



Tecnológico de Monterrey

Materia:

Optimización estocástica

Análisis de puesta en uso de un sistema de entrega ecommerce - Entregable 3

Profesores:

Fernando Elizalde Ramírez

Rafael Muñoz Sánchez

Alumnos:

Ricardo Vargas Garduño A01026909

Gian Marco Innocenti A00834310

Alejandro Medrano Torres A00831829

Juan David Rivera A01026193

Esteban Pérez A00832329

10 de septiembre de 2023, Monterrey, N.L.

I. INVESTIGACIÓN SOBRE EL E-COMMERCE

I.1 Introducción a la problemática

En la actualidad, las compras en línea han crecido significativamente, especialmente durante la recesión, alcanzando un crecimiento del 20%. Sin embargo, se prevé que en 2023 el crecimiento será del 9% y se mantendrá durante algunos años. Debido a esta expansión, las empresas han tenido que rediseñar sus redes logísticas, adquirir más equipos y, lo que es más importante, ser capaces de entregar los pedidos a los clientes en las fechas previstas, ya que los retrasos generan mala reputación y, en consecuencia, pérdida de clientes. (Reto: Optimización Estocástica (Gpo 101), n.d.)

Justificación

Debido a estos cambios, se utilizan simulaciones de su proceso de entrega para planificar los recursos que necesitarán utilizar o adquirir para completar su trabajo de la mejor manera posible. En consecuencia, en el presente reto se propondrá un modelo matemático o computacional que permita simular la entrega de productos de una empresa de comercio electrónico en diversas circunstancias. (Reto: Optimización Estocástica (Gpo 101), n.d.)

Objetivo

Realización de un algoritmo que permita simular la logística de entrega de productos a distintos clientes de una empresa de comercio electrónico con el fin de reducir costes, acortar tiempos de desplazamiento y producir la menor cantidad de tráfico y contaminación dentro de las ciudades. (Reto: Optimización Estocástica (Gpo 101), n.d.)

I.2 Trabajo relacionado

Investigación sobre el problema del agente viajero centrado en el e-commerce

El problema del agente viajero (TSP, por sus siglas en inglés) es un problema clásico de optimización combinatoria que consiste en encontrar el recorrido más corto que pasa por un conjunto de ciudades y vuelve al punto de partida. Este problema tiene muchas aplicaciones prácticas, especialmente en el ámbito del comercio electrónico (e-commerce), donde se busca minimizar los costos y tiempos de entrega de los productos.

En el artículo de Chiung Moon, “*An efficient genetic algorithm for the traveling salesman problem with precedence constraints*” se propone un algoritmo genético mejorado para resolver el TSP con restricciones de precedencia, que es una variante del TSP que surge en el contexto del e-commerce. El algoritmo utiliza una codificación basada en permutaciones, una operación de cruce adaptativa, una operación de mutación basada en la inversión local, y un mecanismo de elitismo. Los resultados experimentales muestran que el algoritmo es capaz de obtener soluciones competitivas en comparación con otros métodos del estado del arte.

Investigar sobre la Simulación de Montecarlo

La Simulación de Montecarlo es un método estocástico cuyo propósito es poder predecir múltiples escenarios de una variable objetivo y la probabilidad de que ese escenario suceda, basado en información histórica de esa variable y otras variables independientes las cuales tienen impacto en la variable objetivo. (Amazon, n.d.) Algo curioso de la Simulación de Montecarlo es que entre más incrementan el número de entradas a la Simulación se obtendrán un mayor número de pronósticos, por lo que si se tiene suficiente información las predicciones a largo plazo serán muy certeras. (IBM, n.d.)

¿Cuántas corridas se tienen que hacer para lograr cierto nivel de eficiencia?

Para lograr un cierto nivel de eficiencia con la Simulación de Montecarlo hay varios factores para tomar en cuenta. Según Josep Maria Losilla de la Universidad Autónoma de Barcelona el número ideal de corridas depende del tipo de datos históricos con los que se está trabajando. Si está trabajando una investigación con datos independientes $\frac{n!}{n_1! n_2! \dots n_k!}$ “donde n simboliza el número de casos de la muestra, y n_i el número de efectivos en cada uno de los niveles de la variable independiente (k simboliza también el número de variables en el diseño de datos apareados) (Losilla, 1994,).

