

Félix Xavier López Da Silva

*El Problema del Viajante de Comercio
Consistente*

The Consistent Travelling Salesman Problem

Trabajo Fin de Grado
Grado en Matemáticas
La Laguna, Julio de 2023

DIRIGIDO POR

Juan José Salazar González

Juan José Salazar González
Departamento de Matemáticas,
Estadística e Investigación
Operativa
Universidad de La Laguna
38200 La Laguna, Tenerife

Agradecimientos

A mi pareja y familia por el apoyo incondicional. A mi tutor por los consejos y la sabiduría.

Félix Xavier López Da Silva
La Laguna, 10 de julio de 2023

Resumen · Abstract

Resumen

El Problema del Viajante de Comercio Consistente (ConTSP por sus siglas en inglés) es una variación del TSP con periodos, donde además se exige consistencia en el tiempo. Sus aplicaciones a la sociedad y el mercado son importantes, como por ejemplo en los servicios sanitarios y en los modelos de gestión de inventario. Mediante el lenguaje de programación Julia y el software de optimización Gurobi, se crea un código que soluciona este problema. Uno de los objetivos de este trabajo es ver como se comportan los tiempos de ejecución del código para sacar conclusiones del problema y del lenguaje de programación. Además se utiliza la librería TSPLIB que nos proporciona ejemplos que se trabajan en este tipo de estudios y se comparan dos de ellos para un análisis mas profundo del problema.

Palabras clave: *Consistencia – TSP – Julia*

Abstract

The Consistent Traveling Salesman Problem (ConTSP) is a variation of the TSP that incorporates time periods and requires time consistency. Its applications in society and the market are important, such as in healthcare services and inventory management models. By using the Julia programming language and the Gurobi optimization software, a code is created to solve this problem. One of the objectives of this work is to observe the code's execution times to draw conclusions about the problem and the programming language. Additionally, the TSPLIB library is used, which provides examples commonly used in these types of studies. Two of these examples are compared for a more in-depth analysis of the problem.

Keywords: *Consistence – TSP – Julia*

Contenido

| | |
|--|-----|
| Agradecimientos | III |
| Resumen/Abstract | V |
| Introducción | IX |
| 1. TSP y consistencia. | 1 |
| 1.1. TSP | 1 |
| 1.1.1. Definición | 1 |
| 1.1.2. Modelo matemático | 2 |
| 1.1.3. Variantes del TSP | 2 |
| 1.2. Servicio de consistencia | 3 |
| 1.2.1. Consistencia en el reparto de paquetes pequeños | 4 |
| 1.2.2. Consistencia en los cuidados sanitarios | 4 |
| 2. TSP consistente | 7 |
| 2.1. Definición y notación del problema | 7 |
| 2.2. Formulación | 8 |
| 2.2.1. Formulación 1 | 9 |
| 2.3. Ejemplos ilustrativos | 10 |
| 2.3.1. Formulación 2 y 3 | 12 |
| 3. Resultados computacionales | 15 |
| 3.1. Resultados para el conTSP sin tiempos de espera | 15 |
| 3.1.1. burma14 | 16 |
| 3.1.2. ulysses16 | 19 |
| 3.1.3. Ajuste de la M grande. | 21 |
| A. Apéndice | 27 |
| A.1. Distribuciones VMI | 27 |
| A.2. Ciclo Hamiltoniano | 27 |

| | |
|-----------------------------|----|
| A.3. Código utilizado | 27 |
| Bibliografía | 29 |
| Poster | 31 |

Introducción

Imaginemos que soy un proveedor que tengo una cartera de clientes que visitar en un mes. Cada día del mes tengo que visitar a ciertos clientes dados, de manera que al menos a uno lo tengo que visitar mas de una vez al mes. Pero para ofrecerles un mejor servicio a mis clientes , quiero que el día que toca visitarlos, hacerlo mas o menos a la misma hora , es decir , no quiero que entre dos visitas al mismo cliente haya una diferencia de mas de dos horas, por ejemplo. De esta manera, ¿cómo puedo minimizar el número de kilómetros recorridos durante el mes?

La programación lineal tienen una amplia aplicabilidad en este tipo de problemas. En concreto el TSP y sus variantes son unos de los problemas de programación lineal entera mas estudiados y su importancia ha ido creciendo durante los años. Pero la sociedad exige más servicio que los que puede cumplir el TSP, y es aquí donde surge la consistencia. Los campos en los que este servicio encuentra utilidad son muchos, como en los modelos de gestión de inventario, en los servicios sanitarios, la industria del reparto de paquetes, entre muchos otros. El ConTSP, Problema del Viajante de Comercio Consistente, surge de imponer la consistencia en los tiempos de llegada.

Debido a la complejidad de cálculo que tienen estos problemas, se recurre a softwares y lenguajes de programación que permitan que la máquina nos ayude a computar todas estas operaciones. En esta disertación el lenguaje de programación escogido es Julia, un lenguaje menos conocido que Python y Matlab, en el que se demostrará la eficacia que tiene a la hora de codificar este tipo de problemas. Por otro lado, todo el proceso de optimización se ha llevado a cabo con el software Gurobi.

Gracias al código creado se pretende hacer un análisis de como se comporta el ConTSP si se cambian ciertos parámetros, en concreto, la constante de consistencia, el número de nodos y la cantidad periodos.

El trabajo se divide en tres capítulos. En el primero se encuentra un acercamiento a la programación lineal dado por el Problema del Viajante de Comercio(TSP), el cual nos ayudará a comprender el problemas principal de una

manera mas clara y concisa. También en este capítulo se expone que significa un servicio consistente. El segundo capítulo es el marco teórico en el cual vamos a situar todo nuestro trabajo y por ende, donde se define y formula el ConTSP. Además, en él se desarrollan algunos ejemplos y se proponen ciertas variantes. Por último, encontramos el capítulo de resultados computacionales, en cual mediante el código creado se hace una análisis de como se comportan los tiempos de ejecución del código para cada problema, así como los valores óptimos.

TSP y consistencia.

Este capítulo contiene una motivación del problema estudiado y algunas conceptos básicos para una mayor comprensión de este. En primer lugar se hace una introducción al Problema del Viajante de Comercio (TSP) para que los conceptos explicados posteriormente sean más fáciles de asimilar. Todo ello lleva una comprensión implícita de teoría de grafos. Para terminar, se tratan las condiciones de consistencia en este tipo de problemas y cuales son algunas de sus aplicaciones.

1.1. TSP

El Problema del Viajante de Comercio (TSP, por sus siglas en inglés) plantea la siguiente pregunta: Dada una lista de ciudades y las distancias entre cada par de ciudades, ¿cuál es la ruta más corta posible que visite cada ciudad exactamente una vez y regrese a la ciudad de origen? Es un problema NP-duro en optimización combinatoria, importante en ciencias de la computación teórica e investigación operativa.

1.1.1. Definición

Sea el grafo no dirigido $G = (V, A)$, de conjunto nodos $V = \{0, 1, \dots, n + 1\}$ y conjunto de arcos $A = \{(i, j)/i, j \in V, i \neq j\}$, y el costo asociado $c_{i,j}$ a cada arista de G . Por notación, escogemos el nodo 0 como el depósito. Por otro lado, sea $\delta^+(i) := \{j \in V : (i, j) \in A\}$ y $\delta^-(i) := \{j \in V : (j, i) \in A\}$. Por último, dado un subconjunto de nodos $S \subseteq V_p \cup \{0\}$, sea $A(S)$ el conjunto de arcos con nodos en S , es decir, $A(S) := \{(i, j)/i, j \in S, i \neq j\}$. El Problema del Viajante de Comercio consiste en determinar el ciclo (o circuito) Hamiltoniano en G cuyas aristas tengan un costo total mínimo, o en otras palabras, la ruta de menor coste.

Dicho problema aparece en numerosos procesos en los que hay que secuenciar un conjuntos de tareas, siendo c_{ij} el costo de realizar la tarea i y a continuación la j , y se busca minimizar el costo de realizar todas las tareas.

También se podría considerar el hecho de que el costo de una varíe según el sentido que se use, o lo que es equivalente, el problema sobre un grafo orientado. Este el caso de las carreteras urbanas , donde algunas calles tienen sentido unidireccional. Entonces se habla de TSP asimétrico (ATSP).

