

MEMORIA DEL TRABAJO FIN DE GRADO

Análisis del tiempo de trayecto variable en un VRPTW.
Analysis of variable travel time in a VRPTW.

Autoría: D. Andreas Frank

Tutorizado por: D. Domingo Israel Cruz Báez

Grado en ECONOMÍA
FACULTAD DE ECONOMÍA, EMPRESA Y TURISMO
Curso Académico 2021/2022

San Cristóbal de La Laguna, 9 de septiembre de 2022.

RESUMEN

El análisis del VRP con capacidad limitada y ventanas de tiempo gira en torno al tiempo de trayecto en el proceso de optimización de rutas. Para hacerlo se ha definido dos tipos de modelos, uno de tiempo de trayecto constante y otro de tiempo de trayecto variable. El modelo de tiempo constante se ha formulado apoyado en otros modelos VRP y este se ha ampliado para poder incorporar el tiempo de trayecto como variable. Todo el análisis se ha contextualizado con un ejemplo en la isla de Tenerife, un depósito de gasolina ubicado en Santa Cruz de Tenerife, que deberá satisfacer la demanda de combustible a una serie de gasolineras denominadas clientes. El resultado indica que el tiempo constante tal como se ha definido no sustituye un análisis del tiempo variable, sin embargo, el tiempo variable ha mostrado tener un comportamiento que se puede aproximar usando la distancia.

Palabras clave: óptimo, tiempo, mínimo, ruta, restricciones, modelo.

ABSTRACT

The analysis of the VRP with limited capacity and time windows revolves around the travel time in the route optimization process. To do so, two types of models have been defined, a time-constant and a time-varying travel time model. The constant time model has been formulated supported on other VRP models and this has been extended to incorporate travel time as a variable. The whole analysis has been contextualized with an example on the island of Tenerife, a gasoline depot located in Santa Cruz de Tenerife, which must satisfy the demand for fuel to a number of gas stations called customers. The result indicates that the constant time as defined does not substitute an analysis of the variable time, however, the variable time has shown to have a behavior that can be approximated using the distance.

Key words: optimal, time, minimum, route, constraints, model.

ÍNDICE

INTRODUCCIÓN	3
1. REVISIÓN DE LA LITERATURA.	3
1.1 Modelización General.	3
1.2 Modelización de rutas de transporte.	4
1.3 El VRP con ventanas de tiempo.	5
1.4 Métodos de resolución.	6
2. PLANTEAMIENTO INICIAL.	8
3. FORMULACIÓN DEL MODELO VRPTW.	9
4. IMPLEMENTACIÓN DEL TIEMPO VARIABLE.	12
4.1 Análisis econométrico.	12
4.2 Formulación matemática.	14
5. BASE DE DATOS.	16
6. PROGRAMACIÓN MATEMÁTICA.	17
7. RESULTADOS.	21
7.1 Exposición de los resultados.	21
7.2 Defensa de los resultados.	23
8. CONCLUSIONES.	23
9. RECOMENDACIONES.	24
10. BIBLIOGRAFÍA.	25

ÍNDICE DE TABLAS Y FIGURAS

Imagen 1: Localización de los clientes y el depósito de gasolina.	8
Gráfico 1. Minutos de retraso a distintas horas del día para las 23 rutas consideradas.	13
Imagen 2. Resultado de la estimación por MCO del retraso.	13
Gráfico 2. Retraso observado y estimado con respecto a la hora en minutos.	14
Gráfico 3. Dispersión del tiempo de trayecto mínimo frente a la distancia.	16
Tabla 1. Resultados y comparación de la optimización de ambos modelos.	21
Gráfico 4. Resultado gráfico en distancias euclidianas del modelo 1.	21
Gráfico 5. Resultado gráfico en distancias euclidianas del modelo 2.	22

INTRODUCCIÓN.

La optimización es una de las disciplinas más relevantes dentro de la investigación operativa. La investigación operativa se define como la aplicación de métodos científicos para una asignación óptima de los recursos según los objetivos deseados, con el objetivo de ayudar en la toma de decisiones. Para ello, los profesionales de la investigación operativa aplican técnicas derivadas de la matemática, ingeniería y en general cualquier ciencia social, para recoger y analizar datos, desarrollar y probar modelos, y finalmente proponer soluciones y recomendaciones a través de la interpretación de los datos y los resultados. La optimización está dentro de estas técnicas y ha progresado muy rápido desde sus inicios a mediados del siglo XX.

En la actualidad la optimización tiene un papel muy importante en la mejora de la estructura empresarial. Por ello, se han ido desarrollando varios métodos para mejorar la optimización de redes lógicas, centradas sobre todo en el transporte de personas y mercancías como el VRP (Vehicle Routing Problem).

Dada la dificultad de definir un problema VRP fiel a la realidad, la calidad del resultado “óptimo” dependerá de la rigurosidad en la definición del propio problema y si ofrece información útil al decisor. La definición del problema VRP está en manos del analista, quien establece de manera adecuada las variables y la relación entre ellas a través de restricciones. Por tanto, la optimalidad del resultado dependerá mucho de cuanto se asemeja el comportamiento del modelo con la realidad, para que el resultado final sea fiable para la toma de decisiones. Para analizar cuanto depende el óptimo de una definición fiel a la realidad en un VRP, basta con cuestionarse si la optimalidad se mantiene si el costo del trayecto en términos de tiempo es variable. Si el análisis previo del costo de tiempo constante es lo suficientemente robusto, entonces la variabilidad del tiempo de trayecto no debería alterar excesivamente el resultado.

1. REVISIÓN DE LA LITERATURA.

1.1 Modelización general.

Un modelo es una representación matemática de una realidad compleja. Modelizar es la acción de construir un modelo representativo de la realidad. La mayoría de las veces, el desarrollo del modelo puede involucrar a un equipo multidisciplinario que aportan diferentes perspectivas y conocimientos en la descripción de la realidad. Principalmente un modelo es una herramienta que ayuda a la toma de decisiones. Por eso, sus resultados deben ser inteligibles y útiles. Modelizar es la combinación de la capacidad analítica y la materialización de la interpretación (Linares et al.,2001).

La investigación operativa (IO) es la ciencia que estudia las relaciones entre la realidad y la matemática con técnicas específicas que resuelve problemas de acercar los modelo a la realidad con el método científico. Los modelos son representaciones simplificadas de la realidad compuesto por los elementos controlables denominados variables y sus límites a través de restricciones y/o inecuaciones. La simplificación tiene como consecuencia que se distorsiona la imagen del modelo comparado con la realidad, por lo que requiere de una serie de fases de control para asegurar que las conclusiones extraídas se ajustan a la realidad. El proceso de control o

proceso de validación de un modelo tiene numerosas fases que contrastan el modelo con la realidad, y pueden resaltar relaciones que inicialmente no se habían considerado (Salazar, 2001).

La primera etapa en el desarrollo de un modelo es la identificación del problema, que consiste en la recolección y análisis de la información relevante. Los problemas reales suelen estar definidos en términos vagos e imprecisos, por lo que requiere la traducción en frases precisas convertibles en ecuaciones matemáticas. Esta etapa es fundamental para que las soluciones y las conclusiones sean válidas. La segunda etapa es la especificación y formulación matemática. En esta etapa, se define el problema en términos matemáticos del problema de optimización, definiendo sus variables, ecuaciones, función objetivo y parámetros. También se define su tamaño, estructura, restricciones y es muy importante la precisión en la formulación de las ecuaciones que describen el problema. La siguiente etapa es la resolución, establecer el procedimiento de resolución numérica que da lugar a la siguiente etapa, la verificación, validación y refinamiento. Ahora se eliminan errores en la codificación para conseguir que el modelo haga lo que se ha especificado previamente. Es necesario comprobar la validez de las simplificaciones realizadas a través de los resultados obtenidos y su coherencia. Por último, está la interpretación y análisis de los resultados. Esto permite conocer al detalle el comportamiento del modelo y hacer un análisis de sensibilidad en los parámetros, estudiar diferentes escenarios y comprobar la robustez de la solución óptima (Linares et al.,2001).

1.2 Modelización de rutas de transporte.

Este tipo de problemas recibe el nombre de VRP (Vehicle Routing Problem). Es el nombre que reciben aquella clase de problemas donde un conjunto de rutas para una serie de vehículos tiene como origen uno o varios depósitos, y deben atender a un número de clientes o ciudades. Su objetivo es minimizar el costo de las rutas que inician y terminan en un depósito para un conjunto de clientes con demandas conocidas, de tal forma que se atiende a cada cliente una vez por un solo vehículo (Muñoz&Sotero, 2009).

El VRP es una extensión del problema de m Agentes Viajeros o m -TSP, que a su vez es una generalización del TSP (Traveling Salesmann Problem) original o problema del Agente Viajero que solo contempla la creación de una sola ruta pasando por todas las ciudades. El m -TSP como su nombre indica son m Agentes Viajeros, es decir, se pueden crear m rutas que minimicen el coste de pasar por cada ciudad o cliente. El VRP se diferencia al m -TSP en que cada cliente o ciudad tiene asociado una demanda d_i y cada vehículo tiene una capacidad Q si la flota es homogénea (Olivera, 2004).

Según Muñoz y Sotero (2009) este tipo de problemas trabaja en las fronteras de un grafo $G(V, E)$ con una notación específica:

- $V = \{v_0, v_1, \dots, v_n\}$ es el conjunto de vértices o nodos donde v_0 se considera el depósito y $V' = \{v_1, v_2, \dots, v_n\}$ como el conjunto de ciudades o clientes.
- $A = \{(v_i, v_j) \mid \forall v_i, v_j \in V; i \neq j\}$ son el conjunto de arcos posibles entre los nodos.
- C es un matriz no negativa de costos o distancias entre cada cliente v_i y v_j .
- $D = \{d_1, d_2, \dots, d_n\}$ es un vector de demandas de los clientes.
- m es el número de vehículos. Una ruta es asignada a cada vehículo.