¿Qué ventajas y desventajas tiene el método?

Las ventajas que tiene la Simulación Montecarlo es que es un método muy directo y flexible, La simulación permite resolver problemas que no tienen solución analítica y da como resultado una aproximación muy certera. Finalmente la simulación permite formular condiciones extremas con riesgos nulos. Una de las desventajas de la simulación es que el método no genera soluciones óptimas globales. (Meneses, 2012)

Investigación sobre el problema de ruteo de vehículos

El problema de Ruteo de vehículos es un problema clásico de logística en el cual se busca optimizar una ruta de entrega a partir de vehículos donde se busca la mejor manera de distribuir vehículos para hacer entregas, minimizando costo de transporte, tiempo, entre lo que sea que busque la compañía. Al ser un problema importante ya hay bastantes formas de abordar la solución del problema, se puede usar ruteo dinámico o ruteo estático, aparte no es un problema único sino que también tiene bastantes variables, algunas de estas son:

- Estocástico
- Con recogidas y entregas
- Con múltiples centros de distribución
- Con ventanas de tiempo

Este problema es aplicable a la gran mayoría de industrias y es un problema en el cual constantemente se buscan nuevas maneras de mejorar el rendimiento cambiando diferentes parámetros. El aprender a solucionar este tipo de problemas es esencial para el aprendizaje de optimización y resolución de problemas de logística.

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

En el artículo “MODELO DE RUTEO DE VEHÍCULOS COMO ALTERNATIVA DE TRANSPORTE, ESTUDIO DE CASO: UMNG SEDE CAMPUS”, se expone una problemática donde el rápido crecimiento de la población académica en la Universidad Militar Nueva Granada y las desfavorables condiciones de movilidad que enfrentan los miembros de la comunidad que se trasladan a la Sede Campus Nueva Granada, fueron los motores que impulsaron el desarrollo del proyecto de investigación aplicando metodologías de solución a un problema de rutas CVRP-HF. Utilizando herramientas de software, principios de programación matemática y procesos heurísticos, se adhiere a una metodología que comienza con la recolección de datos y caracterización de variables y termina con un diagnóstico del escenario. Después, se implementa un modelo de ruta hacia la oficina y se valida mediante comparaciones con la simulación del sistema real. Como principal aportación, este artículo sugiere un proceso eficaz para el caso de estudio y la creación de un modelo de ruta, maximizando los beneficios, eliminando los obstáculos y aumentando el valor de los servicios universitarios. (Caicedo, 2018)

I.3 Definición del problema.

Debido al incremento de las compras por internet en los últimos años, se está buscando reorganizar el sistema de entrega de paquetes a domicilio de una empresa e-commerce. De manera que se puedan optimizar la cantidad de recursos disponibles y ajustar su planeación para que sean empleados de la manera más eficiente.

Para resolver este problema se necesitará generar un algoritmo que sea capaz de simular la logística de entrega de los productos a diferentes clientes en el mejor tiempo posible, con una disminución de costos relacionados y reducir el consumo de gasolina y la generación de tráfico, ayudando a reducir la emisión de contaminantes.

Para la elaboración de este algoritmo se tienen que tomar en cuenta varios factores que afectan el modelo. El horario de los clientes así como el de los conductores es un parámetro importante a considerar. Otros parámetros a definir serían la cantidad de camiones disponibles, la cantidad de rampas de carga disponibles y la capacidad de camiones. Los cuales tienen que ser trabajados de forma que no falte, ni que sobre demasiada capacidad. Es decir, evitar que falten recursos o que se tengan demasiados recursos sin usar.

El modelo tomará en cuenta a que lugares que tienen que visitar los camiones. Las variables que le dan forma al modelo serían principalmente si se fue o no a entregar a un cliente y desde donde está saliendo para llegar al siguiente nodo. Se puede entonces entender el modelo bajo las siguientes restricciones:

- Cantidad de horas disponibles.
- Cantidad de camiones disponibles.
- Horarios de entrega disponibles.
- Los camiones deben regresar al punto de inicio.
- Que todos los clientes reciban el pedido.

