

Instituto Tecnológico y de Estudios Superiores de Monterrey
Escuela de Ingeniería y Ciencias
Ingeniería en Ciencias de Datos y Matemáticas



Reto: Análisis de puesta en uso de un sistema de entrega e-commerce.

Paola Montserrat Vega Ortega - *A01274773*
Antonio Patjane Ceballos - *A01657978*
Luis Angel López Chávez - *A01571000*
Yuu Ricardo Akachi Tanaka - *A01351969*
Donnet Emmanuel Hernández Franco - *A01352049*
Pablo Monzón Terrazas - *A01562619*

Optimización estocástica

Grupo: 101

Profesores:

Dr. Rafael Muñoz Sánchez
Dr. Fernando Elizalde Ramírez

Monterrey, Nuevo León
30 de agosto de 2023

Índice

1. Introducción al problema	3
1.1. Justificación del problema	3
1.2. Objetivo	4
2. Trabajo relacionado	4
2.1. Problema del agente viajero	4
2.2. Problema de ruteo de vehículos	6
2.3. Simulación de Montecarlo	6
2.4. Simulación de sistemas de entrega de productos o de ruteo de vehículos	7
3. Definición del problema	8
3.1. Identificación de las variables y posibles parámetros	8
3.2. Función objetivo a optimizar y restricciones	9
4. Modelo matemático	9
4.1. Función objetivo a optimizar y restricciones	9
4.2. Selección de vehículo	11
4.3. Distribución de probabilidad del número de unidades que demanda un cliente	12
4.4. Distribución de probabilidad de productos ordenados	13
4.5. Método heurístico o metaheurístico a considerar	14
5. Simulación	15
5.1. Simulación preliminar	17
5.1.1. Greedy	18
5.1.2. Greedy Aleatorio	20
5.1.3. Minimizar Distancia o Tiempo	21
5.2. Simulación de 1000 iteraciones	21
5.2.1. Buscando la mejor combinación de número de camiones y rampas	23
5.3. Greedy	24
5.4. Greedy Aleatorio	24
5.5. Página interactiva	25
6. Conclusiones	26
7. Características de PC	27
8. Anexos	27

1. Introducción al problema

En México, comparando el año 2019 con el 2020, el e-commerce creció 81 % tras la pandemia de Covid-19 [1]. La pandemia provocó un confinamiento lo cual generó nuevos hábitos en los consumidores y un nuevo rol en los canales móviles para la compra de productos y servicios, dado que se evitaba el contacto entre las personas. De igual manera, anteriormente se planteaba la necesidad de incluir un servicio más digitalizado, pero tras la contingencia se tuvieron que tomar estas medidas más rápido desarrollando aplicaciones que sustituyeran actividades que comúnmente se hacían de forma física [2].

Para las empresas, el comercio electrónico representa un valor agregado dado que distintas herramientas tecnológicas pueden retribuir a lograr mayores ventas. Y en comparativa del primer trimestre del 2020 con el primero del 2021 el comercio tuvo un 300 % de desarrollo a pesar del semáforo cambiante de la pandemia. En resumen, se ha visto que en los últimos 10 años el modelo de negocio donde se tiene una tienda virtual dentro de las plataformas ha tenido un crecimiento acelerado, donde en comparación con este año y el 2019 las ventas digitales minoristas han triplicado el valor que tenían. Colocando a México dentro del top 5 países con mayor crecimiento del comercio electrónico después de Indonesia, Filipinas e India [3].

Sin embargo, las empresas ante esta aceleración tuvieron que enfrentar distintos problemas relacionados con retail, catálogo y entrega [4]. Por consiguiente el problema de la demanda por las compras en línea y su sistema de transporte necesita de un proceso que agilice la entrega y satisfaga la demanda de los productos en el menor tiempo posible.

1.1. Justificación del problema

En los últimos años, el comercio electrónico ha experimentado un crecimiento extraordinario. En consecuencia, las empresas necesitan ajustarse a esta tendencia para mantenerse competitivas en términos de innovación y, lo que es aún más importante, para seguir siendo una elección viable para sus clientes [5].

Indudablemente, el acelerado progreso económico conlleva ventajas para las personas, pero el incremento continuo de industrias y empresas también atribuye responsabilidades inherentes. Uno de los desafíos que debemos afrontar se relaciona con la emisión considerable de contaminantes que se liberan en nuestra atmósfera, los cuales están causando daños cada vez más graves a nuestro planeta. Aquí es donde cobra vital importancia la implementación de la logística verde [6]. Un beneficio inherente a la optimización de procesos, particularmente en el ámbito del transporte, es su capacidad para prevenir o al menos reducir la contaminación ambiental.

De igual forma, la optimización de los procesos de entrega de productos, incluyendo la planificación de rutas eficientes, tiene varias justificaciones económicas fundamentales que respaldan su implementación. Algunas de estas razones son:

- Reducción de costos operativos: Minimizar distancias recorridas y el tiempo de entrega involucra obtener ahorros significativos en combustible, mantenimiento de vehículos, horas de trabajo, entre otros costos asociados al transporte. Menos recursos gastados resultan en una mejora de la rentabilidad de una empresa.
- Al tener entregas más rápidas, se tiene una ventaja competitiva en el mercado, atrayendo más

clientes.

- Reducir la distancia minimiza el desgaste de los vehículos y su necesidad de mantenimiento, lo que a su vez alarga su vida útil y reduce los costos de reemplazo.
- Evitar aglomeraciones de camiones durante la carga, así como la inactividad de los trabajadores por una mala distribución de trabajo y tiempo.

Estas razones hacen que la inversión en la optimización de rutas sea una decisión estratégica que puede tener un impacto positivo en la salud financiera y la posición competitiva de una empresa. Sin embargo, esto no termina ahí, este crecimiento extraordinario ha creado que las situaciones se transformen en de manera acelerada, por lo que es necesario optimizar los procesos de logística dentro de la compañía. Para que esto sea lo más apegado a la vida real, se utiliza la simulación de los procesos actuales y se someten a situaciones variadas para encontrar las rutas que se asemejen más al comportamiento que sigue la demanda dadas las distribuciones de productos y cantidad de pedidos.

1.2. Objetivo

El reto consiste en desarrollar un algoritmo capaz de simular la gestión logística para la entrega de productos a múltiples clientes en una empresa de comercio electrónico. El propósito es simular los procesos actuales en diversos escenarios para las bases del comercio. Comenzando con la red de carga, la cual involucra los camiones necesarios y las rampas para abastecerlos, dependiendo de la demanda más probable de acuerdo a los datos históricos. La propuesta también busca mejorar la eficiencia de los tiempos de recorrido, reducir los gastos involucrados, minimizar tanto el impacto del tráfico como la emisión de contaminantes en áreas urbanas y eliminar la inactividad y el tiempo perdido que se podría generar por parte de los trabajadores.

Como resultado final se busca crear una simulación del camino a tomar por parte de los vehículos para la entrega de productos donde se tomen en cuenta varios escenarios posibles. Esto incluye que se pueda adaptar a diferentes cantidades de clientes y su demanda de productos, considerando cuántos y cuáles productos pedirá cada uno según cómo se distribuyen estos datos según su frecuencia. Finalmente, con ayuda de procesos heurísticos y simulaciones, se propone una solución final donde se indica cuántos y cuáles vehículos se usarán según su capacidad, cuántas rampas de carga se tendrán instaladas y las rutas de los vehículos para satisfacer toda la demanda de los clientes.

2. Trabajo relacionado

2.1. Problema del agente viajero

Para este tipo de algoritmos donde se enfoca en la optimización de una o varias rutas intentando minimizar una función de costo o maximizar el beneficio, el problema del agente viajero (o TSP por sus siglas en inglés) es uno de los más relevantes para conocer la naturaleza de estas situaciones. En este problema se establece el agente viajero quien tiene el objetivo de visitar todas las ciudades o puntos de interés presentes siguiendo la ruta óptima [7]. Esta ruta comúnmente es definida como aquel camino donde se minimice la distancia total recorrida cumpliendo el mismo objetivo. Para la resolución de este

tipo de situaciones se puede establecer como un problema de programación lineal [8]. En este método se establece una función objetivo que se busca maximizar o minimizar según el planteo del problema la cual está sujeta a ciertas restricciones controladas por las variables de decisión [8]. Un planteo general siguiendo este método sería como el siguiente, donde se muestra una función objetivo a minimizar:

$$\min \sum_{i=1}^n \sum_{j=1}^n x_{ij} c_{ij} \quad i \neq j \quad (1)$$

Las variables de decisión en este planteamiento consisten de variables binarias $x_{ij} \in \{0, 1\}$, donde si $x_{ij} = 1$ indica que el agente sí viaja de la ciudad i a la ciudad j , 0 de lo contrario. Mientras tanto el parámetro c_{ij} indica el costo de realizar el viaje de la ciudad i a la ciudad j el cual puede consistir de la distancia o costo monetario entre ellos. Por otro lado, generalmente las restricciones en este problema se plantean de la siguiente manera [8]:

$$\sum_i x_{ij} = 1 \quad \forall j \neq i \quad (2)$$

$$\sum_j x_{ij} = 1 \quad \forall i \neq j \quad (3)$$

$$u_i - u_j + n \cdot x_{ij} \leq n - 1 \quad 2 \leq i \leq n, \quad i \neq j \quad (4)$$

En este caso las restricciones se refieren a que por cada ciudad visitada, el agente se dirige a una única ciudad vista en la ecuación (6), y para toda ciudad visitada se proviene de una única ciudad vista en la ecuación (7). Finalmente en la ecuación (8), se asegura que se visitan todas las ciudades y que no se regrese a un mismo punto hasta lograrlo, es decir, para generar un ciclo completo y evitar que se generen subrutinas. En este caso la variable u consiste de un valor discreto que se refiere a la posición del orden en que se visitó una ciudad, donde i es una ciudad de origen, j es su ciudad de destino, y n se refiere al número total de ciudades a visitar [8].