Cuando los costes $C_{ij} = C_{ji}$ para todos los (v_i, v_j) se dice que el problema es simétrico y es común reemplazar los arcos A por un conjunto $E = \{(v_i, v_j) \mid \forall v_i, v_j \in V; i < j\}$. Para mejorar el cómputo, se suele definir $b(V) = \lceil (\sum_{v_i \in V} d_i) / Q \rceil$ como límite inferior del número de vehículos necesarios para servir a todos los clientes dentro del conjunto V (Muñoz&Sotero, 2009).

Según Muñoz y Sotero (2009) en su forma más básica el problema se formula de la siguiente manera:

$$\min \sum_{i,j \in A} C_{ij} x_{ij} \quad (1.1)$$

$$s. a. \sum_{vj \in V'} x_{0j} = m \quad (1.2)$$

$$\sum_{vi \in V'} x_{i0} = m \quad (1.3)$$

$$\sum_{vi \in V} x_{ij} = 1 \quad \forall vj \in V \setminus \{0\} \quad (1.4)$$

$$\sum_{vj \in V} x_{ij} = 1 \quad \forall vi \in V \setminus \{0\} \quad (1.5)$$

$$\sum_{i \in S, j \in V \setminus S} x_{ij} \geq r(S) \quad \forall S \in V \setminus \{0\} \quad (1.6)$$

La función objetivo (1.1) es el coste total de la solución. Las restricciones (1.2) y (1.3) indican que m es la cantidad de vehículos utilizados en la solución y que todos los vehículos que parten del depósito deben regresar. Las restricciones (1.4) y (1.5) aseguran que todo cliente es un nodo intermedio de alguna ruta. Finalmente, la restricción (1.6) actúa eliminación de subtours y a la vez impone que la demanda total de los clientes visitados por un vehículo no puede superar la capacidad Q (Olivera, 2004). El concepto de subtours se refiere a bucles de solución que cumplen las restricciones, pero que no generan rutas coherentes con el propósito del problema. La restricción (1.6) requiere la resolución de un problema adicional para determinar el valor de $r(S)$, incrementando lo complejidad de la formulación del problema.

Este proceso se ha podido simplificar con la formulación MTZ, que se adaptó a este tipo de problemas (Pérez et al., 2011). En la propuesta de Miller, Tucker y Zemlin se crea una variable adicional que se acumula a lo largo de la ruta, pudiendo así establecer un valor máximo acumulado Q que no se puede superar, a la vez que asegura la eliminación de subtours.

1.3 El VRP con ventanas de tiempo.

En esta extensión del VRP se abrevia VRPTW (Vehicule Routing Problem with Time Windows) se mantienen las restricciones de capacidades, pero ahora cada cliente tiene asociado no solo una demanda sino también una ventana de tiempo que establece el intervalo (e_i, l_i) en el que debe ser atendido el cliente (Olivera, 2004).

La formulación matemática del VRPTW es similar a la de Muñoz y Sotero, pero hay que añadir una serie de restricciones que limiten una hora de llegada del vehículo. En la formulación de Olivera (2004) lo hace de la siguiente manera:

$$y_{jk} - y_{ik} \geq s_i + t_{ijk} - M(1 - x_{ijk}) \quad Ai, j \in A; k \in K \quad (2.1)$$

$$e_i \leq y_{ik} \leq l_i \quad Ai \in V'; k \in K \quad (2.2)$$

$$x_{ijk} \in \{0, 1\} \quad Ai, j \in A; k \in K \quad (2.3)$$

$$y_{ik} \geq 0 \quad Ai \in V; k \in K \quad (2.8)$$

Esta formulación del problema incrementa el número de subíndices de 2 a 3. El tercer subíndice surge al introducir un conjunto $k = (1, \dots, m)$ que indica el número de vehículos desde 1 hasta m , y se asigna a cada valor k todo el conjunto A . Para la formulación de la ventana de tiempo, añade una variable s_i continua para todo V' que indica la hora a la que se atiende al cliente i . Además, requiere de los parámetros e_i y l_i que indican la ventana de tiempo de cada cliente i , donde e_i es la hora mínima a la que se puede atender al cliente i y l_i la hora máxima. Las restricciones (2.1) define la ventana de tiempo que se tiene que cumplir, condicionado a que el arco asociado x_{ijk} desde el nodo v_i al nodo v_j del vehículo k debe estar activo. La restricción (2.2) indica que la hora de atención al cliente s_i tiene que estar dentro de la ventana de tiempo de este. Finalmente, las restricciones (2.3) y (2.4) indican la naturaleza de las variables x_{ijk} y y_{ik} .

La mayoría de las heurísticas VRPTW recientemente publicadas son enfoques de dos fases. En primer lugar, se utiliza una heurística de construcción para generar una solución inicial factible y solución inicial tan buena como sea posible. La calidad de un método de dos fases depende de si la elección de la heurística de construcción y mejora de la solución se ajusta a la naturaleza del espacio de búsqueda. La heurística de construcción debe producir una ubicación inicial lo suficientemente buena como para que el algoritmo de mejora comience en una región en la que se puedan alcanzar buenas soluciones. Posteriormente la heurística de mejora tendría que ser capaz de eludir un número suficiente de mínimos locales para llegar a una buena solución (Lau et al., 2003).

1.4 Métodos de resolución.

Los métodos se han ido diversificando a lo largo de los años, todos con el mismo propósito de solucionar el problema que comparten todos los problemas derivados del ruteo de vehículos. El principal problema al que se enfrenta todo problema de rutas es que el conjunto factible crece exponencialmente con el tamaño del problema, y aun hoy en día sigue siendo una tarea difícil desarrollar criterios y algoritmos para simplificar el problema, y con ello el tiempo de computación. A pesar de esta complejidad los métodos que se han ido desarrollando llegan a ser bastante efectivos, sobre todo para problemas grandes, pero generalmente con la desventaja de ser óptimos locales, muy buenos óptimos que pueden estar cerca del óptimo global pero que no tiene por qué ser óptimos del conjunto del problema.

Uno de los métodos más difundidos es el algoritmo de ahorros. Este trata de mejorar la solución inicial juntando rutas que inicialmente estaban separadas. El ahorro que supone juntar esas dos rutas es de $s_{ij} = c_{i0} + c_{0j} - c_{ij}$. Sin embargo, se ha observado que este algoritmo tiene tendencia a generar rutas circulares. Por ello varios autores han propuesto usar un parámetro de penalización denominado parámetro de forma $s_{ij} = c_{i0} + c_{0j} - \lambda \cdot c_{ij}$, que penaliza la unión de rutas de nodos lejanos. Las soluciones generadas por el método de ahorros por lo general pueden ser mejoradas con otro método de búsqueda local (Olivera, 2004). La desventaja de este método es que parte de una solución factible inicial, y no es muy efectivo para soluciones cercanas al óptimo.

Otro método es la inserción secuencial de Mole y Jameson. Este método trata de crear las rutas de manera secuencial que minimiza el coste. En otras palabras, se insertan clientes de manera ordenada en cada ruta, y cuando esta vuelva al origen se crea otra con los clientes restantes hasta visitar a cada uno. El principal problema de este método es que los últimos clientes, al ser los más lejanos, tienden a tener rutas muy ineficientes fruto de la asignación residual. Para remediar este problema se ha desarrollado el método de inserción en paralelo de Christofides, Mingozzi y Toth, que inicia de una primera fase de asignación secuencial, que luego se mejora en su segunda fase, maximizando la diferencia entre realizar la inserción en esa ruta y la segunda mejor para él (Olivera, 2004).

Otro de los métodos más útiles a la hora de simplificar un problema VPR es el método de asignación y ruteo. La base de este método en una asignación previa se asignan los clientes a un clúster. Estos clústeres serán la combinación de clientes que respeta la restricción de capacidad sobre los que trabaja el VRP. Simplifica el problema en el sentido de que crea pequeños subconjuntos de una sola ruta que luego se optimizan por separado en la resolución del problema original. Por ello, estos métodos tratan de aproximar la asignación a la del óptimo del problema desde distintas perspectivas. Un primer método de asignación y ruteo es el método de barrido o Sweep, que trata de dividir el conjunto en clústeres de manera que las rutas no se cruzan, una mecánica que puede ser útil en algunos casos. Otro es el método de asignación generalizada de Fisher y Jaikumar, que proponen resolver un problema de asignación generalizada con k clientes semilla sobre los que se construirán los clústeres. El método de Localización de Bramel y Simchi-Levi que se parece mucho a este último, pero se diferencian en que el método de localización determina los clientes semilla. Esto lo hace a través de clientes concentrador, como centro del clúster y clientes terminal que serán los que se asignen a cada clúster. Sin embargo, esta variante implica mucho cálculo de los valores de costos del conector y de insertar un terminal por lo que fue simplificado con las propuestas STH (Seed Tour Heuristic) y SCH (Star Connection Heuristic). Este último, define el coste como ir y volver del nodo, generando asignaciones en forma de estrella que, bajo algunas hipótesis, se prueba que es una asignación asintóticamente óptima (Olivera, 2004).

Otro procedimiento bastante conocido es el de búsqueda local. Este parte de una solución que se intenta mejorar con algún procedimiento de búsqueda local. Uno de ellos es el operador de λ -intercambios. Es de los operadores de búsqueda local más conocidos. Este procedimiento trata de eliminar λ arcos de la solución inicial, y sustituirlo por otro arco de manera que mejora el resultado. Una solución se denomina λ -óptima si no puede ser mejorado más de λ -intercambios. Una desventaja de este procedimiento es que suele invertir el orden de algunas rutas, que bajo

supuesto de costos simétricos no afecta mucho, pero en el caso contrario puede incrementar mucho el tiempo y costo computacional (Olivera, 2004).