1.1.2. Modelo matemático

Se considera la variable binaria asociada a cada arista :

$$x_{ij} = \begin{cases} 1, & \text{el arco (i,j) es utilizado} \\ 0, & \text{en otro caso} \end{cases}$$

entonces un modelo de Programación Lineal Entera para el TSP sería :

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

sujeto a :

$$\sum_{j \in \delta^+(i)} x_{ij} = 1 \quad i \in V \cup \{0\} \quad i \in V \quad (1.2)$$

$$\sum_{j \in \delta^-(i)} x_{ij} = 1 \quad i \in V \cup \{0\} \quad i \in V \quad (1.3)$$

$$\sum_{(i,j) \in A(S)} x_{ij} \leq 1 \quad S \subset V : 0 \in S \quad (1.4)$$

$$x_{ij} \in \{0, 1\} \quad i, j \in V \quad (1.5)$$

En primer lugar en 1.1 se encuentra la función objetivo que minimizar la suma de los costes de las aristas tomadas en el camino. Las restricciones 1.2 y 1.3 imponen que para cada nodo en el tour entre y salga una arista considerada en el camino óptimo. La eliminación de subciclos en la ruta óptima se debe a la restricción 1.4.

1.1.3. Variantes del TSP

En la literatura , existen numerosas variantes del Problema del Viajante de Comercio. A continuación , se enumeran algunas de ellas.

- MAX-TSP: Consiste en encontrar un circuito hamiltoniano con el coste máximo.
- TSP con cuello de botella: Consiste en encontrar un circuito hamiltoniano que minimice el mayor coste entre todas las aristas, en lugar de minimizar el coste total.

- TSP gráfico: Consiste en encontrar un circuito de coste mínimo que visite cada ciudad al menos una vez.
- TSP agrupado: Las ciudades se dividen en grupos, y el objetivo es encontrar un circuito hamiltoniano de coste mínimo en el que se visiten los nodos de cada grupo de manera consecutiva.
- TSP generalizado: Las ciudades también se dividen en grupos, pero el objetivo es encontrar un circuito de coste mínimo que visite exactamente un nodo de cada grupo.
- TSP con múltiples viajeros: Hay un número m de viajeros, cada uno debe visitar algunas ciudades. El problema se transforma en encontrar una partición de los nodos a visitar X_1, \dots, X_m y m ciclos, uno para cada X_i , de manera que la suma de las distancias recorridas por los m viajeros sea mínima.

1.2. Servicio de consistencia

La calidad del servicio y la satisfacción del cliente son consideraciones importantes y con alta aplicabilidad en los problemas de planificación de rutas de vehículos. Una característica clave de un servicio al cliente de alta calidad, es su consistencia a lo largo del tiempo, y esto es particularmente importante cuando las empresas se enfrentan a decisiones de rutas. En tales casos, generalmente es deseable brindar un servicio consistente a los clientes regulares, es decir, a aquellos clientes que requieren un servicio frecuente en los diferentes períodos. Así se pueden clasificar los servicios de consistencia, como se hace en [3], de la siguiente manera :

- Consistencia en el tiempo de llegada, para el cual el cliente debe ser visitado aproximadamente a la misma hora durante el día.
- Consistencia orientada a la persona, para el cual un cliente debe ser visitado siempre por el mismo trabajador.
- Consistencia en la entrega, para la cual un cliente debe recibir aproximadamente la misma cantidad de mercancía en cada servicio.

Este trabajo se centra en la consistencia en el tiempo de llegada. En este contexto, el proveedor tiene como objetivo reducir la variabilidad en los tiempos reales de la ruta en los que se atiende a un cliente, ya que hacerlo generalmente aumenta el valor del servicio para el cliente. Por ejemplo, en el contexto de las distribuciones VMI (Vendor Managed Inventory) A.1, o gestión del inventario por el proveedor, la consistencia de llegada reduce la necesidad de que el cliente comprometa recursos de muelle de carga durante todo el día. En la industria del cuidado en el hogar, las personas mayores y discapacitadas son sensibles a los cambios en sus rutinas diarias y desarrollar horarios consistentes es de particular importancia. Además, desde el punto de vista del proveedor de servicios, reducir la variabilidad en las entregas repetitivas a lo largo de múltiples períodos de

tiempo puede generar eficiencias que se traducen en ahorros significativos de costos.

Cabe destacar que la comodidad del cliente también se puede lograr de otras formas que no sea mediante la consistencia en el servicio. Por ejemplo, en el Problema de Rutas de Vehículos con Ventanas de Tiempo (VRPTW), los clientes pueden ser visitados solo si están disponibles. En el Problema de Recogida y Entrega, cada pasajero está asociado con una ubicación de recogida y entrega, y se consideran restricciones de tiempo máximo de viaje para evitar inconvenientes a los clientes. En el Problema de Rutas de Vehículos Periódico (PVRP), los clientes requieren un número determinado de visitas durante el horizonte de planificación y especifican un espaciado mínimo y máximo entre dos visitas.

Este trabajo se enfoca en la consistencia y se describen algunas áreas de aplicación para ilustrar su relevancia en planificación de rutas de vehículos.

1.2.1. Consistencia en el reparto de paquetes pequeños

El reparto de paquetes pequeños y cartas se ha convertido en una parte esencial de la industria del transporte. Las ganancias generadas por los transportistas de pequeños paquetes superan los ingresos de todos los principales modos de transporte de carga (por ejemplo, transporte aéreo, ferroviario y barco) excepto el transporte por camión. La competencia en el envío de paquetes pequeños es intensa, tanto las empresas privadas (por ejemplo, DHL, FedEx, United Parcel Service) como las empresas públicas (por ejemplo, servicios postales nacionales) deben operar con máxima eficiencia para mantenerse en el mercado. Para diferenciarse, las empresas ofrecen una variedad de servicios relacionados con el transporte. Además, muchas empresas mejoran la calidad del servicio al proporcionar consistencia.

Una característica distintiva importante del envío de paquetes pequeños es la necesidad de consistencia en el servicio. Los clientes regulares son aquellos que son visitados de manera rutinaria y prefieren ser visitados aproximadamente a la misma hora cada día. La consistencia del conductor mejora la relación entre el conductor y el cliente y permite un servicio personalizado. Por lo general, el aumento en el costo producido por el servicio de consistencia se compensa con creces con una mayor satisfacción del cliente y una mayor familiaridad del conductor con el lugar y los clientes.

1.2.2. Consistencia en los cuidados sanitarios

Mejorar la atención médica es un desafío importante para las sociedades modernas. El número de personas dependientes de cuidados (por ejemplo, enfermos crónicos, discapacitados físicos y personas mayores) está aumentando

constantemente. Al mismo tiempo, las restricciones financieras se están volviendo más estrictas debido a la disminución de la población activa.

Uno de los logros más importantes en la gestión de la atención médica ha sido el cambio de formas estacionarias de atención médica (por ejemplo, hogares de cuidado) a la atención domiciliaria. El servicio de atención domiciliaria permite a los pacientes vivir en su hogar con el mayor grado posible de autonomía mientras son visitados y apoyados por personal especializado. Por lo general, el servicio incluye tareas domésticas, comidas a domicilio y atención médica.

Debido a la creciente competencia, los recortes presupuestarios y, lo que es más importante, las obligaciones de servicios sociales, los proveedores de atención domiciliaria deben equilibrar cuidadosamente la eficiencia de costos y la calidad del servicio; esto es cierto tanto para organizaciones privadas como públicas. Para brindar un servicio de alta calidad, los trabajadores deben tener conocimiento sobre los clientes y sus hogares, y deben ser adiestrados para el servicio requerido. Además, los trabajadores deben estar familiarizados con los gustos y preferencias, habilidades y limitaciones, y rutinas de los clientes. Debido a la intimidad física, los clientes deben sentirse cómodos con el proveedor de servicios. La consistencia del personal mejora significativamente la calidad del servicio; simplifica la comunicación entre las personas involucradas y reduce el tiempo necesario para revisar el historial de un cliente.

TSP consistente

El Problema del Viajante de Comercio consistente o ConTSP (Consistent Traveling Salesman Problem) es una variante del muy conocido Problema del Viajante de Comercio o TSP (Traveling Salesman Problem). Se diferencia en que el ConTSP añade el problema de proporcionar consistencia en las rutas de los múltiples periodos. En el ConTSP se pretende diseñar rutas de coste mínimo, una para cada uno de los múltiples periodos que se consideran, en las cuales es conocido los clientes van a ser atendidos en cada periodo, utilizando un único vehículo. En general un cliente puede o no requerir servicio en un periodo, de manera que solo un subconjunto de clientes requieren servicio en todos los periodos. El servicio de consistencia en los tiempos de llegada es requerido por todos los clientes a los que se le atiende en más de un periodo, lo que significa que todos estos clientes serán atendidos aproximadamente a la misma hora en cada periodo que requieran servicio. El tiempo de llegada sigue siendo una variable de decisión, pero la diferencia en los tiempos de llegada para cada cliente durante los múltiples periodos no debe sobrepasar cierta cantidad constante, a la que se llama máxima diferencia permitida de tiempo de llegada. En la práctica, dicha constante puede ser establecida por el cliente o por el proveedor del servicio. Nótese que, si la máxima diferencia permitida de tiempo de llegada es infinita, entonces el ConTSP se reduce a múltiples TSPs separados, es decir, el ConTSP es un NP-difícil, como el TSP.