La razón por la relevancia e importancia de este problema es su aplicación en un sinnúmero de situaciones similares, pues la optimización de rutas es un escenario deseable para casi cualquier negocio. Esto es especialmente verdad para el contexto del e-commerce para la venta de productos físicos como es el caso de *Amazon*. Debido a la naturaleza de este mercado, el envío a domicilio de los productos a sus clientes se vuelve uno de los temas más importantes a resolver y optimizar. A este concepto ya se le conoce con el nombre de *Last mile delivery* [9]. Se puede aplicar el problema del agente viajero acomodando las definiciones para ser apropiadas a este contexto. Por ejemplo, las *ciudades* establecidas en la formulación general de TSP se pueden ver como los domicilios de los clientes a los cuales todos se deben visitar minimizando el tiempo para cumplir con los límites de horario para la entrega de los productos. Mientras que el agente viajero es sustituido por el medio de transporte, como un camión, que lleva los productos a entregar.

Este planteamiento tiene ciertas limitaciones para este contexto, principalmente el hecho de que se considera en un inicio que el medio de transporte va a visitar todos los domicilios de los clientes lo cual para negocios e-commerce de mayor tamaño y más establecidos ya cuentan con una cantidad plural de medios de entrega. Por lo mismo, no suena como una buena práctica tener un solo vehículo para la entrega de todos los productos de un único lapso de tiempo. Asimismo, el transporte tiene una capacidad limitada de carga de los productos, provocando que de manera práctica el vehículo tendrá que regresar

o llegar a una rampa de carga varias ocasiones para seguir con la entrega de pedidos. No obstante, una posible solución para ajustar la situación a un planteamiento TSP sería definir un número de vehículos predefinido para la entrega de productos a los cuales se les asigne a cada uno una cantidad de domicilios a los cuales llegar [9]. Así logrando seccionar la situación en varios problemas tipo TSP. Sin embargo, surgen nuevas consideraciones, como la cantidad óptima de vehículos a utilizar al igual de como asignar de la mejor manera los domicilios que les corresponde a cada uno de ellos. Es por esto que se han ideado nuevos planteamientos para estas situaciones específicas relacionadas a la entrega de productos de un negocio e-commerce.

2.2. Problema de ruteo de vehículos

Un planteamiento más adecuado para este tipo de situaciones es al basarlo en el problema de ruteo de vehículos (VRP por sus siglas en inglés). Este problema es un tanto similar a TSP, en donde similarmente se busca encontrar un camino óptimo minimizando una función de costo o maximizando una función de beneficio. Sin embargo, en este problema se basa en la premisa que se cuenta con varios vehículos que en conjunto deben visitar todos los puntos de interés (comúnmente vistos como clientes) para satisfacer una demanda [10]. En el contexto del e-commerce, este planteamiento se adapta completamente pues el enfoque se encuentra en la satisfacción de la demanda de clientes mediante la entrega de productos, especialmente cuando se consideran las diferentes variantes de este problema. Algunos de estas variantes tomando en consideración ventanas de tiempo en la que se tienen que realizar las entregas, la capacidad de los vehículos para transportar los productos, e incluso donde los vehículos no deben finalizar sus rutas de los depósitos de los cuales salieron.

Uno de las variantes más relevantes para el e-commerce es el Problema de Enrutamiento del Vehículo con Ventanas de Tiempo (**VRPTW**). En esta variante se tiene un ventana de tiempo donde el cliente debe ser atendido [10]. Este tipo de problema se puede observar en tiendas de e-commerce como en *Amazon* con la famosa entrega de productos en el mismo día, donde la ventana de tiempo es el momento en el que se hizo el pedido hasta el final del día.

Por otro lado, se tiene el Problema de Enrutamiento del Vehículo Capacitado (**CVRP**), dicha variante de igual manera resulta bastante útil al hablar del e-commerce. En esta los vehículos de transporte tienen una capacidad de carga limitada y estos deben de recoger y entregar artículos en varios puntos ya establecidos. Cada uno de estos artículos tienen la información de su peso y/o volumen, mientras que los vehículos tienen una capacidad máxima de peso y/o volumen que pueden transportar [11].

2.3. Simulación de Montecarlo

Al abordar este problema, podemos encontrar situaciones y herramientas similares que se pueden utilizar, una de estas es la simulación Montecarlo. Esta es un modelo probabilístico que puede incluir elementos de incertidumbre y permite predecir los diversos resultados posibles. Para poder utilizar el mismo, es necesario conocer datos históricos del problema a analizar para predecir una serie de resultados futuros, tomando en cuenta que en cada simulación se puede arrojar un resultado diferente. Teniendo la ventaja de poder tomar en cuenta factores aleatorios y las consecuencias de los mismos, este modelo nos permite tener una perspectiva más amplia. Sin embargo, a comparación de un método convencional más

determinista, este no nos arroja una decisión definitiva [12].

Su funcionalidad se basa en la ergodicidad, esta garantiza que cada estado sea accesible desde cualquier otro en un número finito de pasos, describiendo el comportamiento estadístico de un punto en movimiento en un sistema cerrado. El punto en movimiento eventualmente pasará por todos los lugares posibles del sistema [12]. Esto es el fundamento de la simulación de Montecarlo, en la que la computadora ejecuta suficientes simulaciones para producir la salida eventual de diferentes entradas.

Para obtener un resultado con un alto nivel de eficiencia, es necesario realizar una gran cantidad de simulaciones, lo cual puede generar un costo computacional y un retraso de tiempo. Dependiendo del número de variables este puede variar, ya que este debe de simular los posibles escenarios, pero este normalmente cuenta con un mínimo de decenas de miles.

Para realizar un análisis de Montecarlo se necesitan de 3 componentes:

- Variables de entrada: valores aleatorios con alguna distribución de probabilidad.
- Variables de salida: resultado del análisis.
- Modelo matemático: ecuación que relaciona la variable de entrada con la de salida.

2.4. Simulación de sistemas de entrega de productos o de ruteo de vehículos

La simulación de sistemas de entrega de productos o de ruteo de vehículos una de las herramientas más importantes para abordar este tipo de problemas logísticos donde se puede visualizar de manera computacional como optimizar las rutas en un mundo donde el comercio electrónico se vuelve indispensable en el día a día de cada persona [13]. Uno de los métodos más comunes a utilizar siendo la simulación de Montecarlo [12] anteriormente mencionada.

En un principio pareciera una forma de resolución de problemas inferior a los modelos matemáticos que buscan una solución exacta. Sin embargo, su característica compleja y poca de flexibilidad al momento de agregar variables y nuevas consideraciones significan un gran problema al momento de utilizar esta solución a diferencia de las simulaciones [13]. En el e-commerce, el uso de las simulaciones para la resolución de *Last Mile Delivery* [9] son fundamentales ya que de manera relativamente simple se pueden integrar nuevos elementos y variables al problema, incluyendo el tráfico y el clima, e incluso automatizando la recolección de estos factores en tiempo real mediante API's para tener una representación precisa de las rutas a seguir al momento de hacer entrega de productos. Asimismo, la simulación es intuitiva de entender pues es una representación de los casos prácticos de estos problemas, lo cual ayuda a determinar cuales factores son los más importantes a considerar al tener el objetivo de minimizar la huella de carbono generada o el de las rutas [13]. Debido a la naturaleza de las computadoras se tienen que hacer cierto ajustes al momento de plasmar la situación, algunos siendo la discretización del tiempo dentro de la simulación al igual del uso de algoritmos heurísticos llegando a resultados menos precisos [13]. Para el caso específico de sistemas de entrega se tienen que considerar aspectos como la capacidad de carga máxima del vehículo, las rampas de carga, entre otras por la que una de las soluciones más populares en cuanto a código es mediante la programación orientada a objetos usando lenguajes como Python o C#.

Uno de los ejemplos más conocidos es el caso del uso de simulaciones de Uber y Uber Eats. En sus proyectos, tienen como objetivo la normalización de carros con conductores para cumplir con la entrega

de comida y servicio de transporte [14]. En este caso, no solo realizaban simulaciones para probar la Inteligencia Artificial para seguir las normas de tránsito y visión computacional, sino que también para poder determinar las rutas óptimas tomadas por el vehículo minimizando tiempo de espera del cliente o pasajero [14]. Un caso más particular es sobre el uso de simulaciones para la solución de problemas de ruteo en e-commerce es el caso de Amazon Prime Air [15]. En este caso, Amazon tenía el objetivo de evolucionar las rutas de entrega de productos mediante el uso de drones. Gracias a las simulaciones realizadas pudieron caracterizar la aerodinámica de los drones.