Finalmente, de entre los muchos otros métodos restantes está el más básico, el método exacto. Dada la complejidad de estos problemas, este método este limitado a un número reducido de clientes. Pero se pueden resolver con algún método de relajación del problema. La relajación se refiere a la eliminación de alguna de las restricciones más complejas para facilitar la resolución, que le servirá como guía al problema inicial y se resuelve junto con algún esquema de ramificación y acotamiento (Olivera, 2004).

2. PLANTEAMIENTO INICIAL.

Para analizar en profundidad la influencia de la formulación del tiempo dentro de un VRP, primero hay que plantear una situación que requiera de este tipo de proceso de optimización. En este caso girará en torno al transporte y entrega de mercancías de un depósito de combustible. Este depósito de combustible se ha ubicado en la provincia de Santa Cruz de Tenerife, y su objeto será suministrar gasoil a distintas gasolineras de la isla de Tenerife, que serán los clientes. Por tanto, el problema a optimizar será un depósito de combustible en Santa Cruz de Tenerife que quiere optimizar la entrega de su mercancía a los clientes ubicados en la isla de Tenerife.

El origen por la naturaleza del problema será el depósito de combustible en Santa Cruz de Tenerife. Los clientes a los que tendrá que atender serán gasolineras cuyo número se extenderá a una por municipio, es decir, un total de 31 clientes. La imagen 1 muestra la distribución geográfica de los clientes seleccionados y el depósito.

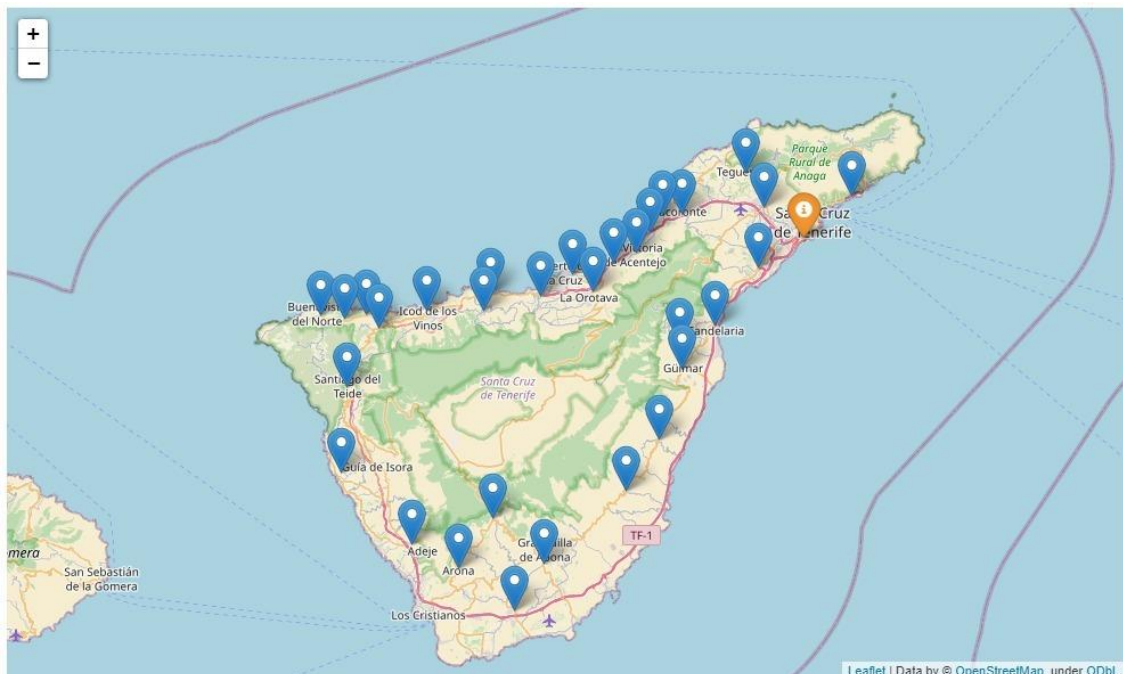


Imagen 1: Localización de los clientes y el depósito de gasolina. Fuente: Elaboración Propia vía Python.

Una vez descrita la situación hay que matizar algunos aspectos para plantear bien el problema. En primer lugar, los clientes tendrán un nivel de demanda determinado previamente que hay que satisfacer al completo sin sobrepasar la capacidad del vehículo. En segundo lugar, esta demanda se tendrá que satisfacer en un margen de tiempo establecido. Con esta información se trata de un VRP de capacidad limitada y ventana de tiempo (VRPTW).

3. FORMULACIÓN DEL MODELO VRPTW.

Definido concretamente el contexto del problema que se va a optimizar, hay que pasar a su formulación matemática. El modelo 1 trabajará en las fronteras del grafo $G(V, A)$, siendo los conjuntos asociados a las variables y parámetros del problema como sigue:

- $N = [v_1, \dots, v_n]$ será el conjunto de clientes que forman la red, que se extenderá desde el 1 hasta el enésimo cliente.
- $V = [v_0, v_1, \dots, v_n]$ representa el número total de nodos dentro de la red, asociando el v_0 al nodo origen, es decir, el depósito de combustible, al que le sigue el conjunto de clientes N .
- $A = [(i, j) \in V \times V, i \neq j]$ es el conjunto de arcos posibles dentro de la red desde cualquier nodo v_i a otro nodo v_j que pertenezcan al conjunto V .
- $E = [(i, j) \in V \times V, i < j]$ como el conjunto de conexiones posibles entre los nodos sin importar el sentido.

Definido el conjunto de índices hay que pasar a definir los distintos parámetros necesarios para el problema:

- c_{ij} es el coste de viaje de ir del nodo v_i al nodo v_j para cualquier arco dentro de V .
- $D = [d_0, d_1, d_2, \dots, d_n]$ es un vector de demandas que define el nivel de demanda asociado a cada nodo v_i dentro del conjunto V . La demanda del nodo origen d_0 es nula.
- $Sv = [sv_0, sv_1, sv_2, \dots, sv_n]$ es el vector tiempos de servicio necesario para cada nodo v_i dentro V . El tiempo de servicio para el nodo origen sv_0 es igual a cero.
- $[a_i, b_i]$ forman los límites temporales de la ventana de tiempo para cada v_i dentro de V , siendo a_i el límite inferior de la ventana de tiempo y b_i el límite superior.
- q define la capacidad de carga máxima del vehículo.

Ahora hay que establecer algunos supuestos sobre el comportamiento y tratamiento de datos. En primer lugar, se trata de un problema simétrico, por tanto, los arcos son perfectamente invertibles en términos de distancia y tiempo, es decir, $c_{ij} = c_{ji}$ para agilizar la recolección de datos. En segundo lugar, los vehículos serán capaces de realizar el trayecto establecido en cada arco tanto en términos de tipo de ruta como en tiempo para completar el trayecto, es decir, las rutas son aptas para este tipo de transporte y el vehículo es capaz de completar el trayecto en el tiempo establecido. Este supuesto es principalmente necesario ya que es difícil comprobar y/o ajustar el tiempo y la distancia empíricamente para cada arco.

Ahora hay que definir las variables de decisión que necesitaremos en el modelo:

- $x_{ij} \in [0, 1]$ es una variable binaria de decisión asociada a cada arco posible dentro del conjunto A , que tomará valor 1 si se usa ese arco y valor 0 en el caso contrario.
- $s_i \in \mathbb{Z}^+$ es una variable entera positiva que establece la hora a la que se atiende al cliente v_i que será necesaria para definir las restricciones de la ventana de tiempo.
- $u_i \in \mathbb{Z}^+$ también es una variable entera positiva que determina el nivel de carga con el que llega el vehículo al nodo. Es una variable de apoyo derivada de la formulación MTZ, que ayuda a limitar el nivel de carga máxima en una ruta.

Una vez definidos todos los elementos necesarios hay que pasar a la formulación matemática del problema de programación lineal entera (PLI):

$$\min \sum_{i,j \in A} c_{ij} x_{ij} \quad (3.1)$$

$$\text{s. a.} \quad \sum_{j \in N \setminus \{0,j\} \in A} x_{0j} \geq \sum_{i \in N} d_i / q \quad (3.2)$$

$$\sum_{i \in N \setminus \{i,0\} \in A} x_{i0} - \sum_{j \in N \setminus \{0,j\} \in A} x_{0j} = 0 \quad (3.3)$$

$$\sum_{j \in N \setminus \{i,j\} \in A} x_{ij} = 1 \quad \forall i \in N \quad (3.4)$$

$$\sum_{j \in N \setminus \{j,i\} \in A} x_{ji} = \sum_{j \in N \setminus \{i,j\} \in A} x_{ij} \quad \forall i \in N \quad (3.5)$$

$$x_{ij} + x_{ji} \leq 1 \quad \forall i, j \in E \quad (3.6)$$

$$u_j - u_i \geq d_j - \text{bigM}(1 - x_{ij}) \quad \forall i, j \in A; i \neq 0; j \neq 0 \quad (3.7)$$

$$d_i \leq u_i \leq q \quad \forall i \in N \quad (3.8)$$

$$s_j - a_0 \geq c_{0j} \quad \forall j \in N \quad (3.9)$$

$$s_j - s_i \geq sv_i + c_{ij} - \text{bigM}(1 - x_{ij}) \quad \forall i, j \in A; j \neq 0; i \neq 0 \quad (3.10)$$

$$b_0 - s_i \geq sv_i + c_{i0} \quad \forall i \in N \quad (3.11)$$

$$a_i \leq s_i \leq b_i \quad \forall i \in N \quad (3.12)$$

$$x_{ij} \in \{0, 1\} \quad \forall i, j \in A \quad (4.13)$$

$$s_i \geq 0 \quad \forall i \in V \quad (3.14)$$

$$u_i \geq 0 \quad \forall i \in V \quad (3.15)$$

La ecuación (3.1) representa la función objetivo del problema, que será la minimización de la suma del coste c_{ij} en términos de tiempo por la variable de decisión x_{ij} asociada al mismo arco (v_i, v_j) para minimizar la suma del tiempo de los arcos activos en la ruta óptima. Esta función estará sujeta a una serie de restricciones, que serán de 3 tipos: estructura de la ruta, capacidad y ventana de tiempo.

