



**Sección de Matemáticas**  
Universidad de La Laguna

Jorge Brito Dávila

# *Optimización de rutas turísticas*

Optimitation of tourist routes

Trabajo Fin de Grado  
Grado en Matemáticas  
La Laguna, Marzo de 2020

DIRIGIDO POR

*José Andrés Moreno Pérez*

*José Andrés Moreno Pérez*  
*Ingeniería Informática y de*  
*Sistemas*  
*Universidad de La Laguna*  
*38200 La Laguna, Tenerife*

---

## Agradecimientos

A mi madre, a mi padre, a mi hermano y al resto de la familia. Gracias a ellos por ser la persona que soy, y para los que solo tengo palabras de agradecimiento por su apoyo incondicional en los buenos y malos momentos, son la principal fuente de consejos y un espejo al que mirar.

A mis tíos laguneros y mi prima. Por los momentos compartidos en Tenerife y por darme todas las facilidades posibles.

A mis amigos. Por estar siempre conmigo y saber que nunca fallan, son personas increíbles y les agradezco todos los momentos inolvidables que me regalan.

A todas las amistades que me ha dado la Universidad. Han sido mi apoyo en Tenerife y han hecho que el camino fuese más fácil. También quiero recordar a las amistades que me llevo del Erasmus, a las que tengo un cariño especial de un año inolvidable.

A mi tutor, José Andrés Moreno Pérez, por facilitarme su tiempo, orientación y apoyo para realizar este trabajo.

En general, a todas las personas que trabajan para la Universidad de La Laguna y hacen que este sea un lugar para el encuentro y la formación de mejores personas.

Jorge Brito Dávila  
La Laguna, 13 de marzo de 2020



---

## Resumen · Abstract

### *Resumen*

---

*Es una prioridad para las empresas del sector turístico satisfacer las necesidades específicas de los turistas mediante la programación de actividades personalizadas con la definición de rutas turísticas, la elaboración del presupuesto y la administración adecuada de sus recursos y tiempos, respondiendo a sus exigencias de calidad y flexibilidad. A través de la utilización de herramientas matemáticas, dentro de la Informática, podemos ofrecer soluciones aplicadas a la economía de servicios y concretamente al diseño de estas rutas. Este trabajo propone un modelo matemático para el diseño de las rutas, con la definición de parámetros, variables, objetivos y restricciones. Así como un procedimiento simple para encontrar soluciones aplicando el modelo y usando la herramienta Gusek. Para la experimentación utilizamos datos reales y mostramos algunos resultados y conclusiones.*

**Palabras clave:** *Rutas turísticas – Optimización combinatoria – Gusek – ...*

### *Abstract*

---

*It's a priority for companies in the tourism sector to meet the specific needs of tourists by programming personalized activities with the definition of tourist routes for budgeting and adequate administration of resources and time, seeking to meet their demands for quality and flexibility. Through the use of mathematical tools, belonging to the field of Computation we can offer solutions applied to the service economy and specifically to the planning of these tourist routes. We will put into operation a mathematical model with the design of an algorithm after defining variables that describe the interests of the tourist, showing the results and conclusions of the model.*

**Keywords:** *Tourist routes – Combinatorial optimization – Gusek – ...*



---

# Contenido

<b>Agradecimientos</b> .....	III
<b>Resumen/Abstract</b> .....	V
<b>Contenido</b> .....	VII
<b>Introducción</b> .....	IX
<b>1. Diseño y planificación de rutas turísticas</b> .....	1
1.1. Innovación en el Turismo .....	1
1.2. Planificación de rutas turísticas .....	3
<b>2. Optimización de rutas turísticas</b> .....	5
2.1. Problemas de Optimización de Rutas Turísticas .....	5
2.2. Formalización del problema .....	6
2.3. Procedimiento de solución exacta .....	10
2.4. Procedimiento heurístico .....	11
2.4.1. Lectura de datos de entrada .....	11
2.4.2. Modelo de optimización de rutas .....	14
2.4.3. Reducción de los beneficios.....	16
<b>3. Experimentación</b> .....	21
3.1. Datos .....	21
3.2. Resultados.....	24
<b>Conclusiones</b> .....	40
<b>Lista de Figuras</b> .....	41

