

Optimización de rutas de transporte para Grupo Coppel

Adrián Landaverde Nava A01745052, Michael Steven Delgado Caicedo A01652281, Manuel Julio Romero Olvera A01752662, Sabrina Nicole Rodríguez Salgado A01745197, Cristian Gonzaga López A01745134, Naomi Padilla Mora A01745914, Profesoras: Diana Denys Jiménez Suro, Iliana Cristina Carrillo Ibarra

Resumen—Con el crecimiento acelerado del e-commerce en años recientes, se ha vuelto crítico para las empresas adaptarse al nuevo panorama. Cada vez es más común ordenar productos en línea, y las empresas en cuestión deben encontrar soluciones de logística adecuadas para procurar sus productos y hacerlos llegar a los clientes. Para lo anterior, la eficiencia y la optimización son clave, dado que las empresas buscan minimizar costos de distribución y tiempos de entrega, y al mismo tiempo cumplir con las expectativas de los clientes. Estas estrategias de logística involucran encontrar rutas óptimas, algo que puede estudiarse a partir de problemas como el Problema del Agente Viajero y el Problema de Enrutamiento de Vehículos con Capacidades, los cuales son aplicaciones de la Programación Lineal. El presente trabajo aplica los conceptos anteriores, así como algoritmos de Machine Learning, para resolver un problema de rutas a la empresa Grupo Coppel, desde uno de sus centros de distribución en Azcapotzalco, Ciudad de México.

Index Terms—TSP, Coppel, E-commerce, Optimización, Operaciones

I. INTRODUCCIÓN

Durante la última década el *e-commerce* ha aumentado, revolucionando las industrias, las economías y la forma de adquirir productos. Además, debido a su gran potencial económico, el *e-commerce* ha sido bien recibido, sobre todo, durante la pandemia causada por el COVID-19, permitiendo que las empresas se mantuvieran activas sin la necesidad de que sus clientes acudieran a sus tiendas físicas. Por ello, cientos de empresas nacionales e internacionales añadieron a su línea de servicios el *e-commerce* o *compras en línea*. Siendo Grupo Coppel una de estas empresas, enfrentado un gran reto por mantenerse a la vanguardia con este importante avance tecnológico.

Uno de los retos más relevantes tanto para Coppel como para muchas empresas en relación al *e-commerce* es la realización de entregas, ya que las entregas implican un capital humano, transporte, tiempo, combustible y seguridad, las entregas deben realizarse de la manera más eficiente para satisfacer al cliente y respetar los costos y utilidades. Es por ello que se tiene el objetivo de proponer una logística en las rutas de transporte para las entregas de Grupo Coppel del CEDIS de Azcapotzalco, CDMX de la manera más eficiente y optimizada posible. Permitiendo a Coppel satisfacer su demanda sin la necesidad de sufrir grandes pérdidas y sin malgastar recursos. Con este fin, se pudieron utilizar herramientas de la programación lineal, como es el problema del agente viajero y el enrutamiento de vehículos con capacidades, para modelar

sus posibles rutas. Al igual, para apoyar y hacer más eficientes y precisos los cálculos, se utilizó el intérprete *Python*.

II. DESCRIPCIÓN DE LA PROBLEMÁTICA

En todo el mundo, el valor total de ventas en e-commerce aumentó de 572 mil millones de dólares en 2010 a 4.29 billones de dólares en el año 2020 [1], lo anterior representa un crecimiento de más del 500 % en una década. En 2020, el comercio electrónico representó un 20 % del dinero gastado en ventas al por menor en todo el mundo [2]. En el año 2021, el e-commerce tuvo un crecimiento de 16.8 %, con ventas por 4.921 billones de dólares [3]. Es un hecho que el e-commerce se ha vuelto parte permanente del panorama comercial y empresas acostumbradas a puntos de venta tradicionales han tenido que adaptarse. Con la constante evolución de las tecnologías de la información y comunicación, el comercio electrónico seguirá creciendo.

Una parte vital del comercio electrónico es la logística del reparto de los productos [2]; un gran número de productos distintos debe fabricarse u obtenerse de alguna forma, almacenarse y enviarse a los clientes. El crecimiento tan rápido del e-commerce ha planteado un reto para las empresas en este aspecto. Una de las tendencias principales de reparto que se han generado en la última década es ofrecer envíos rápidos y gratis (o a precios accesibles), lo cual únicamente es posible implementando un modelo de inventario distribuido, para reducir costos de envío y transporte [4]. Por esto, es necesario que las empresas generen una red de distribución y transporte que haga posible lo anterior. Los envíos eficientes también son clave para retener clientes y aumentar el volumen de consumo de los mismos [4].