Las restricciones (3.2), (3.3), (3.4), (3.5) y (3.6) definen la estructura de la ruta. La (3.2) relaciona el número de arcos que salen del origen con el tamaño de la flota de vehículos, haciendo que tenga que ser como mínimo el cociente entre la demanda total y la capacidad del vehículo. Esta restricción limita el número de rutas que se tienen realizar al mismo tiempo, obligando al problema a satisfacer todas las demandas con el tamaño de flota indicado o más. La restricción (3.3) relaciona el número de arcos desde el origen con el número de arcos al origen igual a cero, en otras palabras, toda ruta que sale del origen tiene que volver a él. La restricción (3.4) limita el número de arcos activos con el mismo origen igual a 1, es decir, impide que el vehículo se quede en un cliente, a la vez que obliga que se le visite al menos una. La (3.5) indica que todo arco hacia un nodo debe tener su contrapartida de salida desde ese nodo en el conjunto de clientes. La restricción (3.6) relaciona cada arco con su inverso dentro del conjunto E igual o menor que uno, asegurando que solo puede estar activo en un solo sentido. Esta restricción a resultado muy útil para mejorar el tiempo de computación.

Las restricciones (3.7) y (3.8) limitan la demanda de la ruta al nivel de capacidad del vehículo. La restricción (3.7) relaciona el nivel de carga del vehículo desde el nodo origen con el nodo destino con la formulación MTZ. Está condicionado a una M-Grande, por tanto, el valor se acumulará necesariamente solo si el arco está activo. Su función es relacionar el nivel de carga del vehículo q desde el inicio hasta el final de la ruta, sumando al nivel de demanda acumulado del origen u_i con la demanda del destino d_j , que será igual al nivel de carga necesario hasta el cliente u_j . Esta formulación no solo permite controlar la carga acumulada a lo largo de la ruta sino también permite eliminar la generación de subtours. La restricción (3.8) completa la anterior limitando el nivel de carga en la ruta en sus dos extremos. Cualquier u_i tendrá que ser como mínimo el propio nivel de demanda del cliente v_i asociado y siempre menor o igual que q , es decir, la carga máxima de un vehículo. Esta restricción establece el rango de valores de u_i , lo que asegura que el nivel de carga de una ruta sea suficiente para satisfacer la demanda y no supere la capacidad máxima del vehículo.

Finalmente, las restricciones (3.9), (3.10), (3.11) y (3.12) son las restricciones de la ventana de tiempo asociado a cada nodo. La (3.9) obliga a que el tiempo de atención del nodo s_i no puede ser menor que el límite inferior de la ventana de tiempo del origen más el tiempo de trayecto a ese cliente. La restricción (3.10) relaciona el momento de atención de un cliente s_i con el de cualquier destino s_j salvo el origen. Por tanto, la suma de la hora de atención del nodo anterior s_i , su tiempo de servicio sv_i y el costo de trayecto c_{ij} al siguiente nodo, se tiene que compensar con la hora de atención del siguiente nodo s_j , condicionado a una M-Grande, es decir, no se tiene que cumplir esta restricción si el arco no está activo. La restricción (3.11) asegura que la ruta se completa dentro del margen de tiempo que establece el origen, es decir, que la suma del momento de atención del último cliente s_i , el tiempo de servicio sv_i y el costo de trayecto al origen c_{i0} sea menor que el límite superior de la ventana de tiempo del origen. Por último, la restricción (3.12) asegura

que el momento de atención de cualquier nodo s_i esté dentro de su ventana de tiempo establecida. Las restricciones (3.13), (3.14) y (3.15) indican la naturaleza de las variables.

4. IMPLEMENTACIÓN DEL TIEMPO VARIABLE.

4.1 Análisis econométrico.

Para poder incorporar el coste del trayecto en tiempo primero hay que saber cómo se comporta. Como el intervalo de tiempo de la ruta será desde las 12:00 hasta las 20:00, habrá que analizar como varía el tiempo de trayecto en ese intervalo del día. Como esto no se puede hacer para cada arco habrá que recurrir la estadística para estimar una función de comportamiento del tiempo de trayecto que se aplicará después a cada arco.

La muestra para estimar el comportamiento será una selección de las rutas entre los 31 clientes y el depósito. Las rutas se escogen de manera uniforme en el plano de distancias, con un intervalo de 5 kilómetros entre ellos, salvo del kilómetro 0, que será la ruta más pequeña, sumando en total 23 rutas. Luego se analiza el comportamiento del tiempo de trayecto a lo largo de cada ruta en un intervalo de 15 minutos en las 8 horas consideradas. Esto dio una muestra de 759 datos de tiempo de trayecto relacionados con la distancia del trayecto y la hora del día en minutos desde las 00:00.

Una vez recogida la muestra hay que estimar como afecta la hora del día y la distancia al tiempo de trayecto. Para hacer una estimación más fiable, analizando únicamente la influencia del tiempo y la distancia en el aumento del tiempo de trayecto, y no sesgar la estimación con variaciones en el tiempo de trayecto por las propias condiciones de la vía y límites de velocidad, se ha restado el mínimo tiempo de trayecto posible como un indicador que resumen las características de la vía, a todas las observaciones de la misma ruta. Ahora los datos indican únicamente un nivel de retraso y no el tiempo de trayecto completo en cada instante.

La estimación final será por tanto el comportamiento del retraso derivado del tráfico sobre el tiempo de trayecto dependiendo de la hora y la distancia del recorrido. Esta estimación se ha hecho por MCO con el programa GRETL. El único objetivo de esta estimación es explicar el comportamiento lo mejor posible del tráfico a través la hora y de la distancia.

El gráfico 1 muestra el resultado de la muestra recogida para estimar el comportamiento del nivel de retraso. La estimación va a estar sesgada por tres factores muy importantes, la estimación de Google Maps del tiempo de trayecto, las variables del modelo y su carácter temporal. El tiempo que indica Google Maps también es una estimación, y como tal se hace cada vez más imprecisa a medida que aumenta el tiempo de trayecto, ya que partir de una media hora de trayecto, la estimación se hace en intervalos de 5 minutos, y si el tiempo de trayecto supera los 60 minutos más o menos, el intervalo de estimación es de 10 minutos. Esto hace que la información recogida a través de Google Maps sea poco precisa, generando un nivel de explicación reducido, por lo que el R-cuadrado no va a ser el mejor indicador para seleccionar el modelo de comportamiento del retraso. Este sesgo se ha intentado corregir un poco eliminando las observaciones que generen demasiado retraso por kilómetro, usando el rango intercuartil (IQR o RIC en español) para detectar las observaciones anómalas. Otro factor importante para tener en cuenta es el sesgo por la evidente omisión de variables importantes, como variables dummy para reflejar características importantes de la vía, lo que puede ser un posible causante de evidencias de heteroscedasticidad.

Por último, por el carácter temporal de la variable dependiente no se descarta la presencia de autocorrelación.

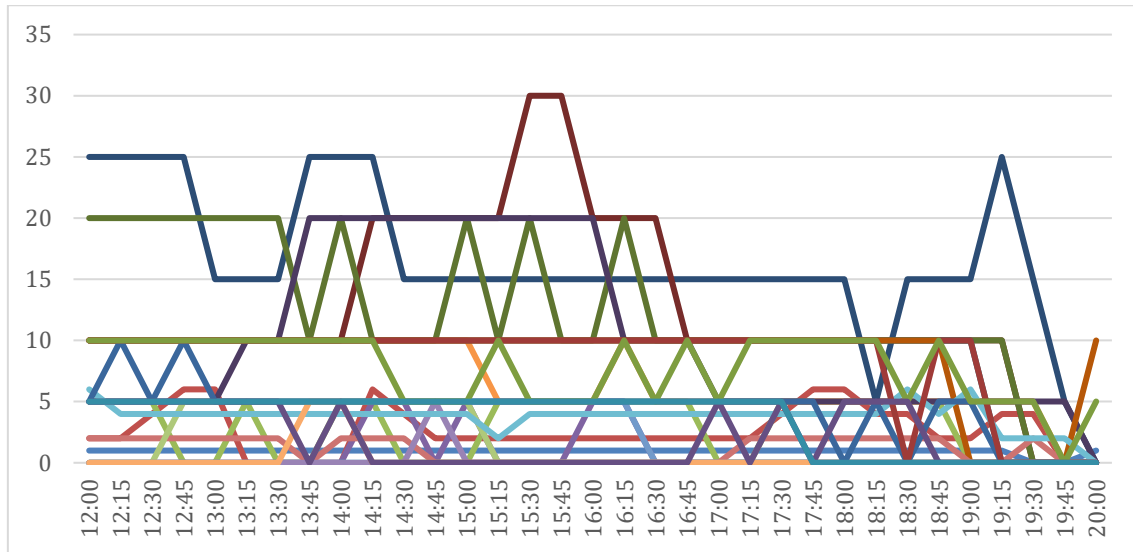


Gráfico 1. Minutos de retraso a distintas horas del día de las 23 rutas consideradas. Fuente: elaboración propia vía Excel.

Por estas razones, la selección de un modelo frente a otro para explicar el retraso será por las características que se observan en el gráfico 1, que son: (1) incremento no lineal del retraso con respecto a la distancia; (2) pendiente negativa con respecto a la hora; (3) un incremento en la pendiente del tiempo con respecto a la distancia; y por último (4) ninguna observación negativa. Por tanto, el modelo que se utilizará será el que cumpla estas características y tenga un R-cuadrado lo mejor posible.