2.1. Definición y notación del problema

Se considera $G = (V, A)$ el grafo completo dirigido de $n+1$ nodos, donde $V = \{0, 1, \dots, n+1\}$ es el conjunto de nodos y $A = \{(i, j) / i, j \in V, i \neq j\}$ el conjunto de arcos del grafo. Se denota el nodo 0 como el depósito, y por ende $V_c = V \setminus \{0\}$ es el subconjunto de clientes. Asociado a cada arco hay un costo $c_{ij} \geq 0$ y una duración del trayecto $t_{ij} \geq 0$. El tiempo de servicio (o de procesamiento) de cada cliente puede ser incorporado en el tiempo de trayecto con la operación $t_{ij} \leftarrow t_{ij} + s_i$, donde $s_i \geq 0$ es el tiempo de servicio para el

cliente i , con $s_0 = 0$. Nótese que no se imponen costos ni tiempos simétricos, es decir, se permite $c_{ij} \neq c_{ji}$ y $t_{ij} \neq t_{ji}$ para cualquier $(i, j) \in A$. En cualquier caso, ya que en general los tiempos de servicio difieren entre clientes, el ConTSP es un problema asimétrico, incluso si los tiempos y los costos de trayectos fueran simétricos.

Ahora sea $\mathcal{P} = \{1, \dots, h\}$ el conjunto de periodos tal que $h \geq 2$. Para cada periodo $p \in \mathcal{P}$, se define $V_p \subseteq V_c$ como el conjunto de clientes que requieren servicio en este periodo y el conjunto de arcos asociados es $A_p := \{(i, j)/i, j \in V_p \cup \{0\}, i \neq j\}$. Además, se asume para cada periodo $p \in \mathcal{P}$ que $V_p \cap (\cup_{q \in V_c, q \neq p} V_q) \neq \emptyset$, por tanto V_p no puede ser vacío y además incluye algún cliente que requiere servicio en algún periodo más. Esto se debe a que si no es así, el problema sería equivalente al TSP.

Se llama ruta del periodo p al ciclo Hamiltoniano (A.1) en $G_p := (V_p \cup \{0\}, A_p)$. Denotamos a cada ruta como $T_p = \langle 0, v_1, v_2, \dots, v_{|V_p|}, 0 \rangle^p$, donde cada v_k es único para cada periodo, es decir, en la ruta de un periodo no se atiende dos veces al mismo cliente. Definimos el coste de esta ruta como $c(T_p) := \sum_{k=1}^{|V_p|} c_{v_{k-1}v_k} + c_{v_{|V_p|}0}$. También definimos el tiempo de llegada al i -ésimo nodo $v_i \in V_p$ como $a_{v_i}^p$, donde $v_0 = 0$ para simplificar la notación.

Una solución del TSP Consistente es una colección de rutas $\{T_1, T_2, \dots, T_h\}$, es decir, una para cada periodo $p \in \mathcal{P}$. Para cada colección, definimos $\Delta a_i^{max} := \max_{p \in \mathcal{P}: i \in V_p} a_i^p - \min_{p \in \mathcal{P}: i \in V_p} a_i^p$ como la máxima diferencia de tiempos de llegada para el cliente i . En el ConTSP queremos forzar a que esa diferencia no exceda un valor permitido $L > 0$, para todos los clientes $i \in V_c$; esto es, una solución del ConTSP es factible si y solo si $\max_{i \in V_c} \Delta a_i^{max} \leq L$. Una colección de rutas se dice consistente si induce una solución factible del ConTSP. El objetivo del ConTSP es buscar la colección de rutas consistente de mínimo coste.

Continuando con la notación, para cada cliente $i \in V_p$ se denota por $\delta_p^+(i)$ el conjunto de nodos j para los cuales hay un arco de i a j en el grafo G_p , es decir $\delta_p^+(i) := \{j \in V : (i, j) \in A_p\}$. De la misma manera, $\delta_p^-(i) := \{j \in V : (j, i) \in A_p\}$. Por último, dado un subconjunto de nodos $S \subseteq V_p \cup \{0\}$, sea $A(S)$ el conjunto de arcos con nodos en S , es decir, $A(S) := \{(i, j)/i, j \in S, i \neq j\}$ y sea $\delta(S)$ el conjunto de arcos con exactamente un nodo en S , es decir, $\delta(S) := \{(i, j) \in S \times S^c\} \cup \{(i, j) \in S^c \times S\}$, donde $S^c = (V_p \cup \{0\}) \setminus S$.

2.2. Formulación

Se presenta aquí una formulación conceptual del problema para luego incorporar la utilizada en la implementación en Julia. En primer lugar se definen las variables que se van a utilizar:

$$x_{ij}^p = \begin{cases} 1, & \text{el arco (i,j) es utilizado en el periodo } p \\ 0, & \text{en otro caso} \end{cases}$$

una variable binaria que nos permite utilizar la siguiente formulación del Con-TSP:

$$\min \sum_{p \in \mathcal{P}} \sum_{(i,j) \in A_p} c_{ij} x_{ij}^p \quad (2.1)$$

sujeto a:

$$x_{ij}^p \in \{0, 1\} \quad (i, j) \in A_p, \quad p \in \mathcal{P} \quad (2.2)$$

$$\sum_{j \in \delta_p^+(i)} x_{ij}^p = 1 \quad i \in V_p \cup \{0\}, \quad p \in \mathcal{P} \quad (2.3)$$

$$\sum_{j \in \delta_p^-(i)} x_{ij}^p = 1 \quad i \in V_p \cup \{0\}, \quad p \in \mathcal{P} \quad (2.4)$$

$$\{(i, j) \in A_p : x_{ij}^p = 1\} = T_p, \quad p \in \mathcal{P} \quad (2.5)$$

$$\{T_p, T_q\} \text{ es consistente, } (p, q) \in \mathcal{P} \times \mathcal{P} : p < q \quad (2.6)$$

En esta formulación las restricciones (2.3) y (2.4) son las asociadas al clásico TSP y aseguran que a cada nodo de la ruta le entra y sale un arco. La restricción (2.5) elimina la creación de subciclos, imponiendo que a cada ruta del periodo se le asocie un ciclo Hamiltoniano. Para terminar, la restricción (2.6) impone la condición de consistencia.

2.2.1. Formulación 1

La formulación que se va a introducir es la que se ha ejecutado en el código utilizado. Requiere la definición de una nueva variable:

$$u_j^p : = \text{tiempo de llegada al nodo } j \text{ en el periodo } p \in \mathcal{P}$$

Además se utiliza una $M \geq 0$ suficientemente grande que ayuda a codificar adecuadamente el modelo. Por otro lado, se utilizan las constantes t_{ij} que representan el tiempo que se tarda del nodo i al j . En cuestión, dicha formulación incluye la función objetivo (2.1), las restricciones (2.2), (2.3), (2.4) y además estas nuevas restricciones:

$$u_j^p \geq u_i^p + t_{ij} x_{ij}^p - M(1 - x_{ij}^p) \quad i \in V_p, \quad j \in V_p \cup \{0\}, \quad p \in \mathcal{P} \quad (2.7)$$

$$u_j^p \leq u_i^p + t_{ij} x_{ij}^p + M(1 - x_{ij}^p) \quad i \in V_p, \quad j \in V_p \cup \{0\}, \quad p \in \mathcal{P} \quad (2.8)$$

$$u_i^k - u_i^l \leq L \quad k, l \in \mathcal{P}, \quad i \in V_k \cap V_l \quad (2.9)$$

La tarea de las restricciones (2.7) y (2.8) es computar adecuadamente los tiempos de llegada a los nodos y de esta manera evitar que se creen subciclos en la ruta. El

papel de la consistencia lo cumple la restricción (2.9). Con todo esto se computa el TSP consistente sin tiempos de espera.

Si se quisiese computar el TSP consistente con tiempos de espera, solamente habría que omitir la restricción (2.8) para que se permitiera esperar gratuitamente en cada nodo el tiempo que hiciese falta.