Aplicado al objetivo de este trabajo, primero se busca definir los distintos escenarios posibles donde se tomen en cuenta, por ejemplo, cuántos productos un cliente pedirá y de esos cuáles productos serán siguiendo una distribución probabilística según su historial de datos recolectados.

3. Definición del problema

El problema a abordar implica la optimización del sistema de transporte contenido en una base de datos de una empresa de e-commerce. Se busca garantizar la satisfacción de las entregas a cada cliente, al mismo tiempo que se reduzcan las emisiones contaminantes, se optimicen los plazos de entrega, minimizar los gastos de transporte, minimizar el tiempo perdido y limitar la congestión vehicular en el entorno urbano especificado.

Esto se resolverá mediante la simulación en donde se ingresará aleatoriamente diferentes pedidos, tomando en cuenta la distribución de número de pedidos y la distribución de cantidad de veces que se pide cada producto. Una vez teniendo eso, se podrá correr la simulación de Montecarlo gran cantidad de veces para así poder definir el número de camiones óptimos para realizar los viajes y el número óptimo de rampas de cargas para evitar grandes filas, pérdidas de tiempo e inactividad (teoría de colas). Finalmente, con ayuda de modelos matemáticos, como lo es el VRP y heurísticas como el voráz (o *Greedy*), se podrá definir las rutas óptimas para entregar los productos y de igual manera definir el modelo y la marca de los vehículos que serán necesarios para transportar y entregar los productos de la empresa de e-commerce.

3.1. Identificación de las variables y posibles parámetros

Variables:

- Número de camiones
- Número de rampas de carga necesarias

Parámetros:

- Capacidad de carga de los camiones (Volumen)
- Número de pedidos
- Ubicación de los pedidos
- Volumen de los pedidos

3.2. Función objetivo a optimizar y restricciones

El propósito consiste en mejorar la eficiencia en la entrega de productos, reducir los gastos relacionados con el transporte, aumentar el rendimiento de entregas por cada unidad de transporte, tener la cantidad óptima de unidades empleadas en las entregas, así como la cantidad de las rampas de carga, ahorrar combustible, seleccionar la unidad más adecuada para el tipo de terreno y entregar los pedidos en tiempo y forma.

Las limitaciones que se deben tener en cuenta al diseñar el algoritmo incluyen la cantidad disponible de unidades, el máximo número de entregas y capacidad por unidad, la cantidad máxima de combustible por unidad, los costos asociados al llenado de los tanques de combustible, velocidad máxima y los horarios establecidos para las entregas.

4. Modelo matemático

Aún tomando en cuenta que el principal los métodos a utilizar consiste en obtener propuestas de soluciones al sistema de ruteo mediante soluciones, es importante plantear un modelo matemático para tener en mente la función objetivo minimizar al igual que las restricciones del sistema. Se busca encontrar una solución donde se intente minimizar una función de costo (sea distancia recorrida o tiempo invertido) el cual regirá finalmente en el número vehículos a utilizarse y el número de rampas de carga empleadas. Mientras tanto, se pueda ocupar el volumen total de los pedidos.

4.1. Función objetivo a optimizar y restricciones

Debido a que se tiene una flotilla de vehículos de distribución heterogénea, en la que cada vehículo cuenta con una capacidad distinta de carga. El modelo matemático a implementar consistiría en una variante del modelo VRP clásico, llamado HFVRP (*heterogeneous fleet vehicle routing problem*). Esta variante del modelo clásico VRP si contempla la distinta capacidad de carga de cada vehículo de la flotilla. Para representarlo matemáticamente se realiza de la siguiente manera donde se establece como un problema de programación lineal con la siguiente función objetivo [16]:

$$\min \sum_{k=1}^p \sum_{i=1}^n \sum_{j=1}^n c_{ij} \cdot x_{ijk} \quad (5)$$

$$x_{ijk} \in \{0, 1\}$$

Para este trabajo se intenta minimizar una función de costo regida por el parámetro de costo c_{ij} el cual indica un costo de viajar del nodo origen i al nodo destino j (en este contexto representando los clientes a satisfacer). Mientras tanto se define la variable de decisión x_{ij} el cual consiste de una variable binaria la cual indica si es que el vehículo k realiza el viaje del nodo i al nodo j . Es por esto que para los índices se definen en los siguientes conjuntos:

$$i \in \{0, 1, 2, \dots, n\}$$

$$j \in \{0, 1, 2, \dots, n\}$$

$$k \in \{0, 1, 2, \dots, p\}$$

En este caso n indica el número total de clientes a atender, mientras que p indica el número de vehículos a emplear para esta tarea. No obstante, debido a que se formula de una manera similar a lo que sería un TSP y el hecho de incorporar consideraciones de capacidades, algunas de las restricciones a seguir son las siguientes:

$$x_{iik} = 0 \quad \forall k \quad (6)$$

$$\sum_{i=1}^n x_{ijk} = \sum_{i=1}^n x_{jik} \quad \forall j \quad (7)$$

$$\sum_{j=2}^n x_{1jk} = 1 \quad \forall k \quad (8)$$

$$\sum_{j=2}^n x_{1jk} = 1 \quad \forall k \quad (9)$$

Como se puede observar, se sigue una lógica muy similar a un escenario TSP cuando en un principio únicamente se toman en cuenta los conceptos generales de este tipo de problemas. Un ejemplo siendo la restricción (6), donde establece que para todo vehículo k no se puede viajar de un nodo a si mismo para evitar bucles. Por otro lado, en la restricción (7) se interpreta como, para todo cliente visitado, todos los vehículos de entregas que entran deben salir del nodo. En la restricción (8) se establece que para todos los vehículos se tiene que salir de la rampa de carga o del punto de origen. Sin embargo, se busca tomar en cuenta las capacidades de los vehículos para satisfacer la entrega de productos a los clientes con cierta demanda.

$$\sum_{i=1}^n \sum_{j=2}^n q_j x_{ijk} \leq Q \quad \forall k \quad (10)$$

En este caso, se toma una capacidad fija Q para cualquiera de los vehículos, mientras que el parámetro q_j indica la demanda del cliente del nodo destino j . En la restricción (10) se asegura que todos nodo visitado para todo vehículo, la demanda del cliente no debe ser mayor a la capacidad de éste. Mientras tanto, una de las restricciones más importantes a considerar en este tipo de problemas es la eliminación de *subtours* donde no se completa un ruta en el cual vehículo salte y termina regresando al punto de origen. Es por esto que se tienen una gran variedad de formas en las que se formulan restricciones para evitar estos casos, uno de ellos siendo la formulación de Miller-Tucker-Zemlin (MTZ) [16].

$$u_j - u_i \geq q_j - Q(1 - x_{ijk}) \quad \forall i, j \in V \setminus \{1\} \quad i \neq j \quad (11)$$

$$q_i \leq u_i \leq Q \quad \forall i \in V \setminus \{1\} \quad (12)$$

En estas dos últimas restricciones (11) y (12) se define de manera similar a un escenario TSP común donde las variables u indican el orden en que se visitaron los nodos mientras que V se refiere al conjunto completo de nodos que se pueden visitar.

$$\sum_{i=1}^n \sum_{j=2}^n t_{i,j} x_{ijk} \leq T \quad \forall k \quad (13)$$

De igual forma, una de las restricciones mas importantes en este caso de estudio, es el tiempo máximo que cada conductor de los camiones puede laborar: T . El parámetro $t_{i,j}$ hace referencia a la matriz del tiempo necesario para llegar desde el nodo i hasta el nodo j . De acuerdo con la Ley Federal del Trabajador, después de haber trabajado 6 horas continuas de trabajo se le asigna por lo menos una hora de descanso para consumir alimentos. Asimismo, la NORMA Oficial Mexicana NOM-087-SCT-2-2017, que establece los tiempos de conducción y pausas para conductores de los servicios de autotransporte federal, se les otorga un descanso de 30 minutos por cada 5 horas de conducción continua. Por ultimo se considera la hora que se tarda en llenar el camión. Tomando en cuenta todas estas condiciones y una jornada laboral común de 8 horas se tiene que $T \leq 6.5hrs$ [17].

4.2. Selección de vehículo

En cuanto a la selección de cantidad de vehículos, se decidió hacer una investigación sobre los beneficios y costos de cualquiera de los siguientes vehículos para realizar el proceso de entrega:

Tipo Unidad	Modelo	Marca	Capacidad de carga	Capacidad tanque
CHASIS CABINA	2021	NISSAN	1,770 kg	75 L
ELF 300	2015	ISUZU	3,000 kg	75 L
514 SEMI LONG S 300	2019	HINO	2,470 kg	70 L
ELF 300	2013	ISUZU	2,400 kg	70 L
716 LONG SERIE 300	2019	HINO	3,920 kg	100 L
MANAGER FURGON L2H2	2021	PEUGEOT	1,850 kg	125 L
CHASIS CABINA	2015	NISSAN	3,000 kg	80 L
MANAGER FURGON L4H2	2021	PEUGEOT	1,850 kg	125 L

Cuadro 1: Vehículos disponibles.

