 0.065, 0.0055, 0.053,
   0.066, 0.082, 0.096],
```

```
    [0.05, 0.088, 0.067, 0.0073, 0.079, 0, 0.029, 0.04, 0.061, 0.073, 0.083,
      0.095, 0.107],
    [0.022, 0.08, 0.064, 0.065, 0.067, 0.029, 0, 0.011, 0.034, 0.047, 0.056,
      0.066, 0.079],
    [0.011, 0.078, 0.065, 0.064, 0.065, 0.04, 0.011, 0, 0.024, 0.037, 0.045,
      0.055, 0.068],
    [0.012, 0.069, 0.063, 0.057, 0.055, 0.061, 0.034, 0.024, 0, 0.013,
     0.023, 0.036, 0.046],
    [0.025, 0.067, 0.065, 0.058, 0.053, 0.073, 0.047, 0.037, 0.013, 0,
     0.015, 0.03, 0.045],
    [0.033, 0.08, 0.08, 0.072, 0.066, 0.083, 0.056, 0.045, 0.023, 0.015, 0,
     0.016, 0.021],
    [0.045, 0.095, 0.095, 0.065, 0.082, 0.095, 0.066, 0.055, 0.036, 0.03,
     0.016, 0, 0.015],
    [0.058, 0.011, 0.06, 0.0103, 0.096, 0.107, 0.079, 0.068, 0.046, 0.045,
     0.021, 0.015, 0],
   ]

Jornada_laboral = 8

Tiempo_carga_descarga = 1

velocidad_kh = 20
def calcular_distancia(ruta):
    distancia = 0
    for i in range(len(ruta) - 1):
        distancia += Distancias[ruta[i]][ruta[i + 1]]
    return distancia
def calcular_tiempo(ruta):
    tiempo = 0
    for i in range(len(ruta) - 1):
```

```python
        tiempo += Tiempos[ruta[i]][ruta[i + 1]]
    return tiempo
def optimizacion(demanda):
    rutas = []
    demanda_pendiente = demanda[:]
    posicion_actual = 0  # Comienza en la bodega
    tiempo_acumulado = 0
    dias = 1

    while sum(demanda_pendiente) > 0:
        ruta = []
        capacidad_restante = Capacidad_vehiculo[posicion_actual]  # Obtener la capacidad del vehiculo para el poligono actual
        tiempo_acumulado = 0  # Reiniciar el tiempo acumulado para cada dia

        while True:
            menor_distancia = float('inf')
            poligono_mas_cercano = None

            for i in range(1, len(Poligonos) + 1):
                if demanda_pendiente[i - 1] > 0:
                    distancia = Distancias[posicion_actual][i]
                    if distancia < menor_distancia:
                        menor_distancia = distancia
                        poligono_mas_cercano = i

            if poligono_mas_cercano is None:
                break

```

```
            tiempo_necesario = Tiempos[posicion_actual][poligono_mas_cercano] + Tiempos[poligono_mas_cercano][0]
            if tiempo_acumulado + tiempo_necesario > Jornada_laboral - Tiempo_carga_descarga:
                dias += 1  # Iniciar un nuevo dia
                tiempo_acumulado = 0  # Reiniciar el tiempo acumulado
                continue

            ruta.append(poligono_mas_cercano)
            capacidad_restante -= demanda_pendiente[poligono_mas_cercano - 1]

            if capacidad_restante < 0:
                demanda_pendiente[poligono_mas_cercano - 1] = abs(capacidad_restante)
                capacidad_restante = 0
            else:
                demanda_pendiente[poligono_mas_cercano - 1] = 0

            if capacidad_restante == 0 or tiempo_acumulado >= Jornada_laboral - Tiempo_carga_descarga:
                break

            posicion_actual = poligono_mas_cercano
            tiempo_acumulado += Tiempos[posicion_actual][poligono_mas_cercano] + Tiempos[poligono_mas_cercano][0]

        rutas.append(ruta)
        posicion_actual = 0  # Regresa a la bodega para la siguiente ruta

    return rutas, dias
```

```
# Guardar resultados de rutas y distancias totales
resultados = []
tiempo_total_horas = 0
distancia_total_general = 0
dias_actuales = 1
tiempo_acumulado_dia = 0