El uso de la M es necesario para introducir en el modelo de forma lineal una condición lógica muy importante en la definición del problema. Se trata de la condición:

- Si $(i, j) \in A_p$ entonces $u_j^p \geq u_i^p + t_{ij}$
- Si $(i, j) \notin A_p$ entonces $u_j^p \geq u_i^p - M$

Gracias al parámetro M la desigualdad (2.7) equivale a dicha condición. Pero es fundamental que al determinar el valor de M , este cumpla con $u_i^p - M \leq 0$ para no obligar nada sobre u_j^p cuando $(i, j) \notin A_p$. Es decir, necesitamos $M \geq u_i^p$ para todo $i \in V_p$ y todo $p \in \mathcal{P}$. Pero a la vez también necesitamos que M sea lo menor posible para que no conlleve problemas numéricos en los cálculos con el ordenador. De esta manera M también es adecuada para la restricción (2.8).

2.3. Ejemplos ilustrativos

Para una comprensión visual, véase la aplicación de lo explicado a ciertos ejemplos. Para ello, se implementa en Julia un código que genere puntos aleatorios en un plano y utilizando la formulación 1, resuelva y encuentre las rutas óptimas.

Ejemplo 1: Sea un conjunto de 15 clientes a los que hay que prestar servicio, distribuidos según la figura 2.1 en un terreno de dimensiones 1000×1000 .

A parte, el número de periodos será 3. Digamos que cada día se visitan los siguientes clientes:

1. {2, 7, 9, 11, 13, 15, 16}
2. {5, 10, 14, 15}
3. {2, 3, 5, 7, 8, 10, 13, 14}

Se quiere probar entonces, como se comporta el problema y su solución para 2 valores distintos de la constante de consistencia $L=100$ y $L=1000$. En definitiva, se tienen 16 puntos distribuidos aleatoriamente en el plano $[0, 1000] \times [0, 1000]$. Se define la distancia euclídea entre los puntos $i, j \in V$ como d_{ij} . Entonces se considera $c_{ij} = d_{ij} = t_{ij}$.

Tras ejecutar el programa con estos datos tenemos los siguientes resultados dados por la tabla 2.3.

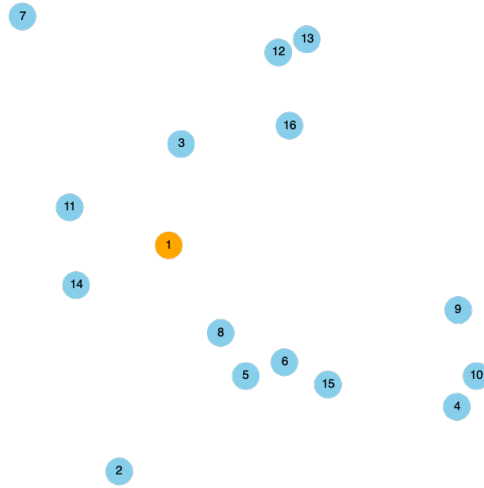


Figura 2.1. Distribución de los 16 nodos en un plano 1000×1000 . El nodo número 1 (de color naranja) es el depósito.

| L | clientes | t(sec) | Valor |
|------|----------|--------|--------|
| 100 | 15 | 9.85 | 8597.5 |
| 1000 | 15 | 0.08 | 7923.8 |

Tabla 2.1. Tabla que representa la constante de consistencia , número de clientes, tiempo de ejecución del código en segundos y el valor objetivo en el conTSP sin tiempos de espera.

Una diferencia observable es que para una constante de consistencia menor, el valor objetivo es mas grande. Esto es lógico ya que cuanto mas pequeña sea la constante de consistencia, más difícil va a ser resolver el problema. Gráficamente, las rutas de cada periodo se representan en las figuras 2.3 y 2.2.

Se observa como las rutas en cada periodo cambian notablemente con simplemente modificar la constante de consistencia. Por otro lado, también cambian los tiempos de ejecución del código. Esto se debe a lo ya comentado, el problema

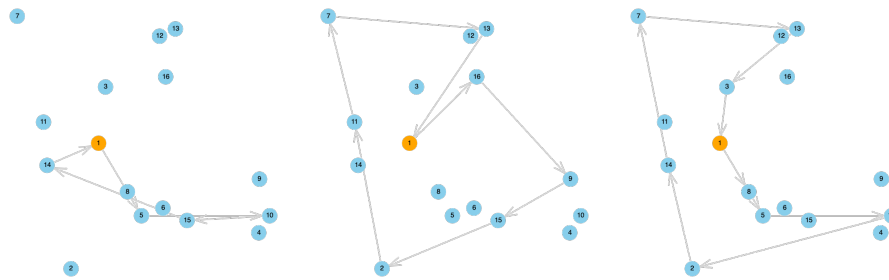


Figura 2.2. Representación de las rutas de los periodos 1, 2 y 3 con $L=100$

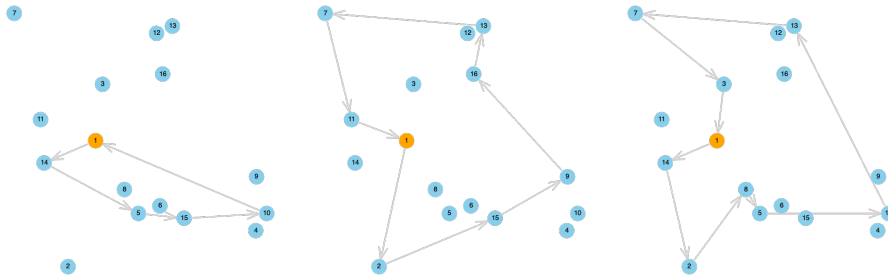


Figura 2.3. Representación de las rutas de los periodos 1, 2 y 3 con $L=1000$.

es más complicado cuanto menor sea la constante de consistencia. Estas variaciones de los tiempos de compilación podría ser interesante estudiarlos de manera mas exhaustiva.

Ejemplo 2: Del mismo modo se puede estudiar el conTSP pero ahora con tiempos de espera. En este caso, antes de viajar hacia otro nodo, el modelo puede decidir esperar en nodo el tiempo que le haga falta. Consideremos la misma distribución de nodos que en ejemplo anterior. Así, surgen los siguientes resultados:

| L | clientes | t(sec) | Valor |
|----------|-----------------|---------------|--------------|
| 0 | 15 | 0.07 | 7923.8 |
| 100 | 15 | 0.03 | 7923.8 |
| 1000 | 15 | 0.08 | 7923.8 |
| 2000 | 15 | 0.08 | 7910.8 |

Tabla 2.2. Tabla que representa la constante de consistencia , número de clientes, tiempo de ejecución del código en segundos y el valor objetivo en el conTSP con tiempos de espera

Naturalmente , los valores objetivos y tiempos de compilación son menores o iguales que en el conTSP sin tiempos de espera, esto es debido al menor número de restricciones cuando hay tiempos de espera. Una cosa que es posible con tiempos de espera , es tomar la constante de consistencia igual a 0. Si uno se fija en los tiempos de llegada a los nodos representados en la tabla 2.3 , se da cuenta de que el viajante espera gratuitamente en cada nodo para cumplir la consistencia.

Esto dentro de un modelo podría ser poco beneficioso, por ello se introduce esta nueva formulación para subsanar estos casos.

2.3.1. Formulación 2 y 3

Dentro del conTSP con tiempos de espera , se podría plantear el hecho de penalizar al modelo si decide esperar en un nodo. Para ello se proponen dos

| nodo | Periodo 1 | Periodo 2 | Periodo 3 |
|------|-----------|-----------|-----------|
| 1 | 0 | 0 | 0 |
| 2 | 2705.5 | na | 2705.5 |
| 3 | na | na | 157.3 |
| 4 | na | na | na |
| 5 | na | 2313.4 | 2313.4 |
| 6 | na | na | na |
| 7 | 554.4 | 0 | 554.4 |
| 8 | na | na | 2399.8 |
| 9 | 1776.5 | na | na |
| 10 | na | 1809.6 | 1809.6 |
| 11 | 223.7 | na | na |
| 12 | na | na | na |
| 13 | 1175.5 | na | 1175.5 |
| 14 | na | 3005.6 | 3005.6 |
| 15 | 2133,9 | 2133.9 | na |
| 16 | 1312,9 | na | na |

Tabla 2.3. Tabla que representa los tiempos de llegada del periodo 1, 2 y 3. Na representa que no pasa por ese nodo en el periodo correspondiente.

modificaciones de la función objetivo. Hay que mencionar que el hecho de estas modificaciones de la f.o. daría lugar a un nuevo problema que no es exactamente el conTSP, pero se parece bastante, y en el cual se consideran las restricciones (2.2)-(2.4), (2.7) y (2.9).

Formulación 2: La nueva función objetivo, viene dada por:

$$\min \sum_{p \in \mathcal{P}} \sum_{(i,j) \in A_p} c_{ij} x_{ij} + \sum_{p \in \mathcal{P}} \sum_{j \in V_p} u_j^p \quad (2.10)$$

De esta manera, los tiempos de espera se suman al valor objetivo y se tratan de minimizar estos tiempos. A su vez, se podría plantear multiplicar la sumatoria de los tiempos de llegada para dar mayor o menor importancia a estos dentro de la función objetivo.