<b>Lista de Tablas</b> .....	43
<b>Bibliografía</b> .....	45
<b>Poster</b> .....	47



---

## Introducción

La finalidad de este trabajo consiste en mostrar cómo la aplicación de la programación combinatoria en conjunción con técnicas informáticas de Inteligencia Artificial permite afrontar con eficiencia el diseño y planificación de rutas turísticas. Las técnicas de Programación Matemática aplicada a los problemas que surgen en este contexto consiguen obtener soluciones óptimas solo en problemas de pequeño tamaño. No obstante, se pretende que sirvan como ayuda al diseño de viajes turísticos en aplicaciones prácticas realistas y sería ideal combinarlas con otras metodologías para conseguir simultáneamente una mayor eficiencia del tiempo y satisfacción del viajero.

Los productos y servicios turísticos ofertados por un destino inteligente deben ofrecer al cliente la posibilidad de aprovechar eficientemente su tiempo en vacaciones. Estos recursos comprenden una serie de actividades donde el viajero es a la vez el agente pasivo disfrutando de dicha oferta y el agente activo que toma sus decisiones. Por tanto, es conveniente facilitar y promover el uso de aplicaciones que le orienten y ayuden en esta toma de decisiones evaluando de forma continua las posibilidades que se le ofrecen y su combinación en itinerarios óptimos para aprovechar su tiempo de vacaciones.

Los recursos tecnológicos para apoyar la elección óptima de las actividades en viajes de turismo y su planificación eficiente se ha convertido en un interesante tema de investigación y de desarrollo tecnológico en los últimos años. El turista utiliza los recursos tecnológicos y la información disponible para planificar las actividades a realizar en el viaje de tal forma que este tenga la mayor satisfacción posible. De esta manera, pueden visitar los lugares más interesantes según sus preferencias aprovechando todo el tiempo que dispongan realizando uno o varios itinerarios, según los días de estancia.

Este tipo de problemas de planificación de rutas óptimas son de interés para el campo de la Investigación Operativa, donde se suele acudir a la programación matemática para su solución. También son de interés en la Ingeniería del Software, donde se implementan desarrollos informáticos que filtran y organizan la información. Desde el enfoque de la Inteligencia Artificial se trata de combinar estos procedimientos con las estrategias de desarrollo de los Sistemas Expertos para obtener aplicaciones de utilidad práctica. La idea básica consiste en implementar programas informáticos que interactúen con los modelos de programación matemática para plantear y resolver los problemas de optimización que reflejan los intereses del turista de forma eficiente.

Por un lado, se trata de buscar la solución óptima de una función objetivo utilizando una serie de variables y restricciones para adaptarse a la situación real que se plantee en cada caso, y por otro de idear mecanismos automatizados de establecer los valores de los parámetros que representan dicha situación y ofrecer los resultados de forma útil.

En este proyecto nos centraremos en el uso de la información disponible de forma abierta en Internet para establecer el problema de optimización que refleja los intereses del turista y a su vez utilizar los resultados del mismo para plantear un nuevo problema. Se identifican así varias variantes de los modelos conocidos como *Orienteering Problems*. El modelo más simple persigue seleccionar los puntos de interés a visitar entre todos los disponibles, generando una ruta de acuerdo al interés y preferencias del turista. Se puede aplicar varias veces, secuencialmente para ir obteniendo los itinerarios para los días disponibles. De esta forma, conseguimos evitar la resolución del problema conjunto de toda la estancia que daría lugar a al uso del modelo TOP (*Team Orienteering Problem*) que pretende obtener simultáneamente la planificación de varias rutas. Incluso la versión más sencilla de este problema resulta muy difícil para ser afrontado con las técnicas usuales de programación matemática usadas en Investigación Operativa.