Otras tendencias de reparto han incluido la internacionalización de los envíos, lo que se ha hecho más accesible con la presencia de empresas de logística contratadas como terceros. La expansión de los negocios a otros países y continentes se ha vuelto más factible. Una tendencia más de reparto es la transparencia de la empresa con los clientes y que ambas partes estén al tanto de los costos de envío, las actividades y el progreso de los transportadores [4]. Apertura con los consumidores sobre posibles retrasos en el envío, sus costos, políticas de devolución, y sobre el estado y la ubicación de su orden son muy importantes [5].

Los elementos que deben considerarse al momento de diseñar una política de reparto son el tiempo que toman los envíos y los costos que involucran. Lo anterior implica

tomar en cuenta dónde se encuentra el inventario, así como la red de distribución y transporte que posee la empresa. Debe encontrarse una distribución adecuada del inventario y las rutas adecuadas para minimizar costos y el tiempo que tarda el envío. Se trata de encontrar la mejor manera de administrar los recursos para el reparto de los productos, para reducir costos para la empresa y mantener la satisfacción del cliente.

III. PROBLEMA DEL AGENTE VIAJERO

El Problema del Agente Viajero (PAV) o *Travelling Salesperson Problem (TSP)* se distingue por ser uno de los problemas de optimización combinatoria más estudiado, perteneciendo al conjunto de problemas denominados NP-completos [6]. Consiste en encontrar la ruta más corta que recorre un viajero, partiendo de una ciudad inicial para visitar un conjunto de N ciudades y regresar al punto de partida, visitando cada ciudad exactamente una vez [6] [7]. De tal modo que una de las aplicaciones más directas y frecuentes de dicho problema se aprecian en el campo de la logística, por ejemplo, en la planificación de ventas, rutas de turismo, reparto de correo, entre otros [6].

IV. PROBLEMA DE ENRUTAMIENTO DE VEHÍCULOS CON CAPACIDADES.

Como ya se había mencionado, este planteamiento del VRP surgió con el fin de describir el envío de gasolina en camiones, entre diferentes terminales un gran número de estaciones de servicio. Cuando el número de estaciones se hace más grande, las opciones de enrutamiento son mayores, por lo tanto dar una solución óptima era una gran problemática. Por ello, este algoritmo se enfoca en proporcionar una aproximación de la solución óptima. Se considera la capacidad de cada camión.

V. VARIABLES Y PARÁMETROS A TOMAR EN CUENTA PARA LA SOLUCIÓN DE UN CVRP.

El desarrollo del CVRP, con respecto al TSP, determina el número mínimo de vehículos para proveer un producto a un número determinado de clientes. Cada vehículo tiene capacidad C , y son requeridos, para satisfacer los n clientes, cada uno con una demanda no negativa [8].

$$\left\lceil \frac{d(s)}{C} \right\rceil$$

Otro caso es el de un agente viajero que debe recorrer de una ciudad de origen, pasando por cada una solo una vez, y regresando al origen.

$$\text{Min } v = \sum_{(i,j) \in A} c_{ij} x_{ij} \quad (1)$$

Siendo v el conjunto de ciudades, A el conjunto de caminos. Esto a su vez está sujeto a [8]:

$$\sum_{\{i:(i,j) \in A\}} x_{ij} = 1 \quad \forall j \in V \quad (2)$$

$$\sum_{\{j:(i,j) \in A\}} x_{ij} = 1 \quad \forall i \in V \quad (3)$$

$$\sum_{\{(i,j) \in A: i \in U: j \in (V-U)\}} x_{ij} \geq 1; \quad 2 \leq |U| \leq |V| - 2 \quad (4)$$

Este conjunto de restricciones nos indica que a partir de una ciudad sólo se puede pasar a una única ciudad, al igual que evita el surgimiento de subtours.

Teniendo ambas formulaciones, se puede expresar de la siguiente manera:

$$\text{Min } \sum_{i \in V} \sum_{j \in V} c_{ij} x_{ij} \quad (5)$$

sujeto a:

$$\sum_{i \in V} x_{ij} = 1 \quad \forall j \in V \setminus \{0\} \quad (6)$$

$$\sum_{j \in V} x_{ij} = 1 \quad \forall i \in V \setminus \{0\} \quad (7)$$

$$\sum_{i \in V} x_{i0} = K \quad (8)$$

$$\sum_{j \in V} x_{0j} = K \quad (9)$$

$$\sum_{i \notin S} \sum_{j \notin S} x_{ij} \geq r(S) \quad \forall S \subset V \setminus \{0\}, S \neq \emptyset \quad (10)$$

$$x_{ij} \in \{0, 1\} \quad \forall i, j \in V \quad (11)$$

Donde [8]:

- $V = \{v_0, v_1, \dots, v_n\}$ es el conjunto de vértices del gráfico donde v_0 corresponde al almacén
- C es la matriz de distancias o costos C_{ij} entre los clientes v_i y v_j
- K es el número de vehículos de capacidad C necesarios para cargar toda la demanda
- S es un subconjunto de vértices del grafo donde $S \subset V \setminus \{0\}$
- $r(S)$ es el número mínimo de vehículos necesarios para servir a todos los clientes en S

De esta manera, con el conocimiento previo del agente viajero, en conjunto con el enrutamiento de vehículos con capacidades, se puede tener un panorama más completo y su asociación con la solución de los puntos de entrega y el CEDIS.