- Dentro de los modelos proporcionados, se puede observar que la mayoría son chasis de camiones de carga, por lo que el volumen a considerar es de la caja que van a transportar.
- Estos camiones tienen un largo promedio entre 3.5 - 4 m, un ancho de 2 - 2.3 m y una altura de 2.45 m, por lo que tomamos un volumen de 18 m^3 [18].
- Para el caso de la camioneta, como tiene menores dimensiones, consideramos un volumen de 15 m^3 [18].

Tomando en cuenta diversos factores como que el comercio tiene principalmente sus pedidos en la ciudad de Monterrey, con puentes que pueden ser un impedimento para los camiones, que el 75 % de los paquetes cuentan con un volumen menor a 50 centímetros cúbicos y que para tener una menor cantidad de retrasos de carga de gasolina es necesario tener mayor capacidad en el tanque. Es por esto que se decidió escoger entre los camiones de Manager Furgon de Peugeot, entre los modelos L2H2 y L4H2, que tienen una capacidad de combustible de 125 lts y una capacidad de carga de 1850 kg. Para decidir la mejor opción, se realizó un análisis más detallado de las fichas técnicas de cada uno de estos camiones, véase en el Cuadro 2.

Característica	L2H2	L4H2
Rendimiento/emisiones en CIUDAD	10.9 km/lt	10.8 km/lt
Rendimiento/emisiones en CARRETERA	15.9 km/lt	15.4 km/lt
Rendimiento/emisiones COMBINADO	13.5 km/lt	13.3 km/lt
Emisiones CO_2 (g/km)	195	199
Precio	\$ 571,900 MXN	\$ 586,900 MXN
Capacidad	11.5 m^3	15 m^3

Cuadro 2: L2H2 vs L4H2 [19]

Con esto, se llegó a la conclusión que el camión más adecuado es el de Manager Furgon L2H2 2021 de Peugeot, esto debido a que más eficiente y tiene más rendimiento en la ciudad, carretera y ambos. Además, este modelo tiene menor emisiones de CO_2 , es decir, es más amigable para el medio ambiente, un aspecto importante para las empresas de hoy en día. La única desventaja que tiene el modelo L2H2 es el volumen de área de carga, con una diferencia de $3.5 m^3$ menor que la de L4H2, no obstante decidimos arriesgar la capacidad del camión por el rendimiento y las emisiones que tiene, esto debido a que los productos a transportar tienen un volumen pequeño, por lo que no será de mucha utilidad tener una capacidad grande ya que la restricción del tiempo se alcanzará primero [20]. En la Figura 1 podemos ver la comparativa entre ambos modelos y sus respectivas dimensiones.

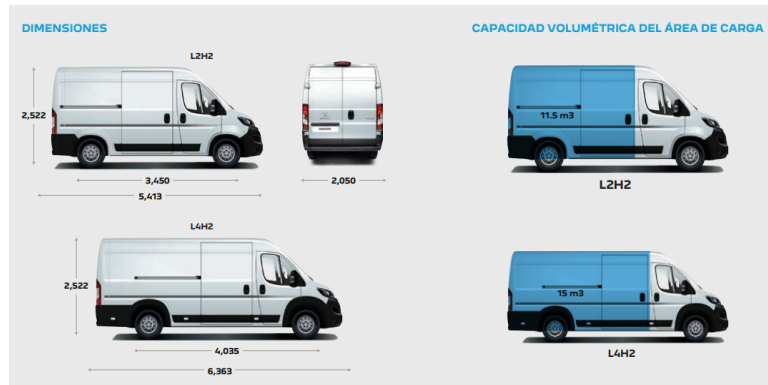


Figura 1: Comparación de modelo L4H2 y L2H2 [19].

Por lo que el parámetro de capacidad máxima quedará de la siguiente manera:

$$Q = 11.5m^3 \quad (14)$$

4.3. Distribución de probabilidad del número de unidades que demanda un cliente

Para poder efectuar dicho modelo se hicieron distintas pruebas con el historial de pedidos realizados por diferentes clientes para poder encontrar la distribución que se ajustara mejor a los datos. Es decir, se hizo un recuento de la cantidad de productos pedidos por cada cliente y agrupamos la cantidad de unidades que demanda un cliente para hallar su distribución.

El histograma de los datos se puede ver representado dentro de la Figura 2.

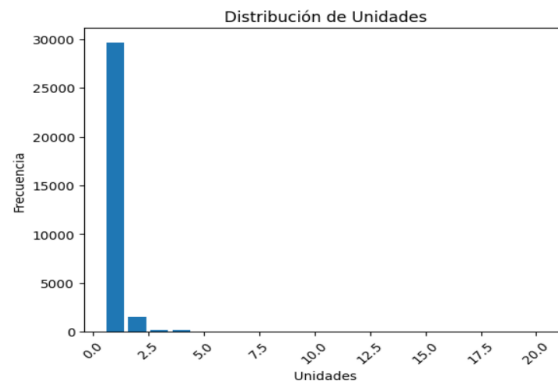


Figura 2: Distribución de unidades solicitadas por cada cliente.

De igual forma, dentro de R, se realizó un ajuste con la distribución normal, cauchy, logística, uniforme, Weibull, F, entre otras, como se muestra en la Figura 3. Donde se validó que las distribuciones que más se acercaban a los valores fueron la F y logarítmica. Para esto, se tomó en cuenta el parámetro de AIC para poder evaluar que tan cercana era la representación de las distribuciones con los datos que se están analizando, donde un número menor representa mayor semejanza a la graficación de los datos reales.

Sin embargo, tras realizar otro análisis en Python junto con las distribuciones que R no tomaba en cuenta, se decidió utilizar para la simulación las frecuencias relativas ya que las distribuciones probadas no se acercaban realmente a la distribución de los datos reales.

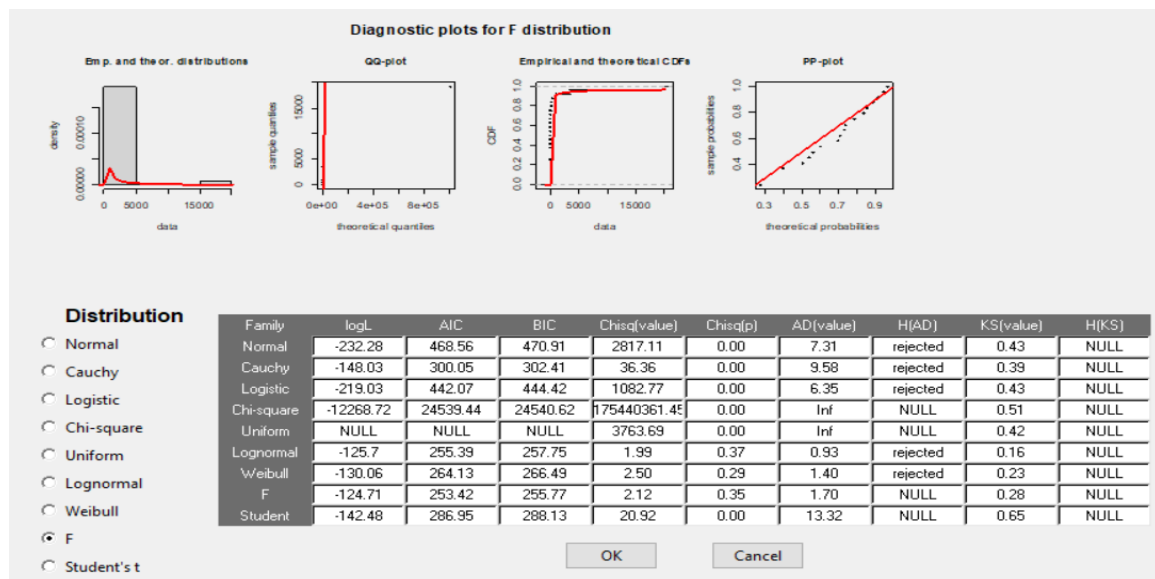


Figura 3: Prueba con diferentes distribuciones en R.

4.4. Distribución de probabilidad de productos ordenados

Con el mismo proceso, se efectuaron diferentes pruebas con las distribuciones de probabilidad de pedir cada producto. A continuación se presenta un histograma de la frecuencia de productos pedidos y la distribución y los diferentes resultados dentro de R.

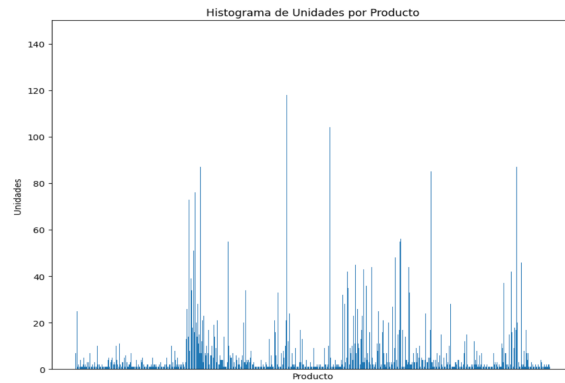


Figura 4: Histograma Productos Ordenados

Podemos apreciar en la Figura 5 que de acuerdo al indicador AIC las distribuciones que más se asemejan son la exponencial, logarítmica y F. No obstante, tras realizar el mismo análisis dentro de Python, se decidió de la misma manera utilizar las frecuencias relativas.