La novedad que se propone en este modelo es clasificar los puntos de interés por categorías. Es decir, en vez de darle una puntuación a cada punto de interés como suele ser habitual, se establecerá una puntuación para cada subconjunto de puntos.

Dicha novedad se explica en detalle en el presente proyecto, el cual se encuentra estructurado en tres capítulos. En el primero se dará una visión sucinta del turismo en general, de la importancia que tiene para mantener su desarrollo incluir propuestas innovadoras para complementar los servicios, y el interés de la aplicación de modelos matemáticos y los avances tecnológicos para la obtención de rutas turísticas óptimas. Se analizará también los avances y aplicaciones desa-

rrolladas en el sector turístico.

En el segundo capítulo se analiza la aplicación de los modelos matemáticos usuales para la resolución de estos problemas de obtención de rutas y se introducirá una breve descripción de los mismos y su formulación. Sumado a esto, se describe algunas de las metodologías empleadas para resolver estos problemas usando los modelos descritos.

El tercer capítulo, se recoge la experimentación desarrollada para resolver el modelo matemático definido para una ruta turística, utilizando el modelo OP propuesto. Incluyendo una evaluación y análisis de las soluciones y resultados del modelo. Este capítulo describe la aplicación combinada de los modelos y de la herramienta Gusek con pequeños programas para resolver el problema con datos de un caso real.

El trabajo finaliza con las conclusiones y las propuestas de trabajos futuros para extender este tipo de estudio.



## Diseño y planificación de rutas turísticas

En este capítulo se contextualiza el problema en las necesidades de innovación del sector turístico y su transformación digital de servicios turísticos.

### 1.1. Innovación en el Turismo

El turismo en España es un sector muy importante para la economía nacional, particularmente en Canarias es el sector económico principal, situándose como la primera región turística europea. En las Islas Canarias en el año 2018 se recibieron 15,56 millones de turistas, generando el 35,2% del Producto Interior Bruto, acogiendo el 40,2% del empleo. Es una de las industrias con mayor crecimiento y con gran posibilidad de crear empleo, ya que engloba distintos ámbitos de la vida cotidiana: “El turismo es un hecho social, humano, económico y cultural”. Esta información se encuentra en el documento del “Plan de Canarias para el Turismo 2025”<sup>1</sup>.

España y específicamente Canarias es líder indiscutible como destino vacacional en producto “sol y playa”, con una amplia oferta paralela en ocio, diversión, cultura y deportes. Su crecimiento a lo largo del tiempo ha estado basado en el control del volumen y producción, y con gran dependencia de las políticas estratégicas de los posicionamientos de los grupos empresariales turísticos globales o touroperadores. No obstante, es necesario mantener la exigencia competitiva y adaptarse a las reglas que el mercado impone en cada momento. Paralelamente se debe diversificar la propuesta de valor con productos que extiendan el hecho turístico más allá de la costa, como son el turismo de interior, gastronómico o el ecoturismo así como incorporar un eje de transformación competitiva que

---

<sup>1</sup> [www.gobiernodecanarias.org/cmsgobcan/export/sites/turismo/dir\\_gral\\_ordenacion\\_promocion/2\\_BASE\\_DOC\\_PECT2025\\_ANTECEDENTES.pdf](http://www.gobiernodecanarias.org/cmsgobcan/export/sites/turismo/dir_gral_ordenacion_promocion/2_BASE_DOC_PECT2025_ANTECEDENTES.pdf).

impulse una estrategia digital en el nuevo entorno turístico.

La industria turística tiene que impulsar líneas de actuación desarrollando planes integrales de calidad para destinos turísticos, oferta de servicios y coordinación de productos. Reforzar las estrategias de marketing digital adaptando la regulación existente al nuevo entorno turístico sostenible. Además, implementar instrumentos de recogida de datos y análisis sobre la calidad del servicio y el grado de satisfacción de los clientes. Esta referencia se puede localizar en el documento de Directrices Generales de la Estrategia de Turismo Sostenible de España 2030 <sup>2</sup>.