Una vez teniendo en cuenta la teoría, hay que obtener la información necesaria para poder llevar a cabo los algoritmos, por lo que se pueden tomar en cuenta diferentes factores

En primer lugar, para calcular del viaje entre cada par de ubicaciones, se puede hacer de diferentes formas como:

- **Distancia Euclidiana:** De esta forma, se puede calcular la distancia 'directa' entre cada par de ubicaciones, sin embargo, esto no refleja una ruta real, ya que nunca es posible avanzar totalmente en línea recta, debido al

trazado de las calles en sí o por algún impedimento natural del camino.

- **Distancia de la ruta:** La forma más idónea, sería calcular la distancia de la ruta en sí, para así saber cuántos metros hay entre cada par de ubicaciones. Sin embargo, algunas rutas pueden tener un mayor congestionamiento, o ser vías rápidas, por lo que no refleja realmente el gasto para el vehículo
- **Tiempo de la ruta:** Una forma aún más clara, es calcular el tiempo que tarda cada una de las rutas entre cada par de ubicaciones, de esta forma, se conoce cuánto tardaría cada transporte al hacer un recorrido

Una vez teniendo en cuenta esto, se tendría que calcular el costo de cada viaje posible. Dada la configuración de este problema en particular, se contaría con 300^2 viajes posibles, es decir, habría que calcular el tiempo que tomaría hacer 90000 viajes, y esto tomando en cuenta sólo 300 días, ya que la configuración inicial marca cerca de 1000 ubicaciones por día, por lo que se obtendría un total de 1000000 de viajes a calcular. Dado que realizar esta tarea implica mucho coste computacional, es preciso realizar ciertas técnicas para hacer más eficiente este procedimiento

- **Separación por municipios y delegaciones:** Se pueden agrupar las ubicaciones en diferentes municipios, de manera que se restrinjan los viajes a un sólo municipio o alcaldía. De esta manera, se podría agrupar aproximadamente en 10 partes iguales, por lo que se calcularían aproximadamente 302×10 viajes, es decir, 9000.
- **Clustering:** Otra forma de calcular estos viajes, es agrupar las ubicaciones mediante técnicas de Machine Learning. Separar las ubicaciones por división política podría ser una decisión arbitraria, sin embargo, puede no ajustarse a los datos, ya que se pueden necesitar más grupos, o pueden no estar distribuidos uniformemente. Es por esto que al usar técnicas de agrupamiento (o clustering), se pueden agrupar de manera que las ubicaciones de cada grupo sigan un patrón detectable por la Inteligencia Artificial.
- **Pétalos:** Aunque realizar clusters es una buena herramienta, hay clusters que quedan muy alejados del CEDIS, y al momento de llegar a un cierto destino, han pasado por ubicaciones donde podrían haber realizado alguna entrega, o que al momento de su regreso al CEDIS vayan completamente vacíos, por lo que el viaje total se vuelve ineficiente. Por lo tanto, se podría optar por un agrupamiento en forma de “pétalos”, es decir, los camiones seguirían una ruta que haría entregas tanto en su ida como en su regreso al CEDIS, siguiendo una forma de “pétalo” en su recorrido. De esta forma se podría optimizar el viaje entre cada una de las ubicaciones, así como la ida y el regreso al CEDIS.

V-A. OR-Tools

Para la realización y desarrollo del algoritmo de rutas se utilizaron las siguientes herramientas del paquete OR-Tools de Google Developers: *Traveling Salesperson Problem (TPS)* [9], *Vehicle Routing Problem (VRP)* [10] y *Capacitated Vehicle Routing Problem (CVRP)* [11]

Comenzando por el TPS el cual consiste en buscar el recorrido óptimo para visitar múltiples ciudades pero con un solo viajero. Por esto último, nace el VRP, que se considera la generalización del TPS ya que este permite encontrar la ruta más óptima para varios vehículos en rutas aisladas, no solo para un sujeto. Gracias al algoritmo VRP nace el CVRP del cual, podemos obtener las rutas óptimas para varios vehículos pero para casos más reales ya que permite agregar una condición extra, que para los fines del reto, fue el volumen de cada envío y la capacidad del camión, también en unidades de volumen. Por lo tanto tenemos tres algoritmos