Tomando en cuenta las restricciones anteriores, la función objetivo sería la cantidad de lugares visitados por el usuario. Donde con base a las restricciones, el modelo debe de minimizar el tiempo de recorrido cumpliendo con todas las visitas.

II. CONSIDERACIÓN DE MODELO VRP

II.1 Modelo matemático

El modelo matemático que se aplicará al Problema de Enrutamiento de Vehículos (VRP), se enfoca en la simulación de múltiples escenarios para prever la demanda futura de productos y su cantidad por parte de los clientes. El propósito es descubrir la ruta óptima que minimice los tiempos de entrega y, por ende, los costos asociados. Al planificar una ruta eficiente, se busca la posibilidad de completar la entrega en un solo viaje, respetando las restricciones de tiempo laboral. Esta optimización no solo puede conducir a la economización de costos relacionados con salarios y combustible, sino también a la potencial reducción de las horas de trabajo necesarias.

La implementación de esta simulación se realiza a través del lenguaje de programación Python.

II.2 Definición de conjuntos

Parámetros

Para la elaboración de este algoritmo se tienen que tomar en cuenta varios factores que afectan el modelo.

- Horario de disponibilidad de camiones
- Horario de disponibilidad de clientes
- Cantidad de camiones disponibles
- Cantidad de rampas de carga disponibles
- Capacidad de los camiones

Los cuales tienen que ser trabajados de forma que no falte, ni que sobre demasiada capacidad. Es decir, evitar que falten recursos o que se tengan demasiados recursos sin usar.

Variables

Las variables que le dan forma al modelo serán principalmente si se fue o no a entregar a un cliente y desde donde está saliendo para llegar al siguiente nodo.

II.3 Planeación del modelo

Función objetivo

La función objetivo consiste en minimizar el tiempo total de los recorridos.

Restricciones

- Cantidad de camiones disponibles.
- Capacidad de volumen de cada camión
- Los trabajadores no deben sobrepasar la cantidad de horas de trabajo.
- Los camiones deben regresar al punto de inicio.
- Que todos los clientes reciban el pedido.

II.4 Simulación

Distribución de probabilidad de demanda de los productos.

Para determinar un aproximado de las probabilidades de demandar un producto determinado calculamos la frecuencia relativa de cada producto, cuya gráfica se encuentra en la figura 2.1. Posteriormente, calculamos los intervalos de probabilidad de cada producto al utilizar la frecuencia acumulada. Estos intervalos se utilizaron para sortear la cantidad de productos aleatoriamente que serán asignados a cada cliente en la simulación.



Figura 2.1. Frecuencia relativa de la demanda de diferentes tipos de productos ordenado de menor a mayor.

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

Para lograr simular que tipo o tipos de productos comprará cada cliente utilizaremos la frecuencia relativa de la demanda de cada cantidad de productos pedidos por un cliente. Esto lo hicimos calculando la frecuencia relativa para cada producto con los datos históricos dados por el Socio Formador. Para lograr determinar estas probabilidades y para tener un dataset funcional para la simulación que escogiera qué producto compra cada cliente, se necesito encontrar la frecuencia relativa, la frecuencia acumulada, y generar un límite superior que sería la frecuencia acumulada hasta ese producto, y un límite inferior que sería la frecuencia acumulada del producto anterior. En la figura 2.2 se puede apreciar un diagrama de barras de la frecuencia relativa de cada cantidad de pedidos realizados por un cliente.