Modelo 4: MCO, usando las observaciones 1-690

Variable dependiente: r

	coeficiente	Desv. típica	Estadístico t	valor p	
t_km	-0.000103814	1.68414e-05	-6.164	1.21e-09	***
t	-0.00288813	0.000754580	-3.827	0.0001	***
l_km	4.05060	0.245784	16.48	1.17e-051	***
sq_km	0.00143270	0.000132407	10.82	2.66e-025	***
Media de la vble. dep.	12.83333	D.T. de la vble. dep.	7.278797		
Suma de cuad. residuos	13622.13	D.T. de la regresión	4.456157		
R-cuadrado no centrado	0.909272	R-cuadrado centrado	0.626830		

Imagen 2. Resultado de la estimación por MCO del retraso. Fuente: elaboración propia vía Gretl.

La imagen 2 muestra el resultado de la estimación del retraso por tráfico dependiendo de la hora de inicio de la ruta y la distancia. En ella se observa que todas las variables incluidas son estadísticamente relevantes y tiene un nivel de explicación de la variable dependiente del 62,68%.

La hora a la que se inicia la ruta al nodo v_j es la hora de atención del nodo s_i más su tiempo de servicio sv_i . La función que explica el retraso es como sigue:

$$f(km_{ij}, s_i, sv_i) = 4,0506 \ln(km_{ij}) + 1,4327 \cdot 10^{-3} km^2 - 2,88813 \cdot 10^{-3} (s_i + sv_i) - 1,03814 \cdot 10^{-4} (s_i + sv_i) km_{ij}$$

El gráfico 2 muestra la estimación del retraso con esta función y el retraso observado con respecto a la hora. Como se puede observar la función cumple con las 4 características necesarias y además tiene un R-cuadrado lo suficientemente elevado como para aceptar esta forma funcional e introducirla dentro del modelo.

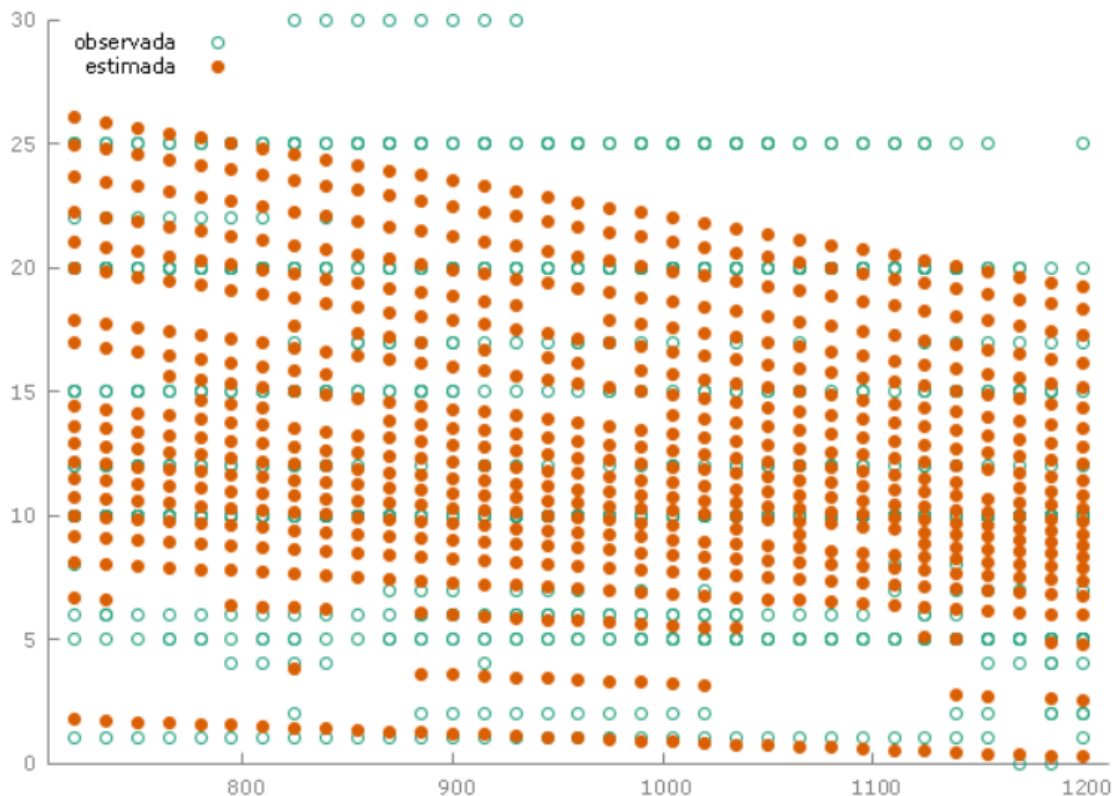


Gráfico 2. Retraso observado y estimado con respecto a la hora en minutos. Fuente: elaboración propia vía Gretl.

4.2 Formulación matemática.

Como la estimación es lineal respecto a la hora del trayecto, no hay problema en incorporar la estimación dentro de él. El modelo 1, sin embargo, tal cual se planteó anteriormente, no va a reflejar en el resultado óptimo la influencia del retraso en el tiempo de trayecto por cómo está definido. Por tanto, hay que modificarlo para que los resultados de ambos problemas sean comparables sin alterar la estructura original. Para poder implementar de manera adecuada la función de retraso hay que añadir un nuevo parámetro al modelo:

- km_{ij} indica la distancia en kilómetros entre el nodo i y el nodo j .

También hay que añadir una nueva variable:

- t_{ij} es una variable entera positiva que indica el tiempo de trayecto necesario desde el nodo v_i al nodo j .

Incluyendo el parámetro y variable en el modelo 1 será el modelo 2, que explica los tiempos de trayecto que se formula de la siguiente manera:

$$\min \sum_{i,j \in A} t_{ij} \quad (4.1)$$

$$\text{s. a.} \quad \sum_{j \in N \setminus \{0,j\} \in A} x_{0j} \geq \sum_{i \in N} d_i/q \quad (4.2)$$

$$\sum_{i \in N \setminus \{i,0\} \in A} x_{i0} - \sum_{j \in N \setminus \{0,j\} \in A} x_{0j} = 0 \quad (4.3)$$

$$\sum_{j \in N \setminus \{i,j\} \in A} x_{ij} = 1 \quad \forall i \in N \quad (4.4)$$

$$\sum_{j \in N \setminus \{j,i\} \in A} x_{ji} = \sum_{j \in N \setminus \{i,j\} \in A} x_{ij} \quad \forall i \in N \quad (4.5)$$

$$x_{ij} + x_{ji} \leq 1 \quad \forall i, j \in E \quad (4.6)$$

$$u_j - u_i \geq d_j - \text{big}M(1 - x_{ij}) \quad \forall i, j \in A; i \neq 0; j \neq 0 \quad (4.7)$$

$$d_i \leq u_i \leq q \quad \forall i \in N \quad (4.8)$$

$$t_{ij} - c_{ij} \geq f(km_{ij}, s_i, sv_i) - \text{big}M \cdot x_{ij} \quad \forall i, j \in A; i \neq 0 \quad (4.9)$$

$$t_{0j} - c_{0j} \geq f(km_{0j}, a_0, sv_0) - \text{big}M \cdot x_{0j} \quad \forall j \in N \quad (4.10)$$

$$s_j - a_0 \geq t_{0j} \quad \forall j \in N \quad (4.11)$$

$$s_j - s_i \geq sv_i + t_{ij} - \text{big}M(1 - x_{ij}) \quad \forall i, j \in A; j \neq 0; i \neq 0 \quad (4.12)$$

$$b_0 - s_i \geq sv_i + t_{i0} \quad \forall i \in N \quad (4.13)$$

$$a_i \leq s_i \leq b_i \quad \forall i \in N \quad (4.14)$$

$$x_{ij} \in \{0, 1\} \quad \forall i, j \in A \quad (4.15)$$

$$s_i \geq 0 \quad \forall i \in V \quad (4.16)$$

$$u_i \geq 0 \quad \forall i \in V \quad (4.17)$$

$$t_{ij} \geq 0 \quad \forall i, j \in A \quad (4.18)$$

La función objetivo (4.1) para este modelo cambia, ya que el costo de trayecto ya no es constante, será la variable t_{ij} que minimiza el costo de tiempo de trayecto como en el modelo anterior. Además, se incorporan las restricciones (4.9) y (4.10). La restricción (4.9) relaciona la variable t_{ij} con su valor si el arco está activo. Por tanto, la diferencia entre tiempo de trayecto t_{ij} y el coste c_{ij} tendrá que ser la función $f(km_{ij}, s_i, sv_i)$ como la estimación del retraso generado solo si el arco esta activo, en caso contrario, como t_{ij} se encuentra en la función objetivo el mecanismo por la M grande le permite tomar el valor 0. La restricción (4.10) es similar a la anterior pero definida para el nodo del depósito en el origen, estableciendo la hora de atención $s_1 = a_0$ como el inicio de la ventana de tiempo del depósito y sustituyendo el valor sv_0 . Las restricciones (4.15), (4.16), (4.17) y (4.18) indican la naturaleza de las variables.

5. BASE DE DATOS.

Creado el modelo hay recolectar toda la información con la que va a trabajar, que será toda aquella necesaria para definir numéricamente todos los parámetros. Los datos necesarios que requiere el problema serán: (1) la distancia entre cada nodo, (2) el tiempo de trayecto entre los nodos, (3) los límites temporales de la ventana de tiempo, (4) la demanda de gasoil de los clientes, (5) el tiempo de servicio a cada cliente y (6) la información del vehículo.