- TSP:
<https://developers.google.com/optimization/routing/tsp>
- VRP:
<https://developers.google.com/optimization/routing/vrp>
- CVRP:
<https://developers.google.com/optimization/routing/cvrp>

VI. UBICACIÓN DE LOS PUNTOS DE ENTREGA DEL DÍA MIÉRCOLES 27 DE ENERO

Se asignaron las ubicaciones del miércoles 27 de enero, mismas que contenían un aproximado de 1000 ubicaciones. Sin embargo, se restringieron a sólo 300 ubicaciones, mismas que fueron tomadas de manera aleatoria y confirmando que corresponden a direcciones reales.

VII. VISUALIZACIÓN DE LOS PUNTOS DE ENTREGA Y DEL CEDIS

Al usar la librería “Folium” de Python, fue posible visualizar los puntos de entrega seleccionados anteriormente y el CEDIS, en la figura 1, se muestra en amarillo la ubicación de cada uno de los puntos de entrega, y el marcador azul muestra la ubicación del CEDIS. Asimismo, las áreas azules muestran los municipios y alcaldías a los que el CEDIS de Azcapotzalco puede hacer entregas. Este mapa se puede observar con mayor detenimiento e interactuar con él en el siguiente link: <https://ma2001b-202-4-ubicaciones.netlify.app/>

VIII. METODOLOGÍA

1. A partir de la base de datos asignada, se seleccionaron las ubicaciones del día seleccionado. Con estas ubicaciones seleccionadas, se quitaron las ubicaciones que estuvieran duplicadas. Posteriormente, para obtener las coordenadas de cada ubicación, se usaron técnicas

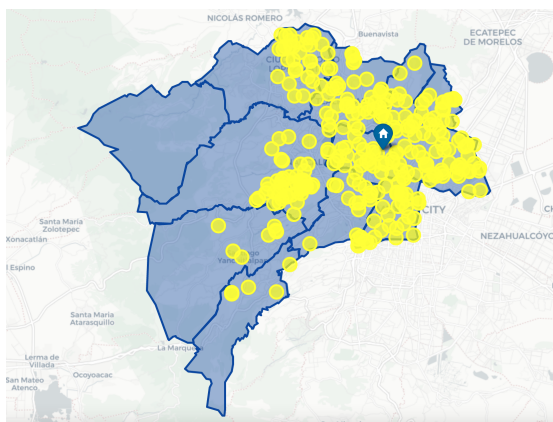


Figura 1. Puntos de entrega

de web scraping para esta búsqueda, es decir, con el apoyo de la librería de “Selenium” en el lenguaje de programación Python, se programó un bot que busca la dirección de cada ubicación en Google Maps, y guarda las coordenadas de cada uno de esos puntos en la base de datos. Para graficar las coordenadas en el mapa se usó una proyección WGS 84, para así poder adecuar la posición de los puntos sobre el mapa correspondiente. Esto se hizo con la misma librería antes mencionada “Folium”.

2. Una vez obtenidos estos datos con las ubicaciones, se seleccionaron sólo 300 de manera aleatoria, cuidando que estos datos hayan sido scrapeados, dicho de otra forma: extraídos, correctamente. Algunas direcciones no fueron ubicadas adecuadamente debido a que algunos nombres de dichas direcciones estaban incompletos, por lo que no correspondían a un lugar real o a la ubicación correcta. Por lo tanto, manualmente, se buscaron los datos que no coincidieran por completo con su dirección correspondiente.
3. Una vez obtenidas estas 300 ubicaciones aleatorias, se graficaron usando la librería “Folium” de Python, misma que graficó estos puntos en el mapa interactivo mostrado anteriormente. En amarillo se muestran las ubicaciones de cada punto de entrega, el marcador azul muestra la ubicación del CEDIS de Azcapotzalco, y las áreas sombreadas muestran los municipios y alcaldías a los que hace entrega este CEDIS.
4. Con el fin de obtener varias rutas de entrega y que no sólo 1 vehículo pase por todos los puntos, se decidió agrupar estos puntos por zonas. Para esto, se usó el algoritmo de Machine Learning no supervisado de agrupación K-means para encontrar estas agrupaciones. Estas se muestran en la figura 2, donde cada color muestra un grupo de entregas distinto.
5. Para poder usar el algoritmo de MEVRP, se debe saber el costo de los viajes entre cada punto, por lo tanto, se usó la API “OSMR” (Open Source Routing Machine), que calcula el tiempo que haría un vehículo entre 2 puntos determinados. Esto permite obtener las variables necesarias para usar posteriormente el algoritmo del

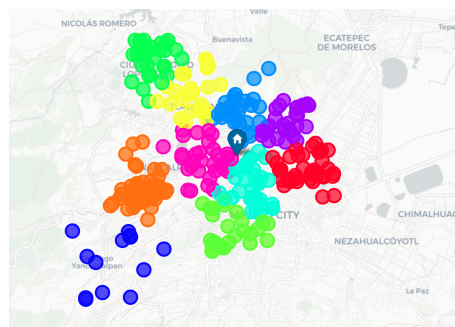


Figura 2. Agrupación por clusters

agente viajero.