for idx, demanda in enumerate(demandas, 1):
    rutas, dias = optimizacion(demanda)
    distancia_total_demanda = sum([calcular_distancia([0] + ruta + [0])
    for ruta in rutas])
    tiempo_total_demanda = sum([calcular_tiempo([0] + ruta + [0]) for ruta
     in rutas])
    resultados.append((idx, rutas, distancia_total_demanda,
    tiempo_total_demanda, dias))
    tiempo_total_horas += tiempo_total_demanda
    distancia_total_general += distancia_total_demanda

    print(f'{nombres_demandas[idx - 1]}:')
    print(f"Ruta {idx}:")
    for ruta_idx, ruta in enumerate(rutas, 1):
        ruta_poligonos = ['Bodega'] + [Poligonos[i - 1] for i in ruta] + [
    'Bodega']
        tiempo_ruta = calcular_tiempo([0] + ruta + [0])
        if tiempo_acumulado_dia + tiempo_ruta > Jornada_laboral -
    Tiempo_carga_descarga:
            dias_actuales += 1
            tiempo_acumulado_dia = tiempo_ruta
            print(f"  Dia {dias_actuales}: Subruta {ruta_idx}: {' -> '.
    join(ruta_poligonos)} - Tiempo: {tiempo_ruta:.2f} horas")
        else:
```

```
            tiempo_acumulado_dia += tiempo_ruta
            print(f"  Dia {dias_actuales}: Subruta {ruta_idx}: {' -> '.join(ruta_poligonos)} - Tiempo: {tiempo_ruta:.2f} horas")

    print(f"  Distancia total: {distancia_total_demanda:.2f} metros")
    print(f"  Tiempo total: {tiempo_total_demanda:.2f} horas")

# Calcular dias totales requeridos
dias_totales = tiempo_total_horas // (Jornada_laboral - Tiempo_carga_descarga) + 1

# Imprimir dias totales requeridos para todas las demandas
print("Total de dias requeridos para todas las demandas:", dias_totales)

# Imprimir la distancia total general
print(f"Distancia total general: {distancia_total_general:.2f} metros")

# Convertir distancia total general a tiempo en horas
tiempo_total_general_horas = distancia_total_general / 1000 / velocidad_kh

# Imprimir la distancia total general en tiempo
print(f"Tiempo total general: {tiempo_total_general_horas:.2f} horas")
```

Listing 1: Código en Python de Optimización de Rutas

Link del codigo: https://colab.research.google.com/drive/1m0iTTWRW1dAtdHKdVTsMfA-g448hcbxU?usp=sharing

Cuadro 7: Cálculo de volúmenes en cm^3

Especie	**Altura (cm)**	**Diámetro (cm)**	**Volumen (cm^3)**
Agave lechuguilla	40	4 (asumiendo 0.1 * altura)	$\pi \times \left(\frac{4}{2}\right)^2 \times 40 = 502{,}65$
Agave salmiana	40	28.483 (asumiendo 0.1 * altura promedio)	$\pi \times \left(\frac{28{,}483}{2}\right)^2 \times 40 = 284{,}83$
Agave scabra	20	6.283 (asumiendo 0.1 * altura promedio)	$\pi \times \left(\frac{6{,}283}{2}\right)^2 \times 20 = 62{,}83$
Agave striata	25	12.272 (asumiendo 0.1 * altura promedio)	$\pi \times \left(\frac{12{,}272}{2}\right)^2 \times 25 = 122{,}72$
Opuntia cantabrigiensis	25	12.272 (asumiendo 0.1 * altura promedio)	$\pi \times \left(\frac{12{,}272}{2}\right)^2 \times 25 = 122{,}72$
Opuntia engelmanii	25	12.272 (asumiendo 0.1 * altura promedio)	$\pi \times \left(\frac{12{,}272}{2}\right)^2 \times 25 = 122{,}72$
Opuntia robusta	25	12.272 (asumiendo 0.1 * altura promedio)	$\pi \times \left(\frac{12{,}272}{2}\right)^2 \times 25 = 122{,}72$
Opuntia streptacantha	27.5	21.206 (asumiendo 0.1 * altura promedio)	$\pi \times \left(\frac{21{,}206}{2}\right)^2 \times 27{,}5 = 212{,}06$
Prosopis laevigata	27.5	21.206 (asumiendo 0.1 * altura promedio)	$\pi \times \left(\frac{21{,}206}{2}\right)^2 \times 27{,}5 = 212{,}06$
Yucca filifera	25	12.272 (asumiendo 0.1 * altura promedio)	$\pi \times \left(\frac{12{,}272}{2}\right)^2 \times 25 = 122{,}72$