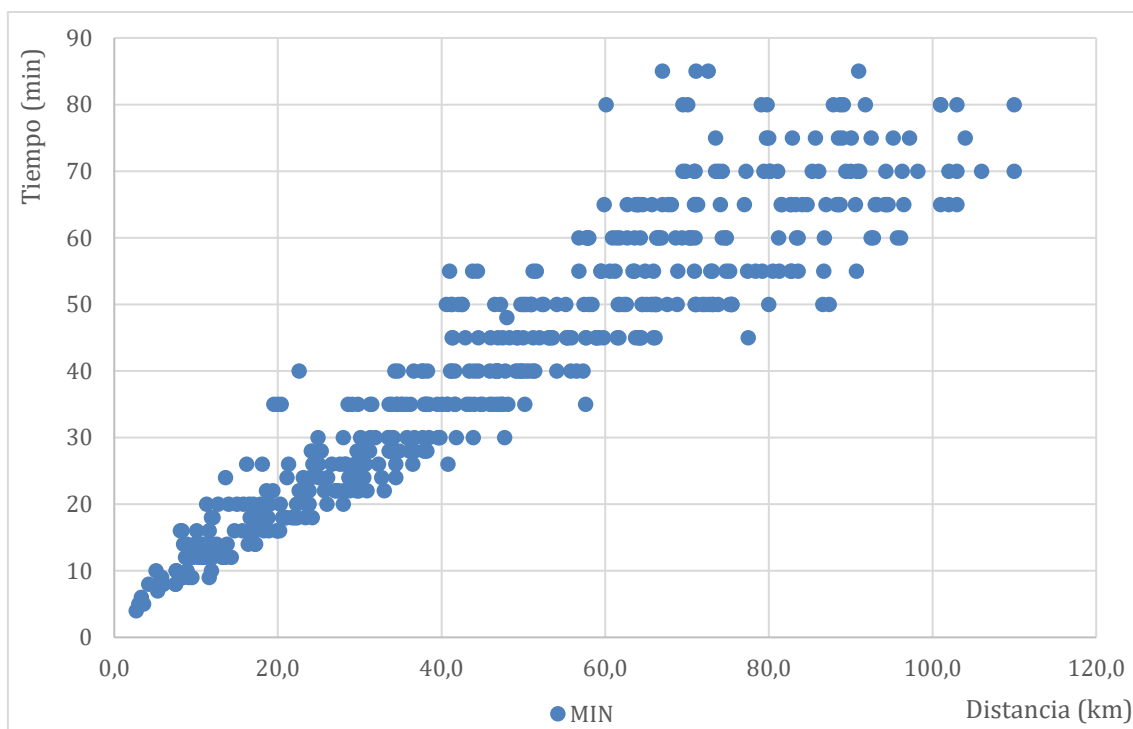


Gráfico 3. Dispersión del tiempo de trayecto mínimo frente a la distancia. Fuente: elaboración propia vía Excel.

Los datos de tiempo de trayecto y distancia solo hará falta recogerlos en una dirección, ya que por construcción del problema se da por supuesto que invertir el trayecto no altera el valor. Los datos de distancia fueron recogidos a través de Google Maps en kilómetros. Los datos de tiempo de trayecto difieren de un modelo a otro, el modelo 2 trabaja con los tiempos de trayecto mínimos para cada ruta, que se recogieron a la vez que las distancias por Google Maps. Los tiempos de

trayecto del modelo 1 será el tiempo que se tarda en realizar el trayecto en la peor situación del tráfico, el tiempo máximo posible, para garantizar que en la aplicación de real del problema se cumple las ventanas de tiempo. Este se aproxima con la estimación del retraso generado por el tráfico, cuyo máximo se encuentra al inicio del intervalo considerado, que se suma al tiempo mínimo de trayecto obteniendo los tiempos máximos de trayecto esperados. El resultado de los datos temporales y distancia recogidos están en el gráfico 3.

El modelo de camión cisterna es homogéneo para todas las rutas que tendrá una capacidad de almacenamiento de 20.000 litros de carburante, y tiene una bomba de 400 litros de gasolina por minuto. Este último dato será necesario para calcular el tiempo de servicio que requiere cada cliente como el tiempo que tarda en suministrar el carburante que demanda cada gasolinera, es decir, los litros demandados dividido entre la potencia de la bomba por minuto. A este tiempo se le ha sumado un tiempo de preparación y gestión. Este tiempo de gracia se ha aproximado en 40 minutos ya que se trata del transporte de mercancía peligrosa, por lo que requiere cumplir numerosas medidas de seguridad y coordinación entre el personal, que puede requerir un tiempo considerable. Los datos de la demanda se generarán aleatoriamente entre 2500 y 4500 litros salvo la demanda del depósito que será nula, al igual que su tiempo de servicio. Los límites de la ventana de tiempo de los clientes también se han generado aleatoriamente entre las 12:00h y las 20:00h, la ventana del tiempo del depósito será el intervalo completo de 8 horas desde las 12:00h hasta las 20:00h.

6. PROGRAMACIÓN MATEMÁTICA.

Para resolver problemas de este tamaño es necesario recurrir a programas informáticos. En este caso se ha usado el programa Python. Este programa no es específico para la optimización, permitiendo otros usos en estadística o análisis de datos entre otros. Las siguientes líneas se refieren al código que traduce el modelo 1 previamente descrito al lenguaje informático con el que trabaja Python.

Primero se definen los parámetros y conjunto de índices que hacen falta para resolver el problema, utilizando la misma notación original del problema. Hay que destacar el uso de la librería numpy para generar las demandas aleatoriamente de cada cliente.

```
n= 31 #nº de clientes
N= [x for x in range(1,n+1)] #conjunto de clientes
V= [0]+N #conjunto de nodos
bigM= 999999 #M-Grande
A= [(i,j) for i in V for j in V if i!=j] #arcos posibles
E=[(i,j) for i in V for j in V if i!=j and i<=j] #conexiones posibles
c= {(i,j):tiempo[i][j] for i,j in A} #coste del arco
q= 20000 #capacidad del vehiculo
lpm= 400 #bomba de gasolina
import numpy as np
rnd= np.random
rnd.seed(1)
demandas= {i:rnd.randint(2500,4500) for i in N} #demandas
d= [0]+list(demandas[i] for i in N)
sv= {i:40+d[i]/lpm for i in V} #tiempo de servicios a cada cliente
```

Los datos de tiempo y distancia del problema se introducen en un formato de Python conocido como listas con el nombre de 'distancia' y 'tiempo'. Los parámetros de cada nodo también se introducen como listas para definir las ventanas de tiempo y su ubicación geográfica.

Usando la librería docplex se puede definir el problema descrito en el apartado 3.

```

from docplex.mp.model import Model
mdl=Model('VRPTW')
#VARIABLES
x=mdl.binary_var_dict(A, name='arco')
u=mdl.integer_var_dict(V, ub=q, name='carga')
s=mdl.integer_var_dict(V, name='hora')
#FUNCION OBJETIVO
mdl.minimize(
    mdl.sum(c[i,j]*x[i,j] for i,j in A))
#RESTRICCIONES
#estructura de la ruta
mdl.add_constraint(
    mdl.sum(x[i,0] for i in N)-mdl.sum(x[0,j] for j in N)==0)

mdl.add_constraint(
    mdl.sum(x[0,j] for j in N)>=mdl.sum(d[i] for i in N)/q)

mdl.add_constraints(
    mdl.sum(x[i,j] for j in V if i!=j)==1
    for i in N)
mdl.add_constraints(
    x[i,j]+x[j,i]<=1
    for i,j in E)
mdl.add_constraints(
    mdl.sum(x[j,i] for j in V if j!=i)==mdl.sum(x[i,j] for j in V if j!=i)
    for i in N)
#nivel de carga
mdl.add_constraints(
    u[j]-u[i]>=d[j]-bigM*(1-x[i,j])
    for i,j in A if j!=0)
mdl.add_constraints(
    u[i]>=d[i]
    for i in N)
#ventana de tiempo
mdl.add_constraints(
    s[j]-nodo[0][1][0]>=c[0,j]
    for j in N)
mdl.add_constraints(
    s[j]-s[i]>=sv[i]+c[i,j]-bigM*(1-x[i,j])
    for i,j in A if j!=0 and i!=0)
mdl.add_constraints(
    nodo[0][1][1]-s[i]>=sv[i]+c[i,0]
    for i in N)
mdl.add_constraints(
    nodo[i][1][0]<=s[i]
    for i in N)
mdl.add_constraints(
    s[i]<=nodo[i][1][1]
    for i in N)
len(x)+len(s)+len(u)

```

En su versión limitada solo permite resolver un problema con máximo 1000 variables y 1000 restricciones, pero hay una versión ilimitada (IBM ILOG CPLEX Optimization Studio) que es gratuita para estudios académicos.

La librería folium es una buena herramienta para la representación geográfica de los nodos, tal como muestra la imagen 1, usando los datos de longitud y latitud previamente introducidos. No influye en la resolución del problema, solo es una herramienta de apoyo para visualizar los datos.

```
import folium as flm
mapa= flm.Map(location=[sum(nodo[i][2] for i in V)/(n+1),sum(nodo[i][3] for i in V)/(n+1)],
                tiles="openstreetmap",
                zoom_start=10)
for i in N:
    flm.Marker(location=[nodo[i][2],nodo[i][3]], popup="Gasolinera").add_to(mapa)
flm.Marker(location=[nodo[0][2],nodo[0][3]], popup="Refinería", icon=flm.Icon(icon='')).add_to(mapa)
mapa
```

Una vez construido el problema se ejecuta el código para iniciar el proceso de resolución del modelo.

```
mdl.parameters.timelimit=999999
solution= mdl.solve(log_output=True)
```

Una vez que se haya resuelto, se identifica el resultado obtenido numéricamente del 0 al 5:

7. 0 – UNKOWN: No se ha podido identificar una solución.
8. 1 – FEASIBLE_SOLUTION: se encontró una solución factible.
9. 2 – OPTIMAL_SOLUTION: se encontró una solución óptima.
10. 3 – INFEASIBLE_SOLUTION: el modelo no tiene solución.
11. 4 – UNBOUNDED_SOLUTION: el modelo no está acotado.
12. 5 – INFEASIBLE_OR_UNBOUNDED_SOLUTION: el modelo no tiene solución o no está acotado.

```
mdl.get_solve_status()
```

Finalmente, si existe una solución, muestra el valor de todas las variables no nulas.

```
print(solution)
```