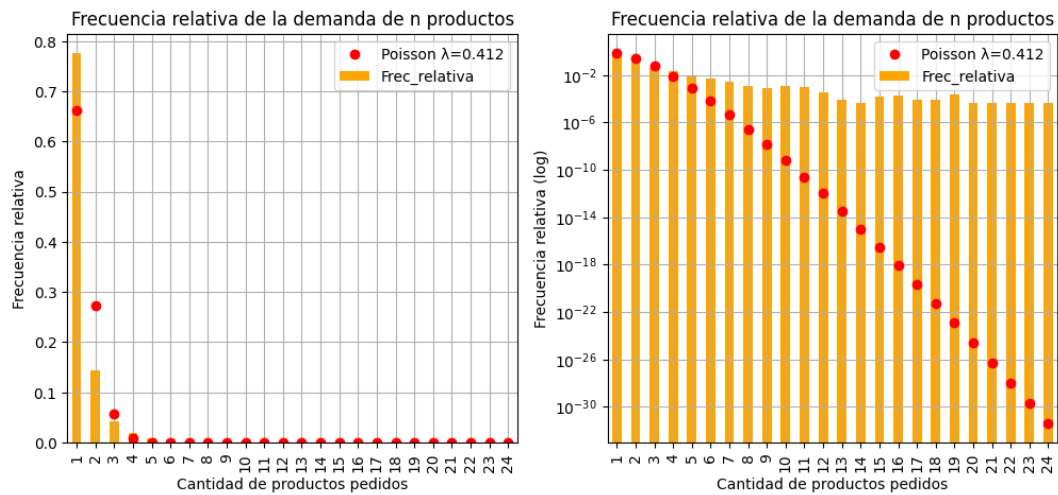


Figura 2.2. Frecuencia relativa de la demanda de n productos en comparación con la distribución de poisson con valor de λ igual a la media de los datos menos 1.

Al tener el dataset con la frecuencia relativa, sus intervalos para cada producto, y el número de productos que comprará cada cliente ya simulados. Se podrá simular el conjunto de productos comprado por cada cliente. Primero se generó una función en la cual se genera un número aleatorio entre 0 y 1. Después se verifica dentro de qué intervalo se encuentra ese número en los intervalos de frecuencia relativa de cada producto y se asigna ese producto a ese cliente. En el caso que se simuló que el cliente compra más de un producto se utilizó un for loop que itera ese número determinado de veces para cada cliente.

II.5 Método heurístico o metaheurístico a considerar

El método heurístico a utilizarse va a ser el cálculo de las distancias en línea recta entre cada punto, de igual manera se va a buscar la ruta entre los puntos con menor distancia entre línea recta, este sería el escenario óptimo, en el que te mueves de punto A al punto B en línea recta, ya que es la forma mas rapida de llegar. Lo cerca que esté la solución de este cálculo de distancias en línea recta, va a ser el indicador de que tan buena es la solución. Para hacer este cálculo se va a utilizar la fórmula del Semiverseno, la cual es:

$$d = 2r \arcsin \left(\sqrt{\sin^2\left(\frac{\phi_2 - \phi_1}{2}\right) + \cos(\phi_1) \cos(\phi_2) \sin^2\left(\frac{\lambda_2 - \lambda_1}{2}\right)} \right)$$

“Consiste en tomar dos puntos de una esfera y conociendo su radio, puedes calcular la distancia que hay entre cada extremo. Cada punto (P) está compuesto por una tupla (latitud, longitud) y ajustaremos R= 6371 (radio de la esfera/tierra)” (¿Como Hallar La Distancia En Kilómetros Entre Dos Coordenadas En Python?, n.d.)

II.6 Resultados preliminares

Para la simulación del ruteo de los vehículos mediante un VRP (Vehicle Routing Problem), se realizó un código en python donde se usa una matriz de tiempos dónde están los clientes y el tiempo de recorrido entre estos del uno al otro, para poder obtener las rutas de los vehículos, esto se hace mediante la selección del destino más cercano desde el punto en el que se está actualmente.

Lo primero que hace el código es crear una lista donde va a almacenar los destinos y su orden que se van a visitar, después lo que hace es revisar de la lista de destinos a visitar, cuál está más cercano a la bodega, una vez seleccionado este destino lo mete a la lista que representa la ruta del camión, procede a ordenar los demás utilizando la misma lógica, sin embargo para poder entrar a la lista tienen que respetar dos condiciones, la primera es que el tiempo de traslado del destino actual al nuevo destino, mas el tiempo acumulado, más el tiempo de regreso a la bodega desde el destino al que se quiere ir, no puede superar nuestro limite de tiempo laboral, la siguiente condición es que la suma del volumen del pedido al que se quiere visitar, más el volumen acumulado, no pueden superar la capacidad del camión, si se pasan estas condiciones, el destino es agregado a la ruta del camión, en el momento en el que un

destino no satisfaga estas condiciones, entonces todos los destinos que no fueron no visitados se guardan en una lista y el camión se cierra y ahí termina esa ruta y se crea una nueva donde se repite el proceso anteriormente mencionado hasta que ya no queden destinos por visitar.