Figura 5: Prueba con diferentes distribuciones en R.

4.5. Método heurístico o metaheurístico a considerar

Algoritmo greedy

El algoritmo Greedy se basa en la idea de que tomando la decisión aparentemente óptima en cada paso, se puede llegar a una solución global óptima o, al menos, a una solución que esté lo más cerca posible de la óptima. Sin embargo, no siempre garantiza la mejor solución, ya que no considera todas las posibilidades y puede quedarse atrapado en óptimos locales. Sin embargo, nos permite tener soluciones mas rápidas y eficientes con un menor costo computacional [21].

El algoritmo toma el punto con menor distancia y se eligió esta metodología porque considera una lista de elementos solo una vez y decide la siguiente actividad, además de que es sencillo de implementar y es menos complejo que otras heurísticas como programación dinámica. Otro punto a favor es que se puede usar en conjunto con otros algoritmos de optimización.



Figura 6: Greedy [22]

Algoritmo greedy aleatorio

Como el Greedy normal, el enfoque no garantiza la mejor solución posible. Depende de las decisiones aleatorias tomadas durante el proceso, pero puede ser de utilidad para desarrollar una solución rápida cuando no se puede tomar en cuenta todas las posibles soluciones. Un problema del Greedy normal es que tiende a quedarse en los mínimos o máximos locales y es ahí donde aparece el Greedy Aleatorio en donde busca salir de los mínimos o máximos locales para así encontrar los mínimos o máximos absolutos.

El algoritmo comienza con una solución inicial aleatoria considerando las variables de decisión dentro del rango permitido. Y de esta forma se iteran diferentes veces en busca de una solución óptima aproximada. Es decir, el Greedy Aleatorio tiene una solución muy similar a la de Greedy normal, no obstante en esta heurística el modelo toma los mejores n nodos y elige el modo de manera aleatoria. Las soluciones son evaluadas y en cada iteración se realiza una pequeña modificación aleatoria a la solución actual para poder agregar, eliminar o cambiar la solución actual dependiendo su resultado. En este caso, se mantiene la solución que favorezca la función objetivo. Para su implementación en las simulaciones se tomarán los 3 clientes más cercanos y se elegirá aleatoriamente uno de ellos.

5. Simulación

Para la implementación del algoritmo primero se definió la matriz de distancias y de tiempo de 100 clientes aleatorios ya que la base de datos proporcionada únicamente indicaba las siguientes columnas:

- Colonia.
- Código postal
- Calle
- Número de casa

Por lo tanto, se utilizó la API de Bing Maps que como resultado indica los valores de latitud y longitud para después obtener las distancias y el tiempo tomando en cuenta un viaje en carro. En algunos recorridos hay valores negativos, lo cual indica que no se encontró alguna ruta que conectara esas dos ubicaciones. Esto se debía ya que la API de Bing Maps identificaba las direcciones mal, dándonos como resultado latitudes y longitudes en otros países. Para arreglar este problema se buscó manualmente dónde se encontraban los lugares y se modificó uno por uno las latitudes y longitudes para asegurarnos

de que no se encontraran errores al obtener las matrices de distancia y tiempo con la API de Bing. En el Cuadro 3 podemos observar unos ejemplos de las direcciones ya con su respectiva latitud y longitud.

Colonia	C.P.	Calle	Núm. Casa	Latitud	Longitud
NUEVO REPUEBLO MTY	64700	CHIAPAS	1209	25.6558923	-100.3022032
MODERNA	64530	GARDENIA	3337	25.69927262	-100.279797
PASEO DE LOS ANDES SAN	66444	VULCANO	1417	25.7375605	-100.2610573
ANTIGUA STA ROSA	66465	SANTA ROSA	102	25.791586	-100.2296511
VILLA AZUL SAN	66420	CINCO	613	25.75621867	-100.3020355
...

Cuadro 3: Ejemplo de Direcciones con Latitud y Longitud.

Después, para poder obtener la demanda, productos solicitados y volumen correspondiente de usuarios se realizó la simulación de los mismos dado las frecuencias relativas. En el Cuadro 4 se puede observar los rangos utilizados para la simulación de la demanda por cliente:

Unidades	Frecuencia	p(x)	Limite Inferior	Limite Superior
1	19235	0.775511	0	0.775511
2	3546	0.142967	0.775511	0.918478
3	1024	0.041285	0.918478	0.959763
4	478	0.019272	0.959763	0.979035
5	194	0.007822	0.979035	0.986856
...

Cuadro 4: Distribución de los productos.

Dicha simulación se realizó en Python, inicialmente se indicaban dentro de los datos el código del producto, unidades solicitadas y el ID de la persona que pidió ese producto y unidades. Primero se hizo un recuento de la suma de unidades pedidas por producto para poder calcular la frecuencia relativa de los productos. Además, junto con estas mediciones se pudo obtener la frecuencia relativa acumulada, el límite inferior y superior. En el siguiente cuadro 5 se pueden apreciar los resultados de los primeros productos.

Producto	Unidades	p(x) relativa	p(x) Relativa Acumulada	Limite Inferior	Limite Superior
23	2	0.000057	0.000057	1.00E-28	0.000057
29	2	0.000057	0.000114	5.71E-05	0.000114
30	7	0.0002	0.000314	1.14E-04	0.000314
60	1	0.000029	0.000343	3.14E-04	0.000343
66	7	0.0002	0.000543	3.43E-04	0.000543

Cuadro 5: Distribución de los productos.

Con ello, se hizo un algoritmo que generara un número aleatorio comprendido entre $[0,1]$ y como

paso siguiente se ubicara dicho numero aleatorio dentro de los rangos de frecuencia relativa acumulada de la cantidad de productos por factura. Este proceso se realizo para los 100 clientes para determinar la demanda de cada uno.

Para poder generar la simulación de qué productos se solicitaron se realizó el mismo procedimiento de ubicación de números aleatorios dentro de dichos rangos de frecuencia relativa acumulada, para n productos que hayan sido demandados por el cliente i por medio una función para generar un vector de probabilidades de n productos. A continuación se despliegan unos ejemplos de dichos vectores al realizar una única simulación, es decir los siguientes ejemplos cambian cada vez que se ejecuta la simulación.

Cliente	Demanda	Vector Probabilidades Aleatorias
0	0	[0.0, 0.0, 0.0, 0.0]
1	1	[0.0, 0.9462064596611349, 0.0, 0.0]
2	3	[0.8117741, 0.06473312, 0.73956884, 0.0]
3	1	[0.0, 0.0, 0.8931135181516047, 0.0]
4	1	[0.9146910055905922, 0.0, 0.0, 0.0]
...

Cuadro 6: Vector de probabilidades.

Para poder dar un significado a ese vector de probabilidades se definió una función para determinar el producto de dicha probabilidad. La selección del producto está dada por el límite inferior y superior de las probabilidades de los productos mencionadas en el Cuadro 5. Igualmente, obtenidos los productos solicitados se determinó el volumen de los mismos. Donde se sumaba el volumen de todos los productos solicitados.

En el Cuadro 7 podemos ver ejemplificado algunos clientes junto con su demanda, productos y volumen.

Cliente	Demanda	Productos Comprados	Volumen Total
1	1	[34361.0]	0.000013
2	3	[27618.0, 22776.0, 35394.0]	0.002517
3	1	[37188.0]	0.010000
4	1	[17439.0]	0.020912
...

Cuadro 7: Primeras filas de la simulación.

5.1. Simulación preliminar

Una vez teniendo la matriz de demanda y volumen por cliente, para poder hacer uso del algoritmo de Greedy y Greedy Aleatorio se tomó en cuenta que la capacidad máxima de los camiones es de $11.5 m^3$ dado la selección del transporte explicado en la sección anterior. A su vez, otra restricción indicada es el tiempo máximo dado por 6.5 horas (23400 segundos) de la jornada laboral considerando descanso. Para cada simulación se decidió guardar varios resultados para poder observar el desempeño que tenían

las rutas, estos fueron los siguientes:

- Nodos a visitar en orden
- Distancia recorrida por ruta
- Tiempo recorridio por ruta
- Volumen utilizado del camión
- Distancia Total recorrida por todas las rutas
- Tiempo total utilizado por todas las rutas
- % de volumen no utilizado en total
- % de tiempo no utilizado en total
- Cantidad de rutas a realizar

Se realizaron dos algoritmos, uno que minimiza la distancia recorrida y otro el tiempo a recorrer con el objetivo de observar cuál tenía mejores resultados, es decir el que minimizara más los costos de distancia y tiempo.

5.1.1. Greedy

Distancia Mínima

Las siguientes rutas fueron las que se encontraron, donde el Nodo 0 es la central de carga de productos.