En este sentido, la implantación de las nuevas tecnologías se hace imprescindible en el sector turístico. El uso de la Inteligencia Artificial con el empleo de “chatbots” para ofrecer información y asistencia a viajeros, es una solución muy utilizada actualmente en turismo. Así como el empleo de robots como asistentes de viaje, sistemas de identificación de viajeros a través de reconocimiento facial para acceder a las habitaciones o a ciertos servicios. También destacan los guías turísticos virtuales o servicios predictivos para ofertas y viajes que harán una experiencia totalmente personalizada y ágil.

El potencial que actualmente supone esta transformación digital y la capacidad de procesamiento de una gran cantidad de datos hacen que cada día podamos poner en funcionamiento sistemas cada vez más eficaces para detectar patrones, automatizando procesos y que agilicen la toma de decisiones, y es ahí donde intervienen la optimización matemática.

Los turistas necesitan planificar sus vacaciones, no solo antes de su estancia demandan información para planear y elegir entre múltiples opciones, sino que también durante el viaje. Con el desarrollo de las nuevas tecnologías de la información y de la comunicación, pueden acceder a distintas páginas webs, aplicaciones y guías turísticas, lo que les permite tener una idea de cómo programar sus viajes. En cierto modo, lo que se quiere es que estas guías se elaboren con la ayuda de la optimización matemática, ya que a través de ella logramos aconsejar a los turistas de forma que queden lo más satisfechos posible.

El alcance que tiene este trabajo de investigación es que el turista administre eficazmente sus recursos y encuentre la ruta adecuada alcanzando su máximo disfrute. El trabajo propone solucionar un problema de un itinerario de varios días sugiriendo qué puntos visitar, así como la secuencia de los mismos que incluye el recorrido de la ruta y los tiempos en realizarla.

---

<sup>2</sup> <https://turismo.gob.es/es-es/estrategia-turismo-sostenible/Documents/directrices-estrategia-turismo-sostenible.pdf>.

## 1.2. Planificación de rutas turísticas

Con la creciente facilidad en el acceso a Internet, los turistas cada vez están más interesados en realizar viajes de manera independiente. Incluso aprenden, tienen mayor información del destino y desean establecer una planificación diaria de acuerdo a sus intereses. Con la navegación y la geolocalización, el turista desea encontrar una ruta que en base a una definición formal de tipos, propiedades y relaciones variables saquen el mayor provecho de su viaje. Basándonos en situaciones reales, podemos aplicar herramientas matemáticas y algorítmicas, asentadas en un modelo matemático y las estrategias de Inteligencia Artificial, que puedan dar solución a los problemas planteados de obtener las rutas que mejor satisfacen los intereses del turista.

En este contexto surgen problemas de optimización que modelizan las tareas a las que se enfrenta el sistema al dar respuesta a estas situaciones. Generalmente, los problemas de optimización están estructurados en varias partes. Las decisiones a tomar se determinan por los valores asignados a un conjunto de variables; llamadas variables de decisión. La función objetivo establece la forma de realizar la evaluación de las propuestas de solución a partir de estas variables y que se pretende minimizar o maximizar. Además existen unas restricciones, que son las condiciones que deben verificar los valores asignados a las variables para que den lugar una elección factible. En los problemas en los que se trata de seleccionar rutas destaca el Problema del Vendedor o *Travelling Salesman Problem* (TSP), donde un vendedor tiene que visitar un conjunto determinado de nodos para ofertar sus productos por lo que se busca la ruta más corta. Otro problema básico de rutas son los Problemas de Rutas de Vehículos o *Vehicle Routing Problems* (VRP), donde se cuenta con un número determinado de vehículos para trazar las rutas correspondientes con las que visitar un conjunto de clientes. Ambos modelos en general tienen como objetivo minimizar una función que representa distancia o tiempo total empleado.