Como se muestra en la figura 2.3, el código lo que regresa son dos cosas, las rutas para cada camión y una dataframe con las estadísticas de cada ruta, tiempo utilizado para la ruta, tiempo sobrante, espacio del camión sobrante.

```
Ruta 1
Depot ==> Cliente_5 ==> Cliente_3 ==> Depot

Ruta 2
Depot ==> Cliente_1 ==> Cliente_6 ==> Depot

Ruta 3
Depot ==> Cliente_4 ==> Depot

Ruta 4
Depot ==> Cliente_2 ==> Depot
```

Ruta	Tiempo total de la ruta	Espacio sobrante del camion	Tiempo sobrante de la ruta
1	87	25	393
2	154	4	326
3	216	0	264
4	287	26	193

Figura 2.3. Salida del código ejemplo que muestra la cantidad de rutas realizadas y estadísticas generales de los recorridos.

Esto nos va a permitir evaluar la solución, al igual que nos servirá para la experimentación y poder hacer conclusiones y recomendaciones sobre los diferentes parámetros a utilizar, como la cantidad de rampas óptima y el número de camiones óptimo.

III. EXPERIMENTACIÓN Y RESULTADOS

III.1 Características de la PC y software

- **Modelo de la PC** - MacBook Pro 13” chip M2
- **Sistema Operativo** - macOS
- **Capacidad de disco duro** - 256 GB
- **Memoria Ram y capacidad** - 8 GB
- **Tipo de procesador** - Apple M2
- **Número de núcleos** - 10
- **Software utilizado y versión** - macOS Ventura, version 13.3

III.2 Tamaño del problema

Número de nodos - 100 y 150 nodos (clientes, los nodos son los mismos, solo cambia la cantidad utilizada en cada iteración de la experimentación que fue llevada a cabo utilizando un Montecarlo)

Número de variables consideradas - son 6 variables las que nuestro modelo considera, la primera es la lista de clientes, la segunda es la lista de volúmenes de los pedidos de cada cliente, la tercera es la matriz de tiempos que describe los tiempos de traslado entre cada destino, la cuarta es la capacidad del camión, la quinta es lo que se tarda en cargar el camión y por último la cantidad de rampas.

```
Clientes=['Cliente_1','Cliente_2','Cliente_3','Cliente_4','Cliente_5','Cliente_6']
Volumen = [20,14,3,40,12,16]
MaT = pd.read_csv("Matriz_de_tiempo.csv",index_col=0)
capacidad = 40
TiempoCarga = 60
rampas = 1
Rutas, Rendimientos = VRP(Clientes,Volumen,MaT,capacidad,TiempoCarga,rampas)
```

Figura 3.1. Ejemplo de parámetros de entrada del código.

III.3 Experimentación

Para la experimentación y la simulación se hizo de la siguiente manera. Cabe recalcar que nuestro modelo VRP en cada iteración da como resultado la ruta óptima a seguir para cada camión, el tiempo que cada camión tuvo de sobra que no operó, el volumen faltante que tuvo cada camión, y el número óptimo de camiones utilizados para entregar todas las órdenes dentro del tiempo restringido.

Lo primero fue crear un dataset vacío donde se guardara el tiempo promedio que tomó para camiones hacer su ruta, el promedio del espacio sobrante de los camiones en la iteración, el promedio del tiempo sobrante de la ruta, el número de camiones utilizados en la iteración, el número de rampas utilizados en la iteración, y el número de clientes visitados en la iteración. En la simulación se realizaron 10,000 iteraciones con un grupo de 100 clientes y 10,000 iteraciones con un grupo de 150 clientes. Dentro las cuales en cada iteración se cambió el volumen total de la orden al extraer n clientes diferentes en cada iteración del dataframe de órdenes simuladas la cual contenía 24,000 registros de compras previamente simuladas. Esto se logró con la función de pandas `df.sample(i)` donde la semilla i incrementa por uno en cada iteración, así nos aseguramos de no repetir los volúmenes de la orden, también se le asignó un número de rampas aleatorias entre 1 y 5 rampas a cada iteración.