- Ruta 1: Nodos [0, 36, 44, 90, 83, 27, 32, 26, 98, 68, 13, 3, 85, 39, 28, 41, 70, 16, 34, 47, 7, 14, 79, 54, 59, 45, 72, 5, 71, 77, 43, 50, 21, 38, 62, 61, 93, 89, 15, 76, 10, 66, 22, 91, 37, 67, 56, 58, 88, 0]

Distancia: 75 km, Tiempo: 5.34 hrs, Volumen: $3.7419 m^3$

- Ruta 2: Nodos [0, 64, 48, 95, 51, 75, 11, 42, 30, 20, 19, 81, 17, 55, 31, 35, 53, 57, 84, 46, 63, 25, 52, 8, 29, 2, 1, 94, 24, 40, 4, 87, 74, 82, 23, 86, 97, 0]

Distancia: 110 km, Tiempo: 5.43 hrs, Volumen: $3.7034 m^3$

- Ruta 3: Nodos [0, 78, 49, 65, 9, 96, 33, 92, 73, 69, 6, 12, 18, 99, 0]

Distancia: 178 km, Tiempo: 4.8 hrs, Volumen: $1.4125 m^3$

- Ruta 4: Nodos [0, 60, 80, 0]

Distancia: 84 km, Tiempo: 1.88 hrs, Volumen: $0.0087 m^3$

Dando como resultado de esta simulación lo siguiente:

- Rutas a realizar: 3 rutas
- Distancia Total: 369 km
- Tiempo Total: 13.68 hrs

- % de volumen no utilizado: 63.78 %
- % de tiempo no utilizado: 17.1 %

Tiempo Mínimo

A su vez, se utilizó el mismo algoritmo, pero en este caso minimizando la cantidad de tiempo de recorrido. Los resultados de las rutas se pueden ver a continuación:

- Ruta 1: Nodos [0, 24, 73, 80, 83, 76, 31, 47, 99, 38, 12, 71, 69, 57, 64, 86, 84, 60, 9, 35, 77, 72, 68, 13, 85, 97, 7, 43, 96, 66, 95, 26, 70, 4, 45, 54, 78, 93, 49, 92, 19, 90, 87, 29, 17, 28, 51, 50, 61, 36, 46, 30, 0]

Distancia: 92 km, Tiempo: 4.95 hrs, Volumen: 9.1 m^3

- Ruta 2: Nodos [0, 79, 37, 91, 16, 27, 65, 32, 55, 58, 98, 94, 33, 2, 48, 25, 15, 40, 41, 11, 3, 74, 59, 75, 10, 34, 81, 22, 1, 62, 23, 39, 42, 18, 0]

Distancia: 130 km, Tiempo: 5.01 hrs, Volumen: 9.69 m^3

- Ruta 3: Nodos [0, 67, 56, 63, 14, 20, 44, 53, 82, 88, 89, 8, 6, 5, 21, 0]

Distancia: 217 km, Tiempo: 4.37 hrs, Volumen: 0.58 m^3 .

Dando como resultado de esta simulación lo siguiente:

- Rutas a realizar: 3 rutas
- Distancia Total: 437 km
- Tiempo Total: 14.21 hrs
- % de volumen no utilizado: 63.78 %
- % de tiempo no utilizado: 13.9 %

La ruta 3 se graficó dentro de Google Maps para poder verificar las direcciones y el tiempo indicado del algoritmo. Dicha ruta se puede apreciar en la Figura 7. Marca un tiempo de 3 hr 53 minutos, pero se debe adicionar 30 minutos de trayecto del nodo final al punto inicial. Lo cual nos da un tiempo de 4 horas con 23 minutos.

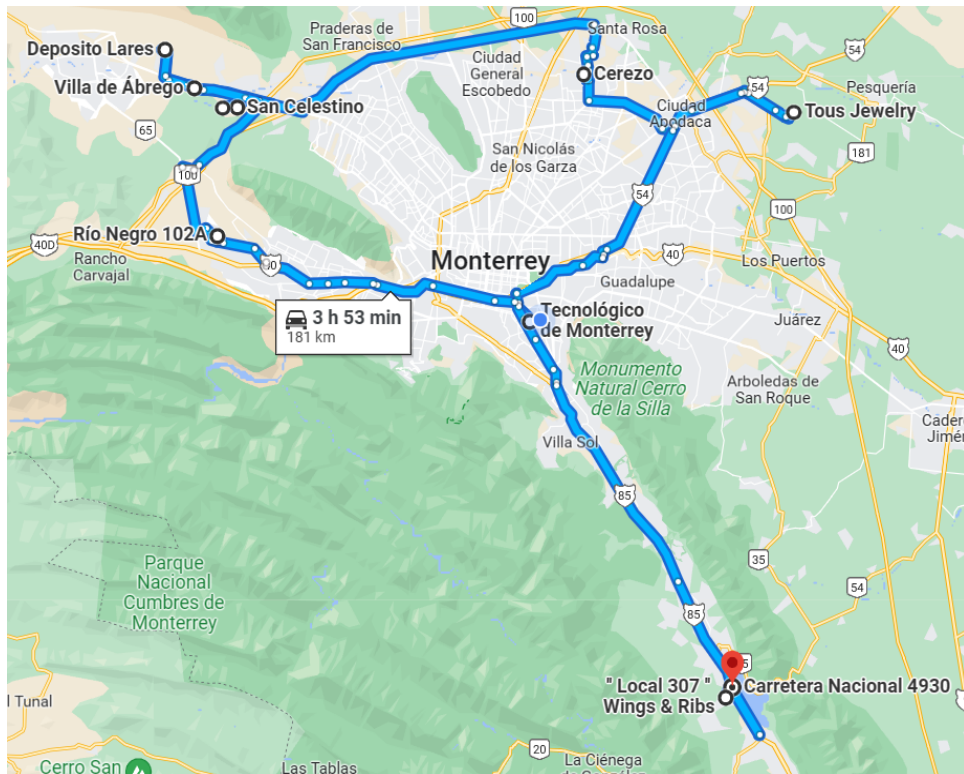


Figura 7: Ruta más corta generada por el algoritmo de VRP tomando en cuenta la distancia.

5.1.2. Greedy Aleatorio

Así mismo se corrió una única simulación para Greedy Aleatorio minimizando primeramente la distancia y seguidamente minimizando el tiempo.

Distancia Mínima

Los resultados de una única simulación fueron los siguientes:

- Rutas a realizar: 7 rutas
- Distancia Total: 1224 km
- Tiempo Total: 28.65 hrs
- % de volumen no utilizado: 90.43 %
- % de tiempo no utilizado: 25.59 %

Tiempo Mínimo

Los resultados de una única simulación fueron los siguientes:

- Rutas a realizar: 10 rutas
- Distancia Total: 1639 km
- Tiempo Total: 40.32 hrs
- % de volumen no utilizado: 89.77 %
- % de tiempo no utilizado: 26.69 %

5.1.3. Minimizar Distancia o Tiempo

Después de tener estos resultados, se compararon los resultados obtenidos al minimizar distancia y al minimizar tiempo, esto con el objetivo de definir con qué se iba a estar trabajando para las 100 simulaciones.

	Greedy		Greedy aleatorio	
	Min distancia	Min tiempo	Min distancia	Min tiempo
Rutas a realizar	3	3	7	10
Distancia total	369 km	437 km	1224 km	1639 km
Tiempo total	13.68 hrs	14.21 hrs	28.65 hrs	40.32 hrs
% volumen no utilizado	63.78 %	63.78 %	90.43 %	89.77 %
% tiempo utilizado	17.1 %	13.9 %	25.59 %	26.69 %

Cuadro 8: Comparación de Resultados Minimizando Distancia o Tiempo.

Como podemos observar en el Cuadro 8 los resultados cuando minimizamos tiempo son peores que cuando minimizamos distancia, ya que aumentan las rutas, la distancia total y el tiempo total. Es por esto que para las 1000 iteraciones se decidió trabajar minimizando la distancia.

5.2. Simulación de 1000 iteraciones

Para realizar el procedimiento de esta sección se crearon variables de promedio de los resultados que arroja cada iteración con el objetivo de observar los valores en los que convergen dichos resultados cuando se tienen 1000 iteraciones. Así mismo, se añadieron más ubicaciones en la base de datos de clientes, lo cual nos daba un total de 300 direcciones diferentes. Tomando en cuenta dichas direcciones se realizaron las mismas actividades que con los usuarios pasados en donde en este primer caso se tomaron en cuenta las siguientes variables:

- Simulaciones a realizar
- Número de sampleos a realizar de clientes
- Tiempo máximo para completar la ruta
- Capacidad de carga máxima de los camiones

Se decidió hacer 1000 iteraciones del algoritmo tomando en cuenta 100 direcciones aleatorias, para este caso se tomó la decisión de cambiar las direcciones cada 100 iteraciones, es decir se cambiaron los clientes 10 veces y considerando una vez más una capacidad máxima de carga de $11.5 m^3$ y un tiempo máximo de 6.5 hrs. Esto con el fin de conocer cuántos camiones y rampas son necesarias para lograr repartir todos los pedidos a los clientes en un día. De igual forma se utilizaron los algoritmos de Greedy y Greedy aleatorio donde haciendo la comparativa se tuvieron los resultados expresados en el Cuadro 9.

	Greedy	Greedy Aleatorio
Promedio de rutas	3.5	9.6
Promedio de distancia	426.42 km	1756.47 km
Promedio de tiempo	16.8 hrs	41.51 hrs
Promedio % de volumen no utilizado	65.06 %	87.48 %
Promedio % de tiempo no utilizado	11.83 %	10.14 %

Cuadro 9: Comparativa entre ambos algoritmos.