Sin embargo, los turistas que viajan a una ciudad o región, a menudo les es imposible visitar todo lo que les interesa. Por lo tanto, tienen que seleccionar aquellas atracciones que consideren más valiosas y hacer un plan factible para recorrerlas en el tiempo disponible es generalmente una tarea complicada. En este proyecto trataremos de resolver un problema conocido como *Tourist Trip Design Problems* (TTDP) donde el turista desea visitar los puntos de mayor interés en un tiempo limitado que le impide recorrer todos ellos.

En el TTDP se busca obtener las rutas de más provecho posible sin sobrepasar los límites, así que se enfocan en maximizar la satisfacción o beneficio que se obtiene al visitar los puntos. Estos problemas están relacionados con el *Orienteering Problem* (OP) o el *Team Orienteering Problem* (TOP), que son problemas

similares a los que mencionamos anteriormente (TSP y VRP). Aunque en este caso se trata de maximizar el beneficio obtenido con las visitas sujetos a una limitación de tiempo. El OP se corresponde con el TSP puesto que se quiere planear la ruta de un solo día, mientras que en el TOP se quieren planear las rutas para varios días por lo que se corresponde con VRP, ya que lo que en el VRP son vehículos en el TOP son itinerarios turísticos. Estos modelos también cuentan con las correspondientes variantes al considerar intervalos temporales (*Time Windows*) para los puntos de interés que pueden ser visitados, los cuales se denominan *Orienteering Problem with Time Windows* (OPTW) o *Team Orienteering Problem with Time Windows* (TOPTW), respectivamente.



## Optimización de rutas turísticas

En este capítulo se analizan los problemas de optimización que surgen en el diseño de rutas turísticas y cómo se abordan en los sistemas de planificación y optimización de los itinerarios de los turistas.

### 2.1. Problemas de Optimización de Rutas Turísticas

La mayoría de los problemas de optimización de rutas turísticas o *Tourist Trip Design Problems* (TTDP) que surgen en el diseño y planificación de rutas turísticas se derivan del problema básico conocido por *The Orienteering Problem* (OP). Este problema nace de un juego deportivo de orientación. Los competidores parten de un punto de salida y tienen que realizar un recorrido pasando por otros puntos de control volviendo así al punto inicial o acabando en otro punto final prefijado. Para efectuar dicho recorrido se tiene un límite de tiempo y cada punto de control tiene una puntuación, por lo que el competidor debe elegir qué puntos de control del juego recorrer. Ganará el competidor que consiga obtener la mayor puntuación ajustándose al tiempo acordado.

Los problemas de optimización combinatoria pueden ser resueltos por métodos exactos o por métodos heurísticos y metaheurísticos. Han sido varios los investigadores que han propuesto algoritmos exactos para resolver el OP. Laporte and Martello (1990) [2] y Ramesh et al. (1992) [5] usaron el método de *Branch and Bound* (Ramificación y Acotamiento), un algoritmo diseñado para la resolución de modelos de programación entera donde se generan cotas de forma recursiva para la obtención de las variables de decisión.

Por otra parte, muchos investigadores proponen algoritmos heurísticos para la resolución del problema. Uno de ellos, Tsiligirides (1984) [7] que plantea dos modelos, uno determinista y otro estocástico. Además, comprueba su eficiencia

a través de numerosos problemas ilustrativos y realiza una comparación de ambos. Entre las investigaciones de este siglo destaca la presentada por M. Schilde (2009) [6]. Ellos desarrollaron técnicas de solución heurística para el problema de orientación multi-objetivo. La motivación surge del problema de planificar rutas turísticas individuales en una ciudad. Cada punto de interés en una ciudad proporciona diferentes beneficios para diferentes categorías (por ejemplo, museos, iglesias). Para determinar todas las soluciones óptimas aplican dos técnicas de búsqueda metaheurísticas. Utilizan el algoritmo de optimización de colonias de hormigas de *Eficiencia de Pareto* y el método de búsqueda de vecindad variable. Ambos métodos se hibridan con procedimientos de enlace de ruta. El rendimiento de los dos algoritmos se prueba en varias instancias de referencia, así como en instancias del mundo real de diferentes regiones austriacas y las ciudades de Viena y Padua.