Usando la librería matplotlib se accede a la solución obtenida y los datos geográficos para dibujar un gráfico en distancias euclidianas como representación visual del resultado obtenido.

```

import matplotlib.pyplot as plt
plt.figure(figsize=(14,10))
plt.scatter(list((nodo[i][3] for i in N)),list((nodo[i][2] for i in N)),color='blue',marker='s')
plt.scatter(nodo[0][3],nodo[0][2],color='red',marker='s')
plt.xlabel('Longitud')
plt.ylabel('Latitud')
S= [k for k in A if x[k].solution_value>0.9]
for i,j in S:
    plt.plot([nodo[i][3],nodo[j][3]],[nodo[i][2],nodo[j][2]],color='blue',alpha=0.5)
plt.show()

```

Para completar el análisis del resultado se indica la numero de rutas o vehículos necesarios en la solución y la suma de la distancia recorrida.

```

vehiculos= ((0,j) for j in N)
rutas= [k for k in vehiculos if x[k].solution_value>0.9]
len(rutas) #nº de camiones que se utilizan

```

```

sum(distancia[i][j] for i,j in S)

```

Para el modelo 2 hay que modificar la función objetivo y crear una nueva variable:

```

t=mdl.integer_var_dict(A, name= 'tiempo')

mdl.minimize(
    mdl.sum(t[i,j] for i,j in A))

```

Además, hay que sustituir las restricciones de la ventana de tiempo como sigue:

```

mdl.add_constraints(
    t[0,j]-c[0,j]>=-2.0794536+4.0506*ma.log(distancia[0][j],ma.e)
    +distancia[0][j]*(0.0014327*distancia[0][j]-0.07474608)-bigM*(1-x[0,j])
    for j in N if j!=0)
mdl.add_constraints(
    t[i,j]-c[i,j]>=4.0506*ma.log(distancia[i][j],ma.e)
    +0.0014327*(distancia[i][j]**2)-0.00288813*(s[i]+sv[i])
    -0.000103814*((s[i]+sv[i])*distancia[i][j])-bigM*(1-x[i,j])
    for i,j in A if i!=0)
mdl.add_constraints(
    s[j]-nodo[0][1][0]>=c[0,j]
    for j in N)
mdl.add_constraints(
    s[j]-s[i]>=sv[i]+t[i,j]-bigM*(1-x[i,j])
    for i,j in A if j!=0 and i!=0)
mdl.add_constraints(
    nodo[0][1][1]-s[i]>=sv[i]+t[i,0]
    for i in N)
mdl.add_constraints(
    nodo[i][1][0]<=s[i]
    for i in N)
mdl.add_constraints(
    s[i]<=nodo[i][1][1]
    for i in N)

```

7. RESULTADOS.

7.1 Exposición de los resultados.

n	Modelo 1				Modelo 2				Diferencia (min)	Diferencia (km)	Diferencia (n° de rutas)
	óptimo (min)	n° de rutas	tiempo (s)	Distancia (km)	óptimo (min)	n° de rutas	tiempo (s)	Distancia (km)			
2	176	1	0,02	175,5	174	1	0,6	175,5	-2	0,0	0
4	297	2	0,03	286,3	293	2	0,03	286,3	-4	0,0	0
6	483	3	0,06	426,8	422	2	0,05	377,5	-61	-49,3	-1
8	485	3	0,05	423,5	432	2	0,05	376,6	-53	-46,9	-1
10	511	3	0,05	432,8	490	3	0,09	421,9	-21	-10,9	0
12	568	3	0,08	460,1	557	3	0,14	460,1	-11	0,0	0
14	648	4	0,13	530,59	604	3	0,23	488,89	-44	-41,7	-1
16	717	4	0,38	585,4	669	4	0,78	535,6	-48	-49,8	0
18	756	4	0,52	621,3	721	4	3,03	574,59	-35	-46,7	0
20	805	5	10,52	644,8	753	5	9,77	590,59	-52	-54,2	0
22	841	5	14,47	683,4	789	5	42,09	626,09	-52	-57,3	0
24	860	5	6,58	690,4	829	5	14,69	660,99	-31	-29,4	0
26	947	6	107,02	743,8	888	5	413,77	713,79	-59	-30,0	-1
28	1010	6	325,44	797,4	943	6	283,06	746	-67	-51,4	0
30	1069	6	280,33	834,2	1003	6	436,39	794,09	-66	-40,1	0
31	1093	6	97,23	832,3	1054	7	1697,4	795,7	-39	-36,6	1

Tabla 1. Resultados y comparación de la optimización de ambos modelos. Fuente: elaboración propia vía Excel.

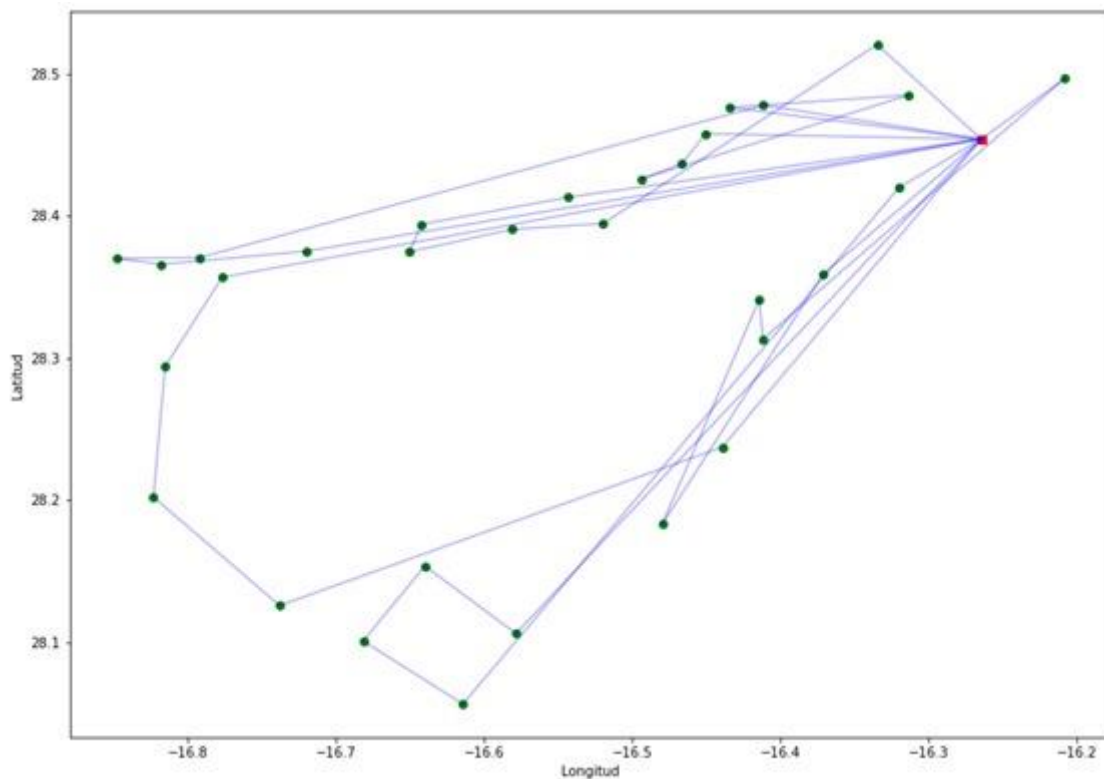


Gráfico 4. Resultado gráfico en distancias euclidianas del modelo 1. Fuente: elaboración propia vía Python.

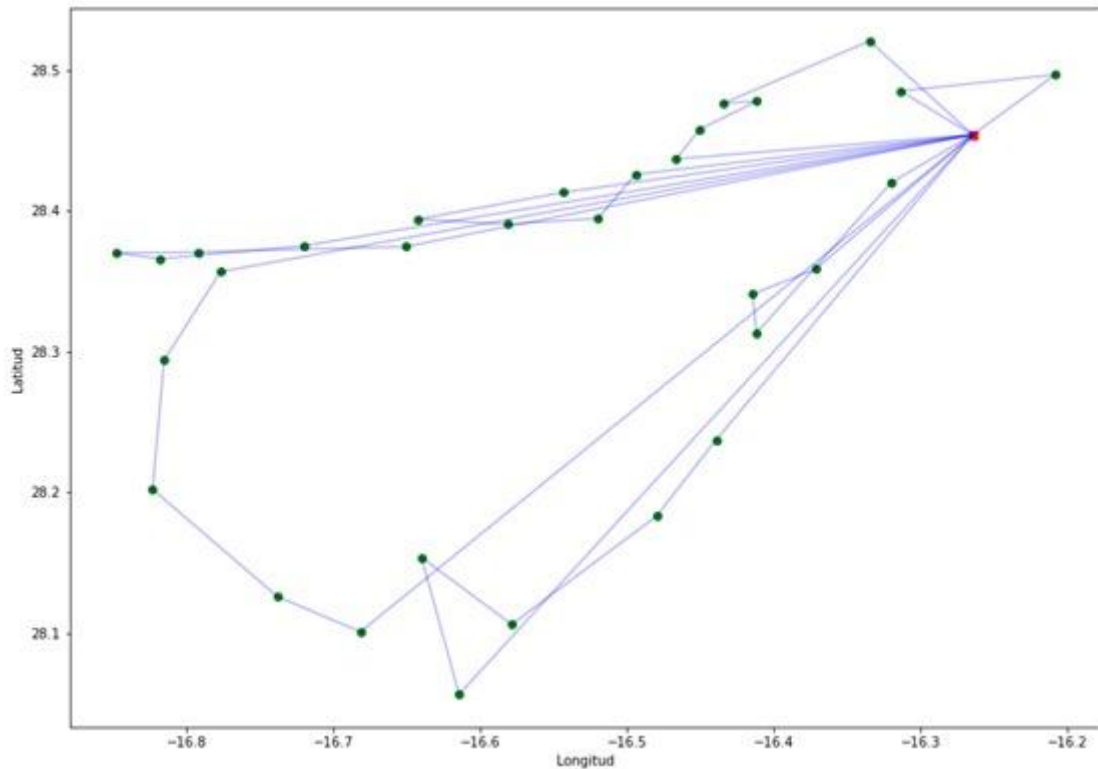


Gráfico 5. Resultado gráfico en distancias euclidianas del modelo 2. Fuente: elaboración propia vía Phyton.