Como se pudo observar en el Cuadro 9 en este caso el algoritmo de Greedy tuvo mucho mejores resultados que el Greedy Aleatorio, ya que en el Greedy se ocupan menos rutas y mucho menor promedio de distancia y tiempo. Por lo que en este caso se decidió analizar cuántas rampas se necesitan para el algoritmo de Greedy.

Rampas

Con el fin de analizar cuántas rampas serían los adecuados para esta simulación se decidió checar cuánto tiempo era lo que estaba utilizando el último camión. Es decir, en la simulación anterior se tuvo un promedio de rutas de 3.5 tomando en cuenta que todas las rutas salen al mismo tiempo, es decir, con 4 camiones (porque no se puede tener 3.5 camiones) y 4 rampas. Sin embargo, al analizar cuánto tiempo le sobra al último camión podemos únicamente poner 2 rampas, haciendo que el camión 3 y 4 se formen en la rampa de carga, estando 2 horas en la zona de carga.

Analizamos únicamente el último camión ya que las otras rutas ya tienen el tiempo lleno, debido a que las rutas se generan cuando el camión anterior ya cumplió con el tiempo máximo. Ignoramos la capacidad máxima ya que los productos tienen volúmenes muy pequeños y no llegan a llenar los camiones.

Al obtener el promedio de tiempo que utilizan las últimas rutas en las 1000 simulaciones se tuvo un resultado de 3.38 horas. Es decir está utilizando únicamente aproximadamente la mitad de las horas.

Teniendo en cuenta lo anterior, se decidió que lo mejor sería pasar algunos productos del camión 3 al camión 4 con el objetivo de bajar una hora de su tiempo que será utilizado para formarse en la rampa y así utilizar las 8 horas de trabajo de cada conductor.

Con esta información, tomando en cuenta 2 y 4 rampas para los respectivos camiones, se realizó de nuevo una simulación de 1000 iteraciones agregándole una hora más a los camiones 3 y 4. A continuación una comparativa de los tiempos y otras métricas importantes para conocer el número de rampas necesarias para este modelo.

	4 rampas	2 rampas
Promedio de rutas	3.5	4
Promedio de distancia	426.42 km	434.99 km
Promedio de tiempo	16.8 hrs	17.97 hrs
Promedio % de volumen no utilizado	65.06 %	64.06 %
Promedio % de tiempo no utilizado	11.83 %	9.23 %

Cuadro 10: Comparativa entre número de rampas.

Como podemos observar en el cuadro 10, aumentó el promedio de tiempo, esto es debido a que 2 camiones se formaron y estuvieron una hora más en la zona de carga esperando. Esto nos dice que el último conductor ya está cumpliendo con más de sus horas. Eso lo podemos comprobar viendo el promedio de % de tiempo no utilizado, en donde observamos que bajó considerablemente. Por lo que se logró distribuir bien el tiempo, utilizando la mayor parte del tiempo de trabajo de los conductores.

5.2.1. Buscando la mejor combinación de número de camiones y rampas

En el caso anterior se realizó la simulación intentando buscar cuál sería la mejor opción de número de camiones y número de rampas observando las rutas generadas por el modelo VRP. No obstante, se decidió realizar más simulaciones con el fin de encontrar cuál sería la mejor combinación de número de camiones y de rampas, es decir si tenemos n camiones, cuántas rampas serían lo óptimo. Para hacer esto se añadieron dos variables más a la simulación:

- Simulaciones a realizar
- Número de sampleos a realizar de clientes
- Tiempo máximo para completar la ruta
- Capacidad de carga máxima de los camiones
- Cantidad de rampas
- Cantidad de camiones

Es decir se añadieron como variables la cantidad de rampas y de camiones debido a que en las simulaciones se intenta encontrar la mejor combinación que minimizara la distancia total, el tiempo total, el % de tiempo no utilizado y el % de volumen no utilizado. Para todos los resultados obtenidos es posible la variación al momento de intentar replicarlos debido a la naturaleza aleatoria de las simulaciones. Para todas las pruebas se hicieron un total de 1,000 simulaciones, donde en cada 100 iteraciones, se realiza un sampleo de clientes de los totales 300 para tener una mejor representación de éstos.

Se dividieron en grupo de muestras con los siguientes resultados:

	SIM 1	SIM 2	SIM 3	SIM 4	SIM 5	SIM 6	SIM 7	SIM 8
# Simulaciones	1000	1000	1000	1000	1000	1000	1000	1000
# Sampleos	10	10	10	10	10	10	10	10
# Camiones	5	7	10	12	5	7	10	12
Tiempo máximo (seg)	23400	23400	23400	23400	23400	23400	23400	23400
Capacidad de carga (m^3)	11,5	11,5	11,5	11,5	11,5	11,5	11,5	11,5
Cantidad de rampas	1	1	1	1	2	2	2	2

Cuadro 11: Simulaciones por grupos 1 y 2 rampas

	SIM 9	SIM 10	SIM 11	SIM 12
# Simulaciones	1000	1000	1000	1000
# Muestras	10	10	10	10
# Vehículos	5	7	10	12
Tiempo máximo (seg)	23400	23400	23400	23400
Capacidad de carga (m^3)	11,5	11,5	11,5	11,5
Cantidad de rampas	4	4	4	4

Cuadro 12: Simulaciones por grupos 4 rampas

5.3. Greedy

	SIM 1	SIM 2	SIM 3	SIM 4	SIM 5	SIM 6	SIM 7	SIM 8
Distancia total (km)	413,052	423,319	420,888	430,41	403,99	434,782	419,266	409,369
Tiempo total (hr)	16,49	16,65	16,6	16,83	16,18	16,79	16,52	16,44
Volumen no utilizado	55,76 %	57,21 %	55,66 %	57,66 %	54,24 %	54,08 %	55,71 %	54,27 %
Tiempo no utilizado	18,48 %	20,25 %	17,80 %	21,59 %	17,17 %	14,40 %	15,64 %	16,12 %
Rutas utilizadas	3,133	3,244	3,127	3,329	3,007	3,021	3,015	3,018

Cuadro 13: Simulacion Greedy con 1 y 2 rampas

	SIM 9	SIM 10	SIM 11	SIM 12
Distancia total (km)	417,48	424,866	415,598	434,3
Tiempo total (hr)	16,58	16,41	16,25	16,71
Volumen no utilizado	53,66 %	53,48 %	54,23 %	54,06 %
Tiempo no utilizado	14,99 %	15,91 %	16,67 %	14,31 %
Rutas utilizadas	3	3,003	3	3

Cuadro 14: Simulacion Greedy con 4 rampas

5.4. Greedy Aleatorio

	SIM 1	SIM 2	SIM 3	SIM 4	SIM 5	SIM 6	SIM 7	SIM 8
Distancia total (km)	987,401	1397,978	1744,488	1726,094	1000,49	1427,362	1710,417	1748,735
Tiempo total (hr)	26,87	37,2	47,41	46,62	27,91	38,12	46,84	47,31
Volumen no utilizado	84,09 %	84,58 %	85,21 %	84,84 %	82,91 %	84,13 %	84,84 %	84,87 %
Tiempo no utilizado	17,34 %	18,23 %	21,87 %	21,95 %	14,13 %	16,21 %	20,39 %	20,44 %
Rutas utilizadas	5	7	9,346	9,201	5	7	9,066	9,161

Cuadro 15: Simulacion Greedy Aleatorio con 1 y 2 rampas

	SIM 9	SIM 10	SIM 11	SIM 12
Distancia total (km)	1077,235	1480,395	1707,955	1739,629
Tiempo total (horas)	29,85	40,18	46,62	46,96
Volumen no utilizado	81,38 %	82,90 %	84,06 %	84,19 %
Tiempo no utilizado	8,16 %	11,70 %	17,00 %	17,28 %
Rutas utilizadas	5	7	8,654	8,746

Cuadro 16: Simulacion Greedy Aleatorio con 4 rampas

Como se puede observar en los resultados obtenidos de las varias simulaciones realizadas, para cada número de rampa (1, 2 o 4), la mejor cantidad de camiones es 5, 5, y 10 camiones respectivamente. Es decir, si se tiene una única rampa, se sugiere utilizar 5 camiones de entrega, para 2 rampas se sugiere 5, y para 4 rampas se sugieren 10 vehículos. Esto si se ve desde un punto de vista donde se intenta minimizar lo más posible ambas mediciones de distancia total recorrida y tiempo total invertido en todos los camiones para la entrega de productos. Sin embargo, es importante notar que conforme aumente la cantidad de vehículos utilizados en cada rampa, el volumen *desperdiciado* aumenta, siendo congruente con la idea de que una misma cantidad de productos se distribuyen entre los camiones sin que todos puedan llegar a su capacidad máxima de $11.5 m^3$. Por este lado, una menor cantidad de vehículos independientemente de la cantidad de rampas sería mejor si se quiere utilizar la capacidad de los vehículos al máximo. En el contexto del e-commerce de un negocio pequeño esto puede ser más relevante ya que utilizar más vehículos de entrega sin aprovechar su capacidad lo más posible sea un gasto innecesario, pero para negocios más grandes (como Amazon) pudieron priorizar minimizar el tiempo de entrega.