6. Finalmente, se utilizó el algoritmo en Python, con el fin de trazar la ruta óptima dada por el CVRP [11]. A continuación se demostrará la solución sólo con 50 puntos en el mapa. Cabe notar que se convirtieron las unidades de volumen de m^3 a cm^3 , y las capacidades de los vehículos a usar en este caso, en cm^3 , son **5000000, 5000000, 10000000, 10000000 y 10000000**. El volumen de todas las entregas en este caso es de $34908321 cm^3$ y toma 20023 segundos hacer todo el viaje, o sea poco más de cinco horas y media. Esta ruta es una muestra, y nos enseña sólo un pequeño sector, dándonos como resultado el siguiente mapa (Figura 3), el cual nos muestra los puntos con su correspondiente numeración y el tiempo que transcurrirá el repartidor en recorrer ese camino. Cada color representa una ruta diferente. En la siguiente liga se puede interactuar y visualizar todos los datos que el mapa recopila: <https://ma2001b-202-4-mapeo50.netlify.app/>

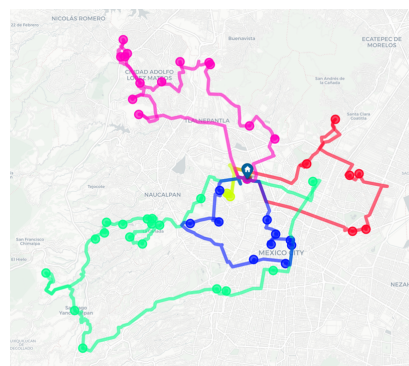


Figura 3. Ruta óptima dada por el algoritmo de CVRP en Python

IX. SOLUCIÓN

Para la solución de este problema, se tomaron en cuenta los 300 puntos a visitar, los tiempos de entrega de cada artículo (para llegar a esto también se consideraron las distancias entre cada par de puntos), el número de vehículos para cumplir con la demanda, y la capacidad de dichos vehículos ($27 m^3$). A continuación se muestran los resultados de las rutas óptimas para las 300 ubicaciones del 27 de enero, usando 6 vehículos:

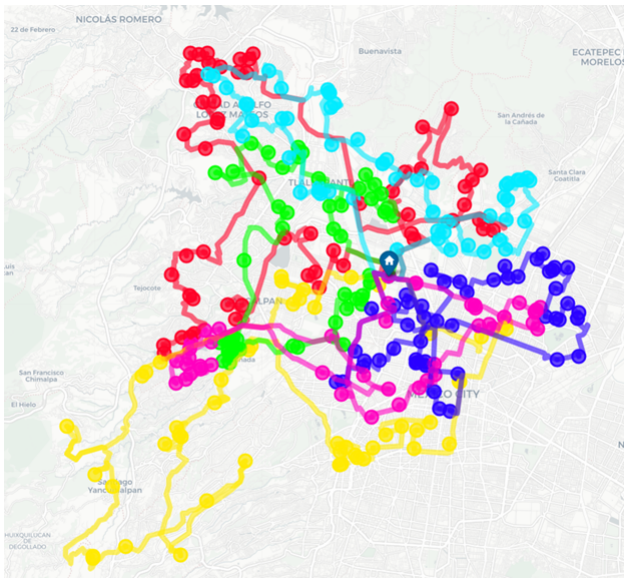


Figura 4. Rutas óptimas para los 300 puntos con 6 vehículos

En este caso, se usaron los 6 vehículos llenos casi en su totalidad, sin embargo, como se observa en la siguiente tabla, al tomar en cuenta un tiempo de entrega basado en el volumen del producto, cada recorrido tarda más de 10 horas en completarse, por lo tanto, no es una solución óptima aplicable debido a las jornadas de los trabajadores. Esto se debe a que, los camiones al ir llenos en su totalidad, tienen la capacidad de realizar muchas entregas, y por lo tanto, tardan mucho tiempo.