Formulación 3: Como novedad la función objetivo es:

$$\min \sum_{p \in \mathcal{P}} \sum_{j \in V_p} u_j^p \quad (2.11)$$

En la cual solo tenemos en cuenta los tiempos de llegada y se trata de minimizar su suma.

Resultados computacionales

Lo que se trata de hacer en este capítulo es, ya que no existen bases de datos para el ConTSP, utilizar la conocida librería TSPLIB [4], en concreto la librería de Julia TSPLIB.jl, para tomar algunos ejemplos que contiene su base de datos y ejecutarlos con el código utilizado A.2. Concretamente se toman los ejemplos que corresponden a los TSP simétricos y que tienen menos de 30 nodos y uno con 42 para el cual los tiempos de ejecución son viables, en total 10 ejemplos.

Para cada ejemplo, de una manera similar a la que se propone en [2], se ejecuta el código teniendo en cuenta los siguiente. Se va a considerar una f que dirá que probabilidad en porcentaje que tiene un cliente de estar en la ruta de un periodo. Por otro lado como, constante de consistencia, L , se tomará un porcentaje específico del tiempo máximo de la solución óptima del problema, al cual llamaremos T . Para finalizar, K será el número de periodos que se toman.

TSPLIB nos proporciona una matriz de distancia entre los nodos, que surge del tipo de distancia considerado en cada ejemplo. Para los ejemplos tomados se utilizan las distancias euclídea, pseudo-euclídea, geométrica y otras definidas por matrices. Lo importante es que la duración de los trayectos t_{ij} y los costos c_{ij} se toman iguales a la distancia entre el nodo i y j .

Por otro lado, el valor que se le da a la M es algo que se podría estudiar también. Sabemos que su valor tiene que cumplir $M \geq u_j^p$. Un valor plausible para esto sería tiempo máximo de la solución óptima del problema, es decir, el tiempo que dura el viajero en completar la ruta con todos los nodos.

Toda la implementación fue desarrolla en el lenguaje de programación Julia. Las computación se realizó en procesador Apple M1 de 8 núcleos. Además se utilizó el software optimizador Gurobi.

3.1. Resultados para el conTSP sin tiempos de espera

Para este problema se utiliza la formulación 1 (2.2.1) en la cual su función objetivo y restricciones definen el problema del conTSP sin permitir que el viajante pueda esperar en las ubicaciones de los clientes. Por otro lado hemos

tomado una $M=10000$ que superen con creces al tiempo óptimo del TSP. Así mismo, se toman las instancias mencionadas para crear la tabla 3.1.

| Ejemplo | nodos | M | Tiempo(s) | | | Valor objetivo | | |
|-----------|-------|-------|-----------|--------|--------|----------------|--------|--------|
| | | | L=10 % | L=20 % | L=30 % | L=10 % | L=20 % | L=30 % |
| bayg29 | 29 | 10000 | 4,61 | 1,46 | 2,08 | 2892.0 | 2891 | 2891 |
| bays29 | 29 | 10000 | 3,31 | 1,65 | 1,96 | 3689 | 3680 | 3680 |
| burma14 | 14 | 10000 | 0,17 | 0,11 | 0,1 | 5806.0 | 5806.0 | 5806 |
| fri26 | 26 | 10000 | 0,97 | 1,3 | 0,5 | 2014 | 2014 | 2014 |
| gr17 | 17 | 10000 | 0,21 | 0,15 | 0,13 | 4861 | 4779 | 4754 |
| gr21 | 21 | 10000 | 0,13 | 0,12 | 0,13 | 5982 | 5982 | 5982 |
| gr24 | 24 | 10000 | 2,61 | 0,6 | 0,32 | 2762 | 2707 | 2665 |
| swiss42 | 42 | 10000 | 290,28 | 149,77 | 10,9 | 2710 | 2702 | 2672 |
| ulysses16 | 16 | 10000 | 1,28 | 0,68 | 0,54 | 11386 | 11299 | 11237 |
| ulysses22 | 22 | 10000 | 10,3 | 10,71 | 9,6 | 10031 | 10031 | 10031 |

Tabla 3.1. Tabla que expone los tiempos de ejecución por un lado y el valor objetivo por otro. Se toman 3 periodos y una $f=50\%$.

En esta tabla se recogen los tiempos de ejecución y valores objetivo para todas las instancias que se van a estudiar en este trabajo. Se elige $K=3$, pues para valores mayores los tiempos son inviables. Lo mismo ocurre con el porcentaje de probabilidad de la existencia de un nodo en una ruta, si se aumenta, los tiempos se disparan. Un claro ejemplo de ello es la diferencia entre el tiempo de ejecución tomando el ejemplo swiss42 con los demás. El hecho de tener una diferencia de 10, 20 nodos por ruta con los demás hace que sea mucho mas costoso de resolver.

Se observa como la M tomada es 10000, sin embargo se podría ajustar a cada problema en específico y analizar como se comportan los tiempos de ejecución.

Se toma ahora uno de estos ejemplos para ver como se comportan los tiempos de espera y los valores óptimos con respecto a las máximas diferencias permitidas de tiempo de llegada y a la f (o aumento de nodos).

3.1.1. burma14

Se escoge burma14 como objeto de este análisis. La tabla 3.1.1 recoge los tiempos de ejecución para el problema de 3 y 5 periodos y con las condiciones de consistencia y cantidad de nodos mencionada. Se puede observar el aumento de los tiempos de ejecución a medida que disminuye la constante de consistencia.

Por otro lado el aumento de los nodos y de los periodos provocan un incremento bastante notable. Gracias a las gráficas 3.1 y 3.2, también se puede mencionar que el aumento de los nodos y de los periodos producen una mayor sensibilidad de los tiempos de espera con respecto a la disminución de la L .

En la gráfica 3.1, la línea gris, que representa el ejemplo para un probabilidad de nodos del 0.9, está por encima de las demás pues su cantidad de nodos

| L | K=3 | | | K=5 | | |
|-------|--------|--------|--------|--------|--------|---------|
| | f=50 % | f=70 % | f=90 % | f=50 % | f=70 % | f=90 % |
| 10 % | 0,17 | 5,28 | 29,8 | 2,21 | 290,75 | 2327,38 |
| 20 % | 0,11 | 7,02 | 18,41 | 0,55 | 30,95 | 604,11 |
| 30 % | 0,1 | 3,7 | 8,57 | 0,19 | 11,7 | 198,86 |
| 40 % | 0,1 | 1,88 | 8,82 | 0,29 | 17,33 | 93,95 |
| 50 % | 0,09 | 1,38 | 6,97 | 0,23 | 26,77 | 204,71 |
| 60 % | 0,09 | 3,6 | 4,84 | 0,16 | 15,64 | 75,96 |
| 70 % | 0,1 | 1,66 | 5,98 | 0,24 | 14,44 | 87,33 |
| 80 % | 0,08 | 2,15 | 7,1 | 0,19 | 7,04 | 63,54 |
| 90 % | 0,09 | 2,06 | 7,12 | 0,18 | 9,91 | 76,92 |
| 100 % | 0,1 | 2,75 | 6,82 | 0,17 | 11,28 | 90,63 |

Tabla 3.2. Tabla que refleja los tiempos de ejecución (en segundos) en el ejemplo burma14.

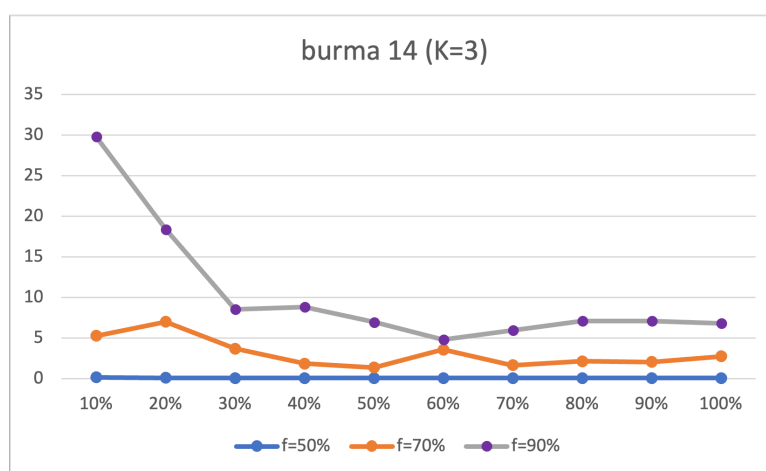


Figura 3.1. Tiempos de ejecución según el porcentaje de L y el porcentaje de f para 3 periodos.

es mayor. Además, ese aumento de nodos hace que sus variaciones sean mas grandes y en cuanto se acerca L al 20 % los tiempos de ejecución empiezan a aumentar.