En cada iteración permanecieron constantes el tiempo de carga de cada camión de 60 minutos, y el volumen de cada camión de 10metros^3 . Al cambiar estos dos parámetros de entrada en el modelo VRP en cada iteración pudimos realizar la simulación Montecarlo con 20,000 iteraciones la cual tomó alrededor de 7 minutos.

Finalmente en cada iteración se fueron agregando los resultados al dataset vacío previamente mencionado para después lograr hacer un análisis de los resultados y poder brindar una buena recomendación. En el anexo 4 se puede encontrar el código con la implementación completa.

III.4 Resultados

Después de procesar los datos y pasarlos a través del VRP en la simulación antes mencionada, el resultado dado es el siguiente.

	Mean Tiempo total de la ruta	Mean Espacio sobrante del camion	Mean Tiempo sobrante de la ruta	Desviacion Tiempo total de la ruta	Desviacion Espacio sobrante del camion	Desviacion Tiempo sobrante de la ruta	Numero_de_Rampas	Numero_de_Camiones	Numero_de_Clientes
0	389.500000	4.171595	90.500000	67.175144	5.298514	67.175144	4	2	100
1	393.000000	1.925428	87.000000	120.208153	2.661977	120.208153	3	2	100
2	457.000000	4.017163	23.000000	19.798990	4.996230	19.798990	1	2	100
3	387.500000	4.490150	92.500000	130.814755	5.083340	130.814755	4	2	100
4	407.500000	2.456786	72.500000	34.648232	2.967576	34.648232	2	2	100
...
19995	364.000000	4.656000	116.000000	134.234869	4.903924	134.234869	1	3	150
19996	294.333333	5.308949	185.666667	147.967339	4.176312	147.967339	3	3	150
19997	284.333333	3.675691	195.666667	192.013888	5.418813	192.013888	3	3	150
19998	285.333333	3.186895	194.666667	67.002488	3.864703	67.002488	4	3	150
19999	384.000000	4.364347	96.000000	21.213203	5.442988	21.213203	5	2	150

20000 rows x 9 columns

Figura 3.2. Data frame con medidas del tiempo, espacio sobrante, rampas, camiones y clientes.

En la figura 3.2 tenemos un data frame donde se puede ver de cada iteración (1000 para un set con 100 clientes y 1000 para un set con 150 clientes) el promedio del tiempo total de la ruta, el espacio sobrante del camión, el tiempo sobrante de la ruta, la desviación de estas mismas, el número de rampas usadas, el número de camiones usados y el número de clientes. A partir de estos datos se puede obtener cuales iteraciones fueron las mejores a partir de obtener las que tengan el promedio de espacio sobrante y tiempo sobrante menor junto con el número de camiones menor. Tendríamos 2 iteraciones, una para 100 clientes y otra para 150 clientes. Ahora el procesamiento final para obtener los resultados que buscamos de los resultados en bruto.

Como lo que se quiere es el dar una recomendación de la solución del problema lo que vamos a hacer es dividir los resultados en los promedios de 100 clientes y 150 clientes y subsecuentemente obtener los promedios de las iteraciones de estos números de clientes para poder llegar a una propuesta de solución para la compañía. Al sacar los promedios estos son los resultados obtenidos

Promedios de las filas de 1 a 1000:									
	Mean Tiempo total de la ruta	Mean Espacio sobrante del camion	Mean Tiempo sobrante de la ruta	Desviacion Tiempo total de la ruta	Desviacion Espacio sobrante del camion	Desviacion Tiempo sobrante de la ruta	Numero_de_Rampas	Numero_de_Camiones	Numero_de_Clientes
promedio	370.39925	4.220531	109.60075	97.783562	4.224881	97.783562	3.038	2.37	100.0