Camión	Carga (cm^3)	Tiempo en ruta	Paradas	Tiempo Total
1	26702304	221	59	700
2	25791948	204	52	668
3	26013793	144	51	588
4	24857385	164	55	618
5	26906803	140	51	645
6	26773390	119	44	514

Es por esto que, como una solución alternativa, se tienen también las rutas óptimas para los mismos puntos del día indicado, pero usando 8 vehículos.

En este caso, a diferencia del anterior, los camiones no van completamente llenos, sino que van casi en un 75 % de su capacidad máxima esto hace que sean más camiones pero que hacen menos paradas. Los datos de estos camiones se observan en la tabla siguiente:

Camión	Carga (cm^3)	Tiempo en ruta	Paradas	Tiempo Total
1	19884800	123	41	467
2	19751088	111	35	465
3	19718570	182	47	536
4	19795912	145	43	484
5	19658456	87	29	381
6	19710903	186	44	565
7	18778238	88	37	427
8	19747656	135	40	480

Por lo tanto, la solución final obtenida puede ser automa-

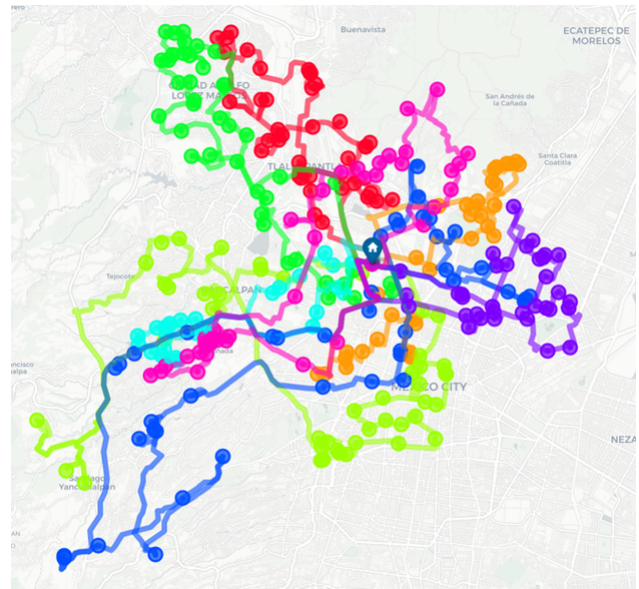


Figura 5. Rutas óptimas para los 300 puntos con 8 vehículos

tizada. Esto debido a que el código desarrollado toma una lista de coordenadas, calcula el tiempo entre cada par de puntos. Luego, usa el algoritmo de CRVP para calcular la ruta más óptima, donde se puede variar la cantidad de camiones, así como la capacidad de cada uno de ellos. Logrando así un modelo de enrutamiento eficiente y optimizado para las entregas requeridas y las restricciones señaladas. Con el código desarrollado se podría realizar una aplicación que permita aplicar soluciones solo ingresando los datos.

X. DETALLE DEL RECORRIDO EN LOS PUNTOS SELECCIONADOS

En la siguiente liga se puede encontrar una página web con los dos mapas antes presentados, figura 1 y figura 3, además de los últimos dos mapas, figura 4 y figura 5, que muestran dos posibles soluciones para la problemática de enrutamiento planteada.

<https://ma2001b.naomipadilla1.repl.co/>

Es posible interactuar con cada uno de los mapas, directamente en la página web del enlace anterior o en la versión maximizada de cada uno disponible en la página. Dentro de estas interacciones podemos realizar y encontrar lo siguiente. Al pasar el cursor sobre un camino, indica el tiempo que toma realizar ese tramo de la ruta (tiempo de enrutamiento). Si se pasa el cursor por uno de los círculos que representan las casas de los clientes, indicará el tiempo en el que se estima realizar la entrega. Al dar click sobre uno de estos puntos, se desplegará un cuadro de texto con la dirección. Además, para cada uno de estos casos, podemos encontrar el orden de ruta.

XI. LIMITANTES DEL RETO

Como se ha mencionado anteriormente, el problema de enrutamiento de vehículos con capacidades puede verse como

una extensión del problema del agente viajero, pero con mayor complejidad mientras más puntos deban visitarse [12]. Para este trabajo se toma en cuenta un número fijo de puntos de entrega en un sólo día, sin embargo, la realidad es que los puntos de entrega pueden cambiar de forma difícil de predecir. Estos cambios no sólo dependen del día y de la época del año, sino que puede haber también clientes regulares y clientes que sólo ordenen algo de la empresa una vez. Igualmente, no cabe duda de que en caso de que haya nuevos clientes, se agregarían nuevos puntos de entrega. Es particularmente necesario considerar que el número de clientes y, por lo tanto, el número de puntos de entrega podría aumentar en el futuro cercano, dado el crecimiento tan acelerado que ha tenido el e-commerce en la última década [1]. En este caso, sólo se tomaron en cuenta dichos puntos en un día específico. La solución óptima para las rutas de entrega entonces sólo es aplicable al día miércoles 27 de enero, con los 300 puntos de entrega seleccionados. Un cambio en algún punto de entrega resultaría en soluciones distintas al problema.