5.5. Página interactiva

Para poder ejemplificar diferentes casos y situaciones se realizó una página donde se pueden realizar diferentes rutas variando el número de simulaciones a realizar, número de grupos de clientes, tiempo máximo para completar la ruta, capacidad de carga máxima de camiones entre otras variables vistas en la Figura 8. Para lograr esto se usó la plataforma de Azure creando una imagen de Docker usando el algoritmo en Python y el *framework* Streamlit para su diseño.

En la página se pueden ingresar dichos datos y después de hacer clic en el botón de *Empezar Simulación* se despliegan las diferentes rutas generadas con los respectivos nodos. Añadiendo el tiempo estimado y la distancia recorrida. Dicha página se puede encontrar en el siguiente enlace:

<https://ma2004b-vrp-greedy.ambitiousground-972599b8.eastus.azurecontainerapps.io>

The image shows a dark-themed user interface for configuring simulations. It features several input fields with numerical values and minus/plus adjustment buttons. The fields are: 'Simulaciones a realizar:' with value 1, 'Número de muestreos a realizar de clientes:' with value 0, 'Cantidad de camiones:' with value 1, 'Tiempo máximo para completar la ruta:' with value 3600, 'Capacidad de carga máxima de los camiones:' with value 3.00, and 'Cantidad de rampas:' with value 1. Each field has a question mark icon to its right. At the bottom, there is a red button labeled 'Empezar Simulación'.

Figura 8: Página inicial para realizar diferentes simulaciones.

6. Conclusiones

El comercio electrónico ha experimentado un crecimiento sin precedentes en la última década, lo que ha llevado a una demanda cada vez mayor de una optimización en la generación de rutas de entrega. La implementación de nuevas tecnologías, como algoritmos de optimización de rutas, se ha convertido en una necesidad para las empresas que buscan mantenerse competitivas en este entorno. Estas soluciones mejoran la experiencia del cliente al ofrecer entregas más rápidas y precisas. En última instancia, la optimización de la generación de rutas de entrega se ha convertido en un pilar fundamental para el éxito continuo del comercio electrónico en la era digital.

Al tener las bases de una compañía de e-commerce, donde se necesita estimar el número de rampas, número de camiones y el modelo de los mismos, así como las posibles rutas, es necesario tomar en cuenta que hay factores cambiantes con regularidad, sin embargo estos se pueden prever y solucionar de manera más sencilla viendo los posibles resultados mediante la simulación. Esta planeación nos permite tener una base más sólida al momento de comenzar la distribución, teniendo claridad en los costos aproximados para la inversión inicial. Esta implementación también representa ahorros de tiempo, gasolina, emisiones, y económico, generando una situación más rentable en todos los aspectos.

Al realizar las diversas simulaciones, se concluyó que la mejor alternativa, tomando en cuenta las diferentes restricciones anteriormente mencionadas, es hacer uso de 4 camiones y 2 rampas dado que el porcentaje de tiempo no utilizado disminuye, con una diferencia de 2%, por lo que los conductores estarán utilizando casi la mayor parte de su horario laboral. Incluso bajando ligeramente el promedio del volumen no utilizado, por lo que se está repartiendo bien los productos.

Por último, cabe mencionar que tomando las 2 rampas la distancia promedio a recorrer es de 434.99 km haciendo un tiempo medio de 17.97 hrs con los 4 camiones como se indica en el Cuadro 10. Lo cual cumple con las restricciones antes establecidas, el de tiempo máximo y volumen máximo por camión. Es importante mencionar que en cada una de las simulaciones (con 1,000), el tiempo de ejecución computacional rondó entre 3 y 7 minutos, haciéndolo computacionalmente una tarea pesada pero razonable

para escalarlo a problemas más grandes.

7. Características de PC

Para poder desarrollar el proyecto y ejecutar dichos algoritmos los requisitos básicos que debe tener una computadora son los siguientes:

- **Sistema Operativo:** Windows 11 Pro
- **Procesador:** Ryzen 7 5800X, 8 núcleos, 16 hilos, con reloj base de 3.8GHz
- **Memoria RAM:** 16 GB, 3200 Mz
- **Almacenamiento:** 1 TB SSD
- **Software y herramientas utilizadas:** Python 3.10, Bing Maps API 2022, Azure Container Registry (2023-01-01), Azure Container Apps (2023-05-01), Streamlit 1.26.0

8. Anexos

1. Simulación: Código Python RETO Simulación 1000
2. Obtener Latitud y Longitud: Código Python RETO Obtener Latitud y Longitud
3. Obtener Intervalos de probabilidad de Demanda y Productos: Código Python RETO Obtener Probabilidad Demanda y Productos

Referencias

- [1] Alberto Carrillo, “El crecimiento del comercio electrónico en México,” 2020, accessed Aug 12, 2023. <https://www.eleconomista.com.mx/opinion/El-crecimiento-del-comercio-electronico-en-Mexico-20200927-0060.html>.
- [2] Carolina Soto Cota, “E-commerce y su crecimiento en México,” 2021, accessed Aug 12, 2023. <https://www.eleconomista.com.mx/opinion/E-commerce-y-su-crecimiento-en-Mexico-20210702-0045.html>.
- [3] Rodrigo Riquelme, “Comercio electrónico en México creció 232023,” accessed Aug 12, 2023. <https://www.eleconomista.com.mx/empresas/Comercio-electronico-en-Mexico-crecio-23-en-2022-20230209-0030.html>.
- [4] Carlo Enrico, “El efecto de covid-19 en el ecommerce,” 2020, accessed Aug 12, 2023. <https://www.forbes.com.mx/el-efecto-de-covid-19-en-el-ecommerce/>.
- [5] Asociación de Internet MX, “Tendencias de crecimiento del e-commerce en México 2020,” 2020, accessed Aug 12, 2023. <https://www.asociaciondeinternet.mx/tendencias-de-crecimiento-del-e-commerce-en-mexico-2020>.
- [6] Mónica Garduño, “E-commerce creció 5 veces en el primer trimestre de 2021,” 2021, accessed Aug 12, 2023. <https://www.forbes.com.mx/tecnologia-comercio-electronico-crece-5-veces-primer-trimestre-2021/>.
- [7] . S. K. Papadimitriou, C. H., *Combinatorial Optimization: Algorithms and Complexity*. New York: Prentice-Hall, 1982.
- [8] G. B. Dantzig, “Linear programming,” *Operations research*, vol. 50, no. 1, pp. 42–47, 2002.
- [9] Y. Yu, F. Lian, and Z. Yang, “Pricing of parcel locker service in urban logistics by a tsp model of last-mile delivery,” *Transport Policy*, vol. 114, pp. 206–214, 2021.
- [10] M. Ranathunga, A. Wijayanayake, and D. Niwunhella, “Solution approaches for combining first-mile pickup and last-mile delivery in an e-commerce logistic network: A systematic literature review,” in *2021 International Research Conference on Smart Computing and Systems Engineering (SCSE)*, vol. 4. IEEE, 2021, pp. 267–275.
- [11] Salazar, B., “Problema de enrutamiento de vehículos capacitados (cvrp),” 2023, accessed Aug 12, 2023. <https://www.ingenieriaindustrialonline.com/investigacion-de-operaciones/problema-de-enrutamiento-de-vehiculos-capacitados-cvrp-con-google-or-tools/>.
- [12] Amazon, “¿qué es la simulación monte carlo?” 2023, accessed Aug 12, 2023. <https://aws.amazon.com/es/what-is/monte-carlo-simulation/>.
- [13] O. Bräysy and M. Gendreau, “Vehicle routing problem with time windows, part i: Route construction and local search algorithms,” *Transportation Science*, vol. 39, pp. 104–118, 02 2005.
- [14] Uber atg: Research and development - simulation. Accessed: August 2, 2023. [Online]. Available: <https://www.uber.com/mx/es/atg/research-and-development/simulation/>

- [15] “Amazon prime air: Revolutionizing drone delivery with siemens,” <https://aws.amazon.com/es/partners/success/prime-air-siemens/>, accessed: August 2, 2023.
- [16] AIMMS, “Formulation: CVRP,” AIMMS How-To, n.d. [Online]. Available: <https://how-to.aimms.com/Articles/332/332-Formulation-CVRP.html>
- [17] Norma oficial mexicana. Accessed: August 30, 2023. [Online]. Available: https://www.dof.gob.mx/nota_detalle.php?codigo=5529381&fecha=28/06/2018#gsc.tab=0
- [18] Cómo calcular la capacidad de un camión. Accessed: August 2, 2023. [Online]. Available: <https://wellpack.org/es/como-calcular-la-capacidad-de-un-camion/>
- [19] Peugeot manager. [Online]. Available: https://www.peugeotstore.com.mx/data/FT_manager_v3_c.pdf
- [20] Peugeot manager. Accessed: August 2, 2023. [Online]. Available: <https://www.peugeotgalerias.com/vehiculos/profesionales/peugeot-manager/>
- [21] Greedy algorithm. [Online]. Available: <https://www.programiz.com/dsa/greedy-algorithm>
- [22] El viajante de comercio. [Online]. Available: <https://knuth.uca.es/moodle/mod/page/view.php?id=3417>