Promedios de las filas de 1001 a 2000:									
	Mean Tiempo total de la ruta	Mean Espacio sobrante del camion	Mean Tiempo sobrante de la ruta	Desviacion Tiempo total de la ruta	Desviacion Espacio sobrante del camion	Desviacion Tiempo sobrante de la ruta	Numero_de_Rampas	Numero_de_Camiones	Numero_de_Clientes
promedio	370.632417	4.20167	109.367583	95.755451	4.273414	95.755451	3.068	2.361	100.0

Figura 3.3. Data frame con promedio y desviación estándar del tiempo, espacio sobrante, rampas, camiones y clientes del total de simulaciones realizadas.

Se puede observar en la figura 3.3 que para 100 clientes se recomienda el usar 3 camiones y 3 rampas, dado que es el número que más aparecía en las 1000 iteraciones lo cual significa que fue el caso más frecuente, también es posible estimar que el espacio promedio que sobra de los camiones es de 4.2 m^3 . Esto es debido a que puede que los paquetes excedan este número o que la última ruta de trabajo no se llena por completo dado que no hay clientes con pedidos pendientes.

Considerando esa cantidad de espacio desaprovechado, no es necesario contar con unidades de transporte con mucho espacio como el modelo “Manager Furgon L2H2 2021 Peugeot” que cuenta con la mayor cantidad de espacio (11.5 m^3). Se recomendaría centrarse en conseguir una unidad priorizando reducir la contaminación, en otras palabras, la unidad que tenga el mejor rendimiento Kilómetros por Litro en ciudad y que de preferencia sea de gasolina. El “Chasis Cabina 2020 Nissan”, de acuerdo con su ficha técnica, es de gasolina y rinde 10.16 Km/L en ciudad, que es una cantidad adecuada para este tipo de vehículos. La recomendación entonces sería conseguir un “Chasis Cabina 2020 Nissan” o un “Manager Furgon L2H2 2021 Peugeot”, el primero es mejor en cuestión ambiental, pero se necesitarán conseguir cajas secas para transporte, mientras que el segundo ya cuenta con cabina incluida.

También se puede notar que la ruta toma un tiempo de 370.399 minutos o aproximadamente 6 horas lo cual implica que el tiempo de repartición es este y las otras 2 horas pueden ser usadas para la recarga de combustible, comida, etc. Por lo que, se considera que es un buen tiempo en general. Para los 150 clientes los resultados son similares e implica que las recomendaciones son las mismas.

En conclusión el número idóneo de camiones y rampas a usar es 3 dado que se usan 2.36 camiones en promedio lo que implica que con 2 camiones no es suficiente ya que normalmente sobra, a la vez se ve que sobra la mitad del espacio en promedio por lo que se piensa que es debido al último camión que probablemente no se llena, sin embargo, la recomendación es emplear 3 camiones y también tener una rampa por camión.

IV. CONCLUSIONES

Luego de analizar los resultados obtenidos de las simulaciones podemos concluir que el mejor número de camiones de transporte que debe adquirir la empresa dadas las circunstancias dadas es de tres camiones, pues, en promedio, las simulaciones realizaban un total de tres recorridos con una desviación estándar muy pequeña.

Además, respecto a las rampas necesarias, el análisis de las simulaciones nos arroja que se recomienda tener una rampa por camión, es decir, tres rampas en total, ya que para poder respetar el tiempo de trabajo por día (8 hrs) y lograr satisfacer toda la demanda, es necesario que los 3 camiones inicien sus recorridos, incluyendo la carga de los paquetes, a primera hora.

Esta recomendación también es debido al tiempo sobrante de cada ruta, en promedio es de 110 minutos, por lo que si se quisieran usar menos camiones no alcanzaría el tiempo para hacer otra ruta ya que de inicio son 60 minutos de cargar el camión más el tiempo de recorrido lo cual no daría lugar a la entrega de muchos paquetes, pues el uso promedio de tiempo de cada ruta es 370 minutos, no daría para completar ni la mitad, es por esto que recomendamos 3 camiones y 3 rampas.