Lo anterior además supondría un problema si se habla de movilidad de último kilómetro, lo cual, en el ámbito de logística, es una gestión de transporte que se centra en el último tramo para realizar durante la entrega final, lo cual sirve como base para distribuir el resto de las entregas [13]. De manera similar, al hacer la agrupación por zonas con el algoritmo K-means, y de aquí determinar la ruta óptima para cada zona, la adición de un punto de entrega nuevo o algún cambio en los puntos existentes involucraría reconfigurar todo desde el principio.

Luego, no estamos al tanto de otras restricciones a las que estén sujetos los transportes, por ejemplo, en cuanto a tiempo o gasolina, y no podemos tomar en cuenta imprevistos, como tráfico. Si bien el problema de enrutamiento de vehículos con capacidades toma en cuenta la capacidad de éstos, falta más información en este aspecto. No hay forma de tener control total sobre la forma en la que se llena el vehículo. Por una parte, podría ser que quede espacio sin aprovechar dentro de un camión si éste únicamente tiene los productos a entregar en la ruta óptima, o puede suceder lo contrario, que un solo transporte no sea suficiente para entregar a todos los puntos pertenecientes a una ruta. Esto implica que no se aprovecharían al máximo los recursos disponibles o que habría que incurrir en costos adicionales. Por ejemplo, para el modelo de este problema se utilizaron los vehículos con caja Isuzu 300, los cuales tienen una capacidad de total para un volumen de 27 m^3 sin embargo, y suponiendo que el volumen total a entregar es de 15.7 m^3 , se desperdicia mucha de la capacidad del vehículo. Por lo tanto, sería más conveniente enviar vehículos con caja de menor capacidad, de lo contrario, el algoritmo daría como solución óptima aquella donde se aproveche en la medida de lo posible el espacio de la caja, sin poner una limitación sobre el máximo de horas que puede tomar un vehículo en completar una ruta de entrega. Siendo el caso actual, el algoritmo indica que la solución óptima para los puntos proporcionados debe ser realizada por un solo vehículo

de caja de 27 m^3 en un tiempo de 37744 segundos lo cual, viéndolo desde una perspectiva realista, no sería lo ideal para la empresa ni para el conductor.

Además, una de las limitantes prácticas que tienen los datos en sí, es que las direcciones están limitadas a una cantidad determinada de caracteres, por lo que algunas direcciones vienen acortadas, ocasionando que al momento de hacer el web scraping de cada dirección, no haya una dirección exacta qué buscar.

XII. PROPUESTAS DE MEJORA

Una de las posibles mejoras para este problema de enrutamiento sería poder incluir dentro de cada ruta tiempos de entrega más exactos. Los tiempos de entrega no forman parte como tal del tiempo que toman los traslados, pero finalmente, en un contexto real, el artículo debe entregarse físicamente al cliente, lo cual también consume tiempo de la ruta. Lo que se hizo para tomar en cuenta los tiempos de entrega fue asignar distintos tiempos dependiendo del volumen del artículo a entregar. Sin embargo, esto pudo haberse implementado de forma más específica para cada caso, lo cual hubiera hecho la solución más realista. Si se realizara un análisis de los tiempos de carga y descarga de cada producto en el CEDIS, se podría generar una base de datos que permitiera conocer con un poco más de precisión los tiempos de entrega. Otra posible mejora involucra la API utilizada para resolver el problema, en el caso de las soluciones anteriores se utilizó la API Open Source Routing Machine (OSMR). Esta API nos indica el tiempo que toma realizar el recorrido real de un punto a otro. Como lo haría un GPS o una aplicación móvil como Waze. Sin embargo, existe otra API diseñada por Google que resultaría más eficiente. La API Distance Matrix, también proporcionaría el tiempo de ruta pero de una manera más precisa, ya que esta API considera más variables para el cálculo y se está actualizando de manera constante. Sin embargo, al no ser software de código abierto, se requiere pagar por utilizarla y es por ello que se implementó la API OSMR a pesar de ser menos precisa. Adicional a esto, también se podría utilizar otra de las API de Google, Places, con la que se pueden obtener las coordenadas de un lugar con base en el nombre de la dirección. Esto hubiera hecho más eficiente el proceso de solución ya que habría ahorrado el tener que verificar cada una de las 300 direcciones a mano y obtener sus coordenadas.