Como será difícil sacar una conclusión clara de un solo problema, la tabla 1 muestra una serie de subproblemas con una selección de n clientes por orden alfabético. Partiendo desde un número mínimo de 2 clientes, se han ido aumentando uniformemente de dos en dos a la vez que se han resuelto para ambos modelos.

La tabla 1 contiene los resultados del problema y todos los subproblemas. La tabla contiene dos grupos de columnas iguales para cada problema. La primera columna es el resultado óptimo del problema en minutos. Por ejemplo, el tiempo total de trayecto para los 10 clientes es de 8h y 31min con el tiempo de trayecto constante y 8h y 10min para el tiempo de trayecto variable. La segunda columna indica el número de rutas que se han generado para llegar a esa solución óptima. La tercera columna indicando el tiempo que ha necesitado el programa en resolver el problema en segundos. La cuarta columna indica la distancia total recorrida en la solución óptima. Las últimas tres columnas comparan los resultados de estos dos modelos en su variación con respecto al tiempo óptimo, la distancia recorrida y el número de rutas.

El gráfico 4 muestra la solución óptima graficada en distancias euclidianas del modelo 1 y el gráfico 5 para el modelo 2. En la última fila de la tabla 1 está el resultado del proceso de optimización ambos modelos. El programa ha tardado 1min y 37s en resolver el modelo 1 y 28min y 17s en resolver el modelo 2. El óptimo del modelo 1 es un total de 6 rutas para atender a todos los clientes en un tiempo total de 18 horas y 13 minutos, recorriendo una distancia total de 832 kilómetros. El óptimo del modelo 2 es un total de 7 rutas con un tiempo total de 17 horas 34 minutos, recorriendo una distancia de 795 kilómetros. En el modelo 2 se ha reducido el tiempo mínimo de trayecto en 39 minutos y una diferencia de recorrido de 39 kilómetros usando una ruta más.

7.2 Defensa de los resultados.

En primer lugar, hay que destacar que, dado el elevado número de clientes, el problema trabaja con un conjunto muy grande de arcos. Este elevado número de arcos generaría bastante tiempo de computación de no ser por las ventanas de tiempo. Las ventanas de tiempo permiten reducir el conjunto de soluciones factibles estableciendo un orden indirecto de atención a cada cliente. En otras palabras, dos ventanas de tiempo que no se solapan establece la preferencia de atención de uno frente al otro, reduciendo el número de posibles soluciones. Sin las ventanas de tiempo el tiempo de resolución aumentaría exponencialmente, haciendo que el problema sea cada vez más difícil de resolver con el incremento del número de clientes. Esta ventaja del VRPTW frente al VRP no solo mejora los requisitos de computación, sino también acerca los problemas de rutas un paso más a la realidad, ya que es difícil imaginar que hoy en día cualquier empresa que requiera de transporte, entrega y/o recogida no trabaje con unas horas aproximadas para la atención de cada cliente. No descarto el hecho de que habrá situaciones que no requiere de ventanas de tiempo, como por ejemplo el transporte a almacenes, y en estos casos la rapidez de resolución depende únicamente de la complejidad del problema para encontrar un mínimo óptimo. Evidentemente, lo que es una ventaja también es su desventaja, ya que requiere de un mayor esfuerzo para recoger los datos necesarios y establecer las restricciones adecuadas para que la solución óptima tenga sentido tanto matemático como económico.

En segundo lugar, a pesar de que sea una solución óptima, el resultado tanto del modelo 1 como del modelo 2 es solo aproximación de la mejor solución real acorde a las restricciones impuestas matemáticamente. Hay un margen de error por factores que no se han incluido en el modelo o directamente no se pueden explicar numéricamente. Por ejemplo, las rutas inversas, aunque sea lógico pensar que se puede ir del punto A al punto B en ambos sentidos sin alterar excesivamente el tiempo y la distancia necesaria, es igual de lógico que no en todas las rutas esta igualdad se va a cumplir. Otro ejemplo es el factor humano, algo completamente impredecible y mucho menos cuantificable, por lo que la solución óptima será cierta únicamente en las condiciones descritas dentro del modelo.

8. CONCLUSIONES.

Finalmente queda responder a la cuestión planteada inicialmente de la influencia del tiempo del trayecto como variable. Analizando las últimas 3 columnas de la tabla 1, se puede sacar 3 argumentos. (1) El cambio ha influido en el tiempo óptimo de ruta, en todas las ocasiones este tiempo se ha reducido. La magnitud de la influencia en el tiempo no parece ser producto del tamaño del problema, ya que si se observa la columna de la diferencia en minutos esta sube y baja a medida que aumenta el tamaño del problema, por lo que depende más del conjunto de rutas que se consideran. (2) Para tener una vaga idea de cómo ha influido en la estructura de las rutas, se puede observar la diferencia en distancia recorrida en la ruta, que tenderá a ser la misma si esto no influye. Lo que respecta la estructura de las rutas también se identifica una clara reducción en la distancia de las rutas, en ocasiones, esta se mantiene o es muy similar, pero no está claro que esta diferencia se incrementa con el tamaño del problema. (3) Por último, lo que respecta el número de rutas varía en ocasiones en una ruta más o menos, sin correlación aparente con el

tamaño del problema. Por tanto, podemos concluir que si difieren bastante los resultados, y por tanto el tiempo como variable altera significativamente el resultado óptimo.

Queda por aclarar cuál de los dos modelos es mejor para predecir una buena estructura de rutas. El modelo 1 con el tiempo constante tiene la ventaja de ser bastante más simple en el tratamiento de datos, sin embargo, tiene la desventaja de que depende del valor aproximado que se le da al tiempo de trayecto. En este caso se ha establecido como el máximo posible, para asegurar que la solución óptima se cumple a pesar de las perturbaciones en el tiempo de trayecto. Se podía haber usado el tiempo medio de retraso, pero esto habría generado dudas sobre el cumplimiento de las ventanas de tiempo en condiciones desfavorables. El modelo 2 con el tiempo variable se espera que sea más preciso en el valor del tiempo de trayecto, porque tiene la ventaja de que puede actualizarlo constantemente a cambio de una mayor complejidad en el tratamiento de los datos, teniendo que recurrir a otras áreas como la econometría.

A pesar de ello, creo que el modelo 2 es el más fiable si a una aplicación real se refiere, no por la diferencia en el tiempo de trayecto o por que llega a reducir la distancia total recorrida indirectamente, sino por cómo se estructura la ruta. Esto es algo que se observa en los gráficos 4 y 5, me refiero a que el modelo 1 suma de la mejor manera posibles los tiempos tal que el tiempo máximo es mínimo, lo que lo lleva a estructurar las rutas bajo ese criterio. Si comparamos ambas gráficas, las rutas del modelo 2 son en cierta manera más ordenadas, es decir, rutas largas seguidas de múltiples rutas cortas, porque interpreta que es óptimo aprovechar ubicaciones cercanas para generar el menor retraso por tráfico posible. Esto ocurre porque el retraso de una ruta y el tiempo mínimo de trayecto dependen positivamente de la distancia que se vaya a recorrer. Por tanto, minimizando la distancia se consigue indirectamente minimizar el tiempo de trayecto y el retraso por el tráfico independientemente de la hora a la que se realiza el trayecto, ya que las rutas más cortas tienen un retraso que tiende a cero, convirtiendo a la distancia en un buen indicador para aproximar una solución óptima de tiempo de trayecto variable.

Lo que respecta el resultado del modelo 1, deja claro que el tiempo máximo de una ruta no resume la influencia del tráfico en ella a la hora de optimizar un modelo, aunque queda abierto si esto también ocurre para otras aproximaciones del tiempo de trayecto constante.

9. RECOMENDACIONES.

Este trabajo se limita al análisis del tiempo de trayecto, pero se podría repetir en términos de costes, y ver si influye de la misma manera. También se puede analizar otras perspectivas del VRP como la capacidad heterogénea o un análisis más profundo de las rutas entre nodos. Todas las variantes tendrán su influencia a nivel matemático ya que se está alterando las restricciones del problema, y esas restricciones pueden ser más restrictivas para omitir un conjunto de soluciones, o menos restrictivas para alcanzar otras. En cualquier caso, siempre es importante tener en cuenta este tipo de análisis ya que pueden mejorar la eficacia de la recolección de datos y la eficiencia de la estimación de rutas óptimas por VRP.

10. BIBLIOGRAFÍA.

Lau, H. C., Sim, M., & Teo, K. M. (2003). Vehicle routing problem with time windows and a limited number of vehicles. *European journal of operational research*, 148(3), 559-569. (Lau et al., 2003)

Linares, P., Ramos, A., Sánchez, P., Sarabia, A., & Vitoriano, B. (2001). Modelos matemáticos de optimización. *Madrid, España*.

Muñoz, E. A. B., & Sotero, J. H. C. (2009). Diseño de un modelo de optimización de rutas de transporte. *El hombre y la máquina*, (32), 52-67.

Olivera, A. (2004). Heurísticas para problemas de ruteo de vehículos. *Reportes Técnicos 04-08*.

Pérez, C. T., Olivares-Benítez, E., & Flores, J. M. (2011). Revisión y programación de modelos de optimización como una plataforma en gams-cplex para problemas de ruteo de vehículos. *Taller Latino Iberoamericano de Investigación de Operaciones, Universidad Popular Autónoma del Estado de Puebla*.

Salazar-González, J.J. (2001) Programación Matemática, Ed. Díaz de Santos.