V. ANEXOS

ANEXO 1: Código de cálculo de distribuciones, pedidos de clientes y matriz de distancias

https://colab.research.google.com/drive/1AdNNKEysYvRiu_M0K1gs1E4-toKDYwO1?usp=sharing

ANEXO 2: Código de VRP

<https://colab.research.google.com/drive/11fg6pOgtygmFUKUml2p80Xo2ZdYwkP5k?usp=sharing>

ANEXO 3: Código de cálculo de heurística

<https://colab.research.google.com/drive/15S4Bioj4m9mfmrIQTeXMwnOvliG9ZDTE?usp=sharing>

ANEXO 4: Código de Simulación de Montecarlo

https://colab.research.google.com/drive/1B_aTYsQmoAnArLKFnDN9j5_Wz2sFsbMt?usp=sharing

ANEXO 5: Archivos requeridos para los códigos

<https://drive.google.com/drive/folders/1ZhmUYW8YoZPm8Y8RHIUwDaegbM0ha65s?usp=sharing>

VI. BIBLIOGRAFÍA

Descubre cómo abordar el problema de ruteo de vehículos (VRP). (2021). *Beetrack.com*. Retrieved August 16, 2023, from <https://www.beetrack.com/es/blog/problema-de-ruteo-de-vehiculos-vrp>

Kuo, M. (2023). Solving the Vehicle Routing Problem (2023). *Routific.com*. Retrieved August 16, 2023, from [https://blog.routific.com/blog/what-is-the-vehicle-routing-problem#:~:text=The%20Vehicle%20Routing%20Problem%20\(VRP,Problems%20\(TSP\)%20combined%20together.](https://blog.routific.com/blog/what-is-the-vehicle-routing-problem#:~:text=The%20Vehicle%20Routing%20Problem%20(VRP,Problems%20(TSP)%20combined%20together.)

Widuch, J. (2020). Current and emerging formulations and models of real-life rich vehicle routing problems. *Elsevier eBooks*, 1–35. Retrieved August 16, 2023, from <https://www.sciencedirect.com/topics/social-sciences/vehicle-routing-problem>

Reto: Optimización estocástica (Gpo 101). (n.d.). <https://experiencia21.tec.mx/courses/405971/pages/reto>

Caicedo, M. a. E. (2018). MODELO DE RUTEO DE VEHÍCULOS COMO ALTERNATIVA DE TRANSPORTE, ESTUDIO DE CASO: UMNG SEDE CAMPUS [1]. <https://www.redalyc.org/journal/6078/607866319006/html/>

Liang Gao, Xinyu Li y Erbao Cao. (2002). *An efficient genetic algorithm for the traveling salesman problem with precedence constraints*. European Journal of Operational Research. Retrieved August 16, 2023, from <https://www.sciencedirect.com/science/article/abs/pii/S0377221701002272>

Amazon. (n.d.). *¿Qué es la simulación de Monte Carlo? - Explicación de la simulación de Monte Carlo - AWS*. Amazon AWS. Retrieved August 15, 2023, from <https://aws.amazon.com/es/what-is/monte-carlo-simulation/>

IBM. (n.d.). *¿Qué es la simulación Monte Carlo?* IBM. Retrieved August 15, 2023, from <https://www.ibm.com/mx-es/topics/monte-carlo-simulation>

Losilla, J. M. (1994). *Herramientas para un laboratorio de estadística fundamentado en técnicas Monte Carlo*. Universidad Autónoma de Barcelona. JMLV 1de2.pdf

Meneses, A. (2012, August 27). *MÉTODO MONTECARLO, ORIGEN, VENTAJAS Y DESVENTAJAS*. Corporación Universitaria del Meta- Simulación. Retrieved August 16, 2023, from <http://unimeta-simulacion-alejandra-meneses.blogspot.com/2012/08/metodo-origen-el-de-montecarlo-un-no.html>

¿Cómo hallar la distancia en kilómetros entre dos coordenadas en python? (n.d.). Stack Overflow En Español. <https://es.stackoverflow.com/questions/511284/como-hallar-la-distancia-en-kilomestros-entre-dos-coordenadas-en-python>