En la figura 3.2 , donde se consideran 2 periodos más , se hacen aún mas visibles las difencias entre las gráficas. De la misma manera, se observa un aumento considerable en cuanto L se acerca al 10 %.

Otra cantidad a considerar, y en la que no se ha hecho hincapié, es el valor objetivo. Si se toma este mismo ejemplo, se percibe que no sufren casi ningún cambio, como se refleja en la tabla 3.1.1.

Nótese que para K=3 los tiempos de ejecución aumenten notablemente en las cercanías a L=10 %, pero los valores objetivos no varían. De una manera parecida ocurre para K=5, pero aquí si es verdad que existe un aumento de la función objetivo cuando L está al 10 %. Véase las dos soluciones para K=5 en las figuras 3.3 y 3.4.

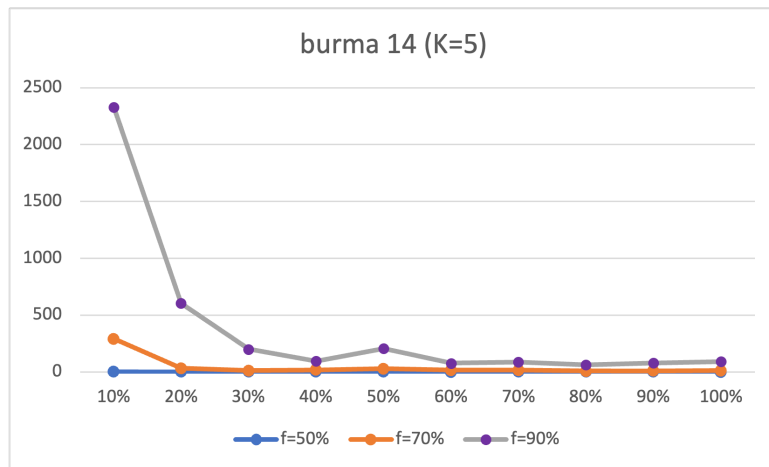


Figura 3.2. Tiempos de ejecución según el porcentaje de L y el porcentaje de f para 5 periodos.

| L | K=3 | | | K=5 | | |
|------------|--------|--------|--------|--------|--------|--------|
| | f=50 % | f=70 % | f=90 % | f=50 % | f=70 % | f=90 % |
| 10 % | 5806 | 7728 | 9239 | 10588 | 12560 | 15206 |
| 20 %-100 % | 5806 | 7728 | 9239 | 10380 | 12496 | 15179 |

Tabla 3.3. Tabla que refleja los valores óptimos en el ejemplo burma14.

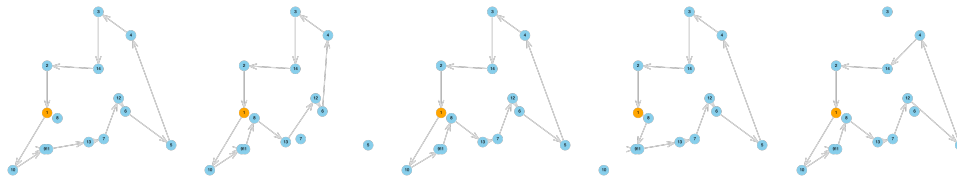


Figura 3.3. Soluciones del burma14 con $K=5$, $f=90\%$ y $L=20\%$

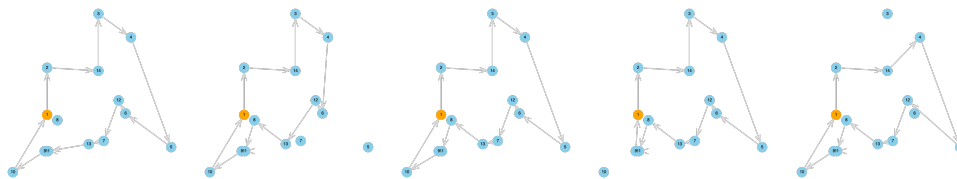


Figura 3.4. Soluciones del burma14 con $K=5$, $f=90\%$ y $L=10\%$

Las rutas en los periodos 1,2,3 y 5 son iguales para ambas constantes de consistencia, el único periodo que cambia ligeramente es el 4, y de ahí la diferencia entre los valores objetivos.

3.1.2. ulysses16

Se pretende hacer otro estudio similar al hecho para el burma14, esta vez para el ejemplo ulysses16, con el objetivo de ver como se comporta la implementación en otro ejemplo y no dejar la interpretación de los resultados al estudio en una instancia. La tabla 3.1.3 recoge los tiempos de ejecución y valores objetivos, al problema de 3 periodos y utilizando una $M=10000$. Hay que mencionar que solo se representa para $f=50\%$ y $f=70\%$, pues al 90% algunos tiempos de ejecución superan las 10 horas.

| L | Tiempo(min) | | valor objetivo | |
|-------|-------------|--------|----------------|--------|
| | f=50 % | f=70 % | f=50 % | f=70 % |
| 10 % | 0,0 | 14,7 | 11386 | 16941 |
| 20 % | 0,0 | 9,0 | 11299 | 16887 |
| 30 % | 0,0 | 5,5 | 11237 | 16451 |
| 40 % | 0,0 | 10,1 | 11237 | 16396 |
| 50 % | 0,0 | 5,3 | 11237 | 16396 |
| 60 % | 0,0 | 5,5 | 11237 | 16349 |
| 70 % | 0,0 | 12,6 | 11237 | 16349 |
| 80 % | 0,0 | 20,5 | 11237 | 16349 |
| 90 % | 0,0 | 4,6 | 11237 | 16349 |
| 100 % | 0,0 | 3,6 | 11237 | 16349 |

Tabla 3.4. Tabla que refleja los tiempos de ejecución y los valores objetivo en el ejemplo ulysses16.

Los tiempos de ejecución para $f=50\%$ son menores de 15 segundos, por ello aparecen 0 minutos. Por otro lado, nótese como con solo 2 nodos más, este ejemplo tiene unos tiempos de ejecución bastante mayores a los del burma14. Así por ejemplo para $K=3$, $L=10\%$ y $f=70\%$ burma14 tarda 29.8 segundos y ulysses16 tarda 14.7 minutos.

En cuanto al efecto de la constante de consistencia en este ejemplo, se observa que no tiene el comportamiento que nos esperábamos. Los tiempos de ejecución no parecen ajustarse a la hecho de que, con una constante menor, el problema debe ser mas difícil de resolver. Esto se puede ver mejor en la gráfica 3.5.

En cuanto al valor objetivo, en este ejemplo hay mas cambios que en el burma14. Para $f=70\%$ existen 4 soluciones distintas. Es decir, las soluciones son mas sensibles al valor de L que en burma14. Es posible que por ello hayan diferencias en los tiempos de ejecución. Las figuras 3.6 y 3.7 muestran las rutas óptimas para dos valores de L con el mismo valor de f.

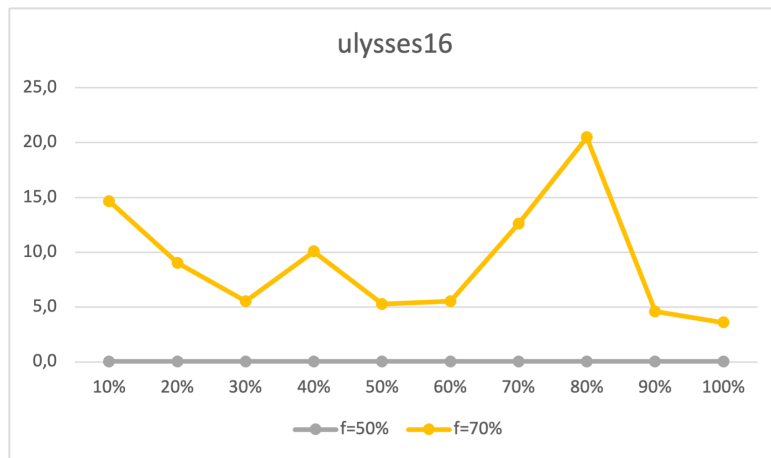


Figura 3.5. Tiempos de ejecución según el porcentaje de L y el porcentaje de f para 3 periodos de ulysses16.