XIII. CONCLUSIONES

En el presente trabajo, más que proponer una serie de rutas específicas para las entregas de Ruta Coppel, se ha querido proponer una metodología a seguir para resolver el problema de manera general, para diversos escenarios, diferentes zonas, puntos de entregas, o días. Una de las fortalezas principales del trabajo presentado es la automatización del proceso, desde la programación del bot para obtener las coordenadas de cada ubicación, hasta la graficación de los puntos de entrega y su agrupación por zonas con Machine Learning. Esta automatización y uso de diversas herramientas hizo más eficiente el trabajo. Por otra parte, un área de oportunidad para este

trabajo está relacionada con la falta de información sobre las instalaciones y operaciones de la empresa en general. Esto hubiera sido útil para conocer otro tipo de restricciones a las que estén sujetos los transportes en la empresa, por ejemplo, o para tomar en cuenta algún aspecto importante a la hora de proponer la metodología de enrutamiento anterior. Luego, hubo que hacer supuestos sobre los tiempos de entrega de cada artículo, o sea el tiempo consumido aparte de los traslados de un punto a otro. Ya que es algo difícil de determinar de forma fija, hubo que usar tiempos de entrega promedio de acuerdo con el volumen de cada producto. Lo anterior supone hacer generalizaciones, pero dado que no había información para este rubro, fue una medida necesaria para poder considerar esto como parte de nuestro problema de enrutamiento.

Finalmente, y en vista de los resultados anteriores, una de las recomendaciones que se le haría a Grupo Coppel sería usar vehículos de menor capacidad, de alrededor de 20 m^3 . Lo anterior sería para poder optimizar el uso del espacio al máximo en cada vehículo y, de esta forma, tener rutas que puedan recorrerse en ocho horas, incluyendo tiempos de entrega al cliente.

REFERENCIAS

- [1] "Global online sales reach nearly \$4.29 trillion in 2020," 2021. [Online]. Available: <https://www.digitalcommerce360.com/article/global-e-commerce-sales/>
- [2] "How e-commerce trends affect delivery and logistics," 2021. [Online]. Available: <https://www.hdi.global/infocenter/insights/2021/e-commerce-trends/>
- [3] "Global ecommerce forecast," 2021. [Online]. Available: <https://www.emarketer.com/content/global-e-commerce-forecast-2021>
- [4] "Ecommerce shipping and delivery: Statistics trends in 2021 to optimize your logistics strategy," 2022. [Online]. Available: <https://www.nosto.com/ecommerce-statistics/shipping-delivery/>
- [5] "Shipping trends: How to deliver on your customers' shipping expectations in 2022," 2022. [Online]. Available: <https://www.shopify.com/blog/shipping-trends>
- [6] F. Cota, "SOLUCIÓN AL PROBLEMA DEL AGENTE VIAJERO APLICANDO ALGORITMOS GENETICOS EN PROCESADORES GRAFICOS," Ph.D. dissertation, CENTRO DE INVESTIGACIÓN Y DESARROLLO DE TECNOLOGIA DIGITAL, INSTITUTO POLITECNICO NACIONAL, Dec 2013.
- [7] S. Ortiz, "SOLUCION DEL PROBLEMA DE MULTIPLES AGENTES VIAJEROS CON COMPUTO EVOLUTIVO EN PARALELOS," Ph.D. dissertation, CENTRO DE INVESTIGACION Y DESARROLLO DE TECNOLOGIA DIGITAL, INSTITUTO POLITECNICO NACIONAL, Dec 2016.
- [8] J. P. Orrego Cardoz, D. Ospina Toro, and E. M. Toro Ocampo, "Solucion al problema de ruteo de vehículos con capacidad limitada (cvrp) usando una técnica metaheurística," *Scientia Et Technica*, pp. 225 – 233, 09 2016. [Online]. Available: <https://www.redalyc.org/pdf/849/84950585004.pdf>
- [9] "Traveling salesperson problem," 2021. [Online]. Available: <https://developers.google.com/optimization/routing/tsp>
- [10] "Vehicle routing problem," 2021. [Online]. Available: <https://developers.google.com/optimization/routing/vrp>
- [11] "Capacity constraints," 2021. [Online]. Available: <https://developers.google.com/optimization/routing/cvrp>
- [12] "Traveling salesman problem (tsp) and the role of ai in solving it," 2020. [Online]. Available: <https://blog.locus.sh/travelling-salesman-problem-and-how-can-tech-solve-it/>
- [13] "La última milla: qué es y por qué es importante en logística," 2019. [Online]. Available: <https://www.alerce-group.com/es/nuestras-noticias/somos-noticia/la-ultima-milla-que-es-y-por-que-es-importante-en-logistica>