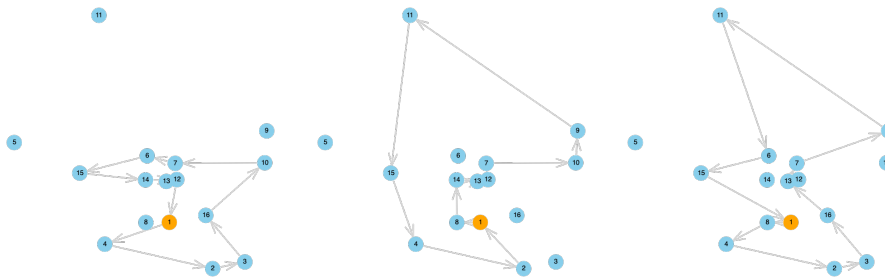


Figura 3.6. Soluciones del ulysses16 con $K=3$, $f=70\%$ y $L=100\%$

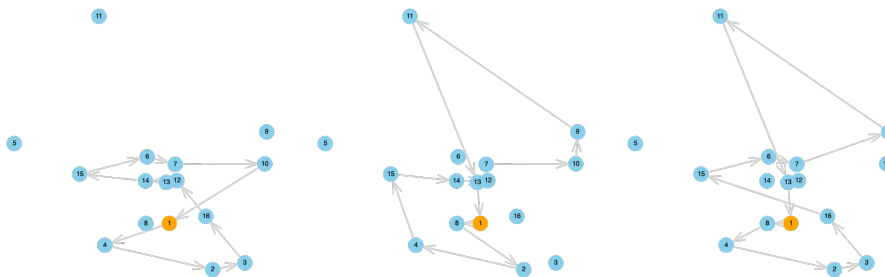


Figura 3.7. Soluciones del ulysses16 con $K=3$, $f=70\%$ y $L=10\%$

Se observa en las figuras que para una L grande las rutas no tienen cruces. Sin embargo, con la L pequeña las rutas empiezan a complicarse más y hay más cruces.

3.1.3. Ajuste de la M grande.

Hasta ahora se ha utilizado un valor para $M=10000$, que con holgura sobrepasaba los tiempos óptimos del TSP. Ahora se propone ajustar la M hasta este tiempo óptimo y comprobar si existe diferencia en los tiempos de ejecución con cuando se utilizaba $M=10000$. En primer lugar, se refleja en la tabla 3.1.3 los tiempos de ejecución con todas las instancias estudiadas para $M=\text{tsp.optimo}$. Las diferencias con la tabla 3.1 son mínimas. Lo único destacable sería con el

| Ejemplo | nodos | M | L=10 % | L=20 % | L=30 % |
|-----------|-------|------|--------|--------|--------|
| bayg29 | 29 | 1610 | 5.22 | 1.44 | 1.128 |
| bays29 | 29 | 2020 | 3,2 | 1,7 | 2,015 |
| burma14 | 14 | 3323 | 0,1 | 0,1 | 0,1 |
| fri26 | 26 | 937 | 1 | 1,1 | 0,9 |
| gr17 | 17 | 2085 | 0,4 | 0,1 | 0,1 |
| gr21 | 21 | 2707 | 0,3 | 0,2 | 0,1 |
| gr24 | 24 | 1272 | 2,84 | 0,4 | 0,24 |
| swiss42 | 42 | 1273 | 603 | 63.82 | 7,2 |
| ulysses16 | 16 | 6859 | 1,3 | 0,7 | 0,5 |
| ulysses22 | 22 | 7013 | 11 | 11,61 | 9,63 |

Tabla 3.5. Tabla que expone los tiempos de ejecución, tomando M como el tiempo óptimo del TSP. Es de 3 periodos y una $f=50\%$.

ejemplo swiss42. Aun así se comparan con los tiempos de ejecución para las dos M en los ejemplos estudiados, como se puede observar en la tabla 3.1.3.

| L | burma14 | | ulysses16 | |
|-------|-----------|--------------------|-----------|--------------------|
| | $M=10000$ | $M=\text{tsp.opt}$ | $M=10000$ | $M=\text{tsp.opt}$ |
| 10 % | 38,8 | 521,8 | 14,7 | 6,8 |
| 20 % | 10,1 | 1,0 | 9,0 | 7,7 |
| 30 % | 3,3 | 29,9 | 5,5 | 7,6 |
| 40 % | 1,6 | 2,0 | 10,1 | 4,0 |
| 50 % | 3,4 | 1,5 | 5,3 | 5,4 |
| 60 % | 1,3 | 1,0 | 5,5 | 4,9 |
| 70 % | 1,5 | 2,5 | 12,6 | 5,4 |
| 80 % | 1,1 | 1,9 | 20,5 | 5,2 |
| 90 % | 1,3 | 0,8 | 4,6 | 4,9 |
| 100 % | 1,5 | 0,8 | 3,6 | 5,4 |

Tabla 3.6. Tabla que compara dos ejemplos, burma14 con $K=5$ y $f=90\%$ y ulysses16 con $K=3$ y $f=70\%$, los tiempos de ejecución(en min) y para los dos valores de M .

Destacable es la diferencia en los valores para $L=10\%$. Pero no se decanta hacia un lado. Si se quisiera representar gráficamente la diferencia de los tiempos de ejecución en porcentajes para burma14, se acudiría a las figura 3.8.

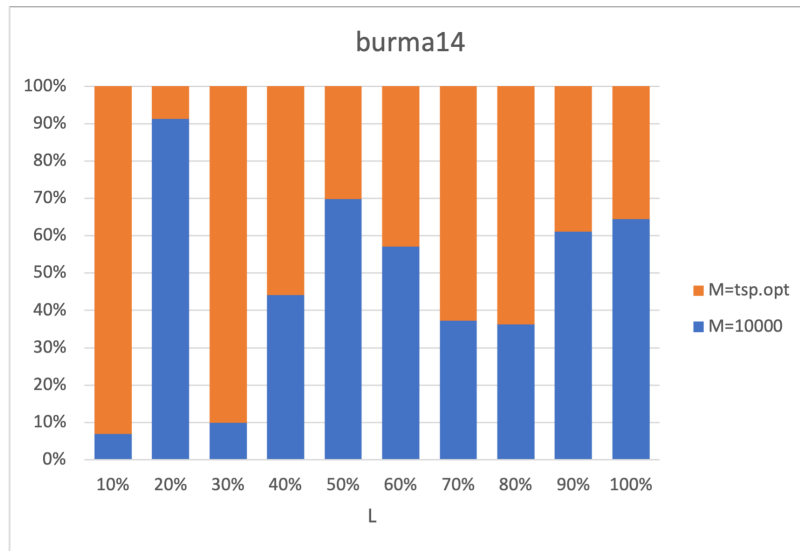


Figura 3.8. Comparación de los tiempos de ejecución en el ejemplo burma14 para los dos valores de M.

Se ve como existen grandes diferencias para valores más pequeños de L. Pero no se decanta hacia ningún lado, por lo que hay indicios de que el ajuste de la M, no tiene por qué disminuir los tiempos de ejecución.

De la misma manera, se hace la misma gráfica, pero en este caso para ulysse16 en la figura 3.9. Aquí se observa un comportamiento más regular, y no existen tantas diferencias en los porcentajes. Lo que nos indica nuevamente que un ajuste de la M no nos va a dar menos tiempos de espera.

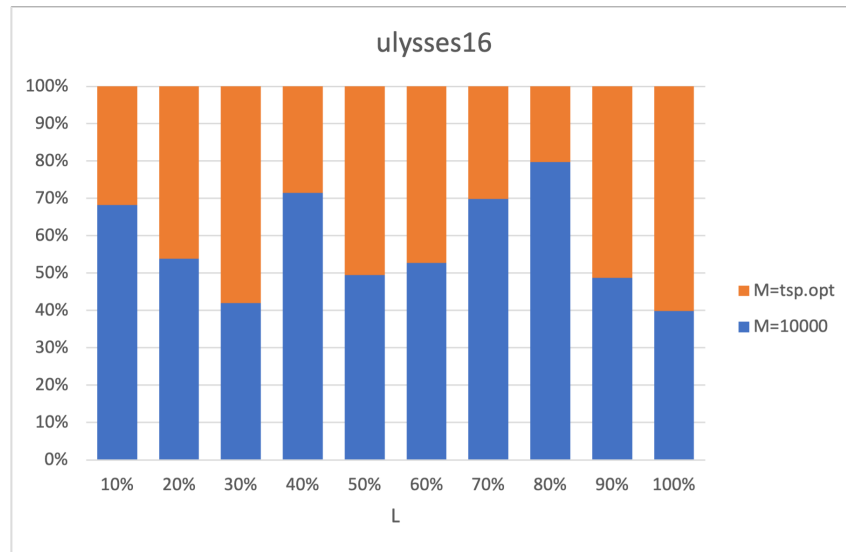


Figura 3.9. Comparación de los tiempos de ejecución en el ejemplo ulysses16 para los dos valores de M.

Conclusiones

El problema ha sido resuelto e implementado con éxito. El lenguaje Julia ha resultado una herramienta potente para este tipo de problemas y las librerías que posee han hecho posible su implementación y representación gráfica. El software optimizador tomado ha sido Gurobi, pues otros presentaban más complicaciones y algunos no estaban capacitados para resolverlo. Sin embargo su uso está restringido a una cuota, por lo que sería deseable encontrar uno gratuito.

En cuanto a los resultados de la ejecución del código se ha podido ver la dependencia del problema con la condición de consistencia, tanto en sus tiempos de ejecución como en el valor objetivo. Sin embargo, los tiempos de ejecución son viables hasta unos 20 nodos si se consideran 3 periodos. Desde que pasan de esa cantidad, la ejecución del código se vuelve muy lenta, llegando a tardar incluso días. Lo que abre dos puertas, una porque la complejidad computacional del problema sea muy grande, y otra porque el código no está lo suficientemente optimizado.

En el estudio de los tiempos de ejecución y valores óptimos no se han considerado tiempos de espera en los nodos, algo que podría ser analizado con facilidad cambiando las restricciones del programa. Además tampoco se han estudiado las variantes mencionadas, donde se cambiaba la función objetivo. De la misma manera, estos valores podrían compararse con la solución del TSP normal y ver cual es el precio que hay que pagar por la consistencia.

Se podría también proponer estudiar el ConVRP (Consistent Vehicle Routing Problem), o Problema de Enrutamiento de Vehículos Consistente mediante Julia y ver también como se comportan los tiempos de ejecución en este problema.

A

Apéndice

A.1. Distribuciones VMI

La gestión del inventario por el proveedor o VMI (Vendor Managed Inventory), es un modelo de negocio en el que el cliente proporciona información al proveedor y éste asume la responsabilidad de mantener un adecuado nivel de inventario de sus productos, llegando incluso a gestionar todo el ciclo de pedido y aprovisionamiento en su nombre. Es una forma de optimizar el rendimiento de la cadena de suministro, ya que es el fabricante el responsable de mantener los niveles de inventario al distribuidor. El fabricante tiene acceso a los datos de inventario del distribuidor y es responsable de generar las órdenes de compra.

A.2. Ciclo Hamiltoniano

Definición: En un grafo , se dice ciclo Hamiltoniano a todo aquel camino cerrado que pasa una sola vez por todos los nodos (vértices) del grafo.

A.3. Código utilizado

En el siguiente enlace <https://github.com/Felixlds/conTSP.git> se encuentran los dos códigos en Julia utilizados para la resolución del ConTSP . Uno se utiliza para generar nodos aleatorios en un plano y resolver el problema sobre él. El otro está mas pensado para resolver el ConTSP utilizando la librería TSPLIB.

Bibliografía

- [1] SALAZAR GONZÁLEZ, Juan José. *Lecciones de optimización*. Servicio de publicaciones Universidad de La Laguna, 2000.
- [2] ANIRUDH SUBRAMANYAM y CHRYSANTHOS E. GOUNARIS A branch-and-cut framework for the consistent traveling salesman problem *European Journal of Operational Research*, 2016, vol. 248, pp. 384–395.
- [3] KOVACS, A. A., GOLDEN, B. L., HATRL, R. F., y PARRAGH, S. N. Vehicle routing problems in which consistency considerations are important: A survey. *Networks*, 2014, vol. 64 , pp. 192–213
- [4] GERHARD REINELT. *TSPLIB 95*, Universität Heidelberg , 1991. Disponible en: <http://comopt.ifl.uni-heidelberg.de/software/TSPLIB95/tsp95.pdf>.
- [5] CALVIÑO MARTINEZ , Aida. *Cooperación en los problemas del viajante (TSP) y de rutas de vehículos (VRP): una panorámica.*, Universidad de Santiago de Compostela, 2011. Disponible en:http://eio.usc.es/pub/mte/descargas/proyectosfinmaster/proyecto_762.pdf
- [6] Gurobi Optimization. *Gurobi Optimizer [Software de optimización]*. 2023. <https://www.gurobi.com>.

The Consistent Travelling Salesman Problem

Félix Xavier López Da Silva

Facultad de Ciencias • Sección de Matemáticas

Universidad de La Laguna

alu0101124971@ull.edu.es

Abstract

The Consistent Traveling Salesman Problem (ConTSP) is a variation of the TSP that incorporates time periods and requires time consistency. Its applications in society and the market are significant, such as in healthcare services and inventory management models. By using the Julia programming language and the Gurobi optimization software, a code is created to solve this problem. One of the objectives of this work is to observe the code's execution times in order to draw conclusions about the problem and the programming language. Additionally, a library is used that provides examples commonly used in these types of studies, and two of these examples are compared for a more in-depth analysis of the problem.

1. Introduction

Let's imagine that I am a supplier who has a portfolio of 15 clients to visit in 3 days. Each day, I have to visit certain specified clients, and it is required that at least one client be visited more than once within these three days. However, in order to provide better service to my clients, I want to visit them at approximately the same time on the day assigned for their visit. In other words, I don't want there to be a time difference of more than two hours between two consecutive visits to the same client. In this way, how can I minimize the number of kilometers traveled?

2. Solution

A mathematical model for the generalization of the previous problem would be:

$$\min \sum_{p \in \mathcal{P}} \sum_{(i,j) \in A_p} c_{ij} x_{ij}^p \quad (1)$$

s.t:

$$\sum_{j \in \delta_p^+(i)} x_{ij}^p = 1 \quad i \in V_p \cup \{0\}, \quad p \in \mathcal{P} \quad (2)$$

$$\sum_{j \in \delta_p^-(i)} x_{ij}^p = 1 \quad i \in V_p \cup \{0\}, \quad p \in \mathcal{P} \quad (3)$$

$$u_j^p \geq u_i^p + t_{ij} x_{ij}^p - M(1 - x_{ij}^p) \quad i \in V_p, \quad j \in V_p \cup \{0\}, \quad p \in \mathcal{P} \quad (4)$$

$$u_j^p \leq u_i^p + t_{ij} x_{ij}^p + M(1 - x_{ij}^p) \quad i \in V_p, \quad j \in V_p \cup \{0\}, \quad p \in \mathcal{P} \quad (5)$$

$$u_i^k - u_i^l \leq L \quad k, l \in \mathcal{P}, \quad i \in V_k \cap V_l \quad (6)$$

$$x_{ij}^p \in \{0, 1\} \quad (i, j) \in A_p, \quad p \in \mathcal{P} \quad (7)$$

By implementing it in Julia and using the Gurobi optimizer, we were able to find solutions to this problem, which for the given specific example could be:

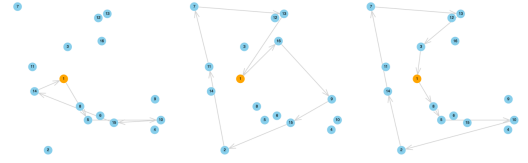


Figure 1: Tours for days 1, 2 and 3.

3. The time spent and optimal cost

The time spent on program execution and the objective value are important variables to study in this type of work. Therefore, the TSPLIB library is chosen to run certain examples from its database and observe how the execution times and optimal values behave in relation to the number of nodes, consistency constraint, and number of periods.

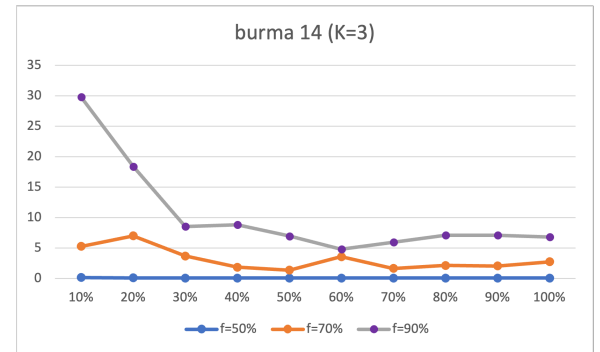


Figure 2: For the example burma14, with 3 periods and a probability f of node existence.

References

- [1] ANIRUDH SUBRAMANYAM y CHRYSANTHOS E. GOUNARIS A branch-and-cut framework for the consistent traveling salesman problem *European Journal of Operational Research*, 2016, vol. 248, pp. 384–395.
- [2] KOVACS, A. A., GOLDEN, B. L., HATRL, R. F., y PARRAGH, S. N. Vehicle routing problems in which consistency considerations are important: A survey. *Networks*, 2014, vol. 64, pp. 192–213
- [3] GERHARD REINELT, *TSPLIB95*, Universitat Heidelberg, 1991. Disponible en: <http://comopt.ifi.uni-heidelberg.de/software/TSPLIB95/tsp95.pdf>.