Actualmente, existen múltiples apps que nos ayudan a preparar nuestro viaje. Una de ellas es *Visitacity*, una aplicación que planifica viajes a ciudades para un máximo de 7 días. Tiene registrada unas 1500 ciudades de todo el mundo con las rutas programadas. Asimismo, te permite seleccionar el hotel de alojamiento y eliminar los puntos de interés que no te llamen la atención.

## 2.2. Formalización del problema

Para afrontar un problema de optimización es necesario establecer su formulación matemática usando variables, restricciones y la función objetivo. Para la planificación de las rutas turísticas la información más relevante es la que hace referencia a los puntos de interés o *Points Of Interest* (POI) que incluyen las atracciones y servicios turísticos y otros puntos que el turista debe contemplar en su ruta. En los problemas de optimización que surgen en el diseño de rutas turísticas *Tourist Trip Design Problems* (TTDP) se usan los siguientes datos o parámetros:

- $n \in \mathbb{N}$ . Número puntos de interés (POIs).
- $V = \{1, \dots, n\}$ . Conjunto de POIs.
- $T_{max} \in \mathbb{R}^+$ . Tiempo máximo disponible para realizar la ruta turística.
- $b_i \in \mathbb{R}^+, \forall i \in V$ . Beneficio que aporta visitar el POI  $i$ .
- $t_{ij} \in \mathbb{R}^+, \forall i, j \in V$ . Tiempo que se tarda en viajar desde el POI  $i$  al POI  $j$ .
- $d_i \in \mathbb{R}^+, \forall i \in V$ . Duración de la visita el POI  $i$ .

Las variables que permiten especificar las rutas o itinerarios de los turistas y valorar su factibilidad y optimización son las siguientes:

- $x_{ij} \in \{0, 1\}$ . Variable binaria que adquiere el valor 1 cuando la ruta pasa del POI  $i$  al POI  $j$ , y el valor 0 en otro caso.

- $y_i \in \{0, 1\}$ . Variable binaria que adquiere el valor 1 si el POI  $i$  es visitado, y el valor 0 en otro caso.
- $u_i \in \{0, 1, \dots, n\}$ . Variable de posición. Nos indica en qué puesto va a ser visitado el POI  $i$  dentro de la ruta.

La función objetivo a utilizar debe reflejar el grado de satisfacción o beneficio alcanzado por el turista que generalmente se evalúa sumando el beneficio o puntuación de los puntos de interés incluidos en la ruta. La función a maximizar es:

$$f(x, y, u) = \sum_{i=1}^n b_i y_i$$

Las restricciones que garantizan que las asignaciones de valores a las variables  $x_{ij}$ ,  $y_i$  y  $u_i$  corresponden a rutas que se pueden llevar a cabo son las siguientes:

- Cada ruta debe empezar y terminar en el POI 1.
- Un POI puede ser visitado como máximo una vez excepto el punto de partida.
- Se debe asegurar la conectividad en cada ruta.
- No debe superarse el tiempo permitido  $T_{max}$ .

En el problema básico del *Orienteering Problem* se formula partiendo de un conjunto de  $n$  POIs dado, cada uno de los cuales tiene un *score* o beneficio  $b_i$ . En nuestro caso, formularemos el problema de tal manera que el punto de partida (POI 1) será también el punto final. Asimismo, se conocen todos los tiempos  $t_{i,j}$  necesarios para viajar de un POI  $i$  a un POI  $j$ . Además, no todos los POIs pueden ser visitados ya que existe un tiempo disponible limitado dado por  $T_{max}$ . El fin de este problema es determinar la ruta, atendiendo a que tenemos un tiempo limitado para recorrerla, obteniendo el máximo beneficio posible. Cabe destacar que cada POI solo podemos visitarlo una vez. Haciendo uso de la notación anterior, podemos formularlo como un problema programación combinatoria.

Maximizar:

$$f(x, y, u) = \sum_{i=2}^n b_i y_i \quad (2.1)$$

sujeto a:

$$\sum_{j=2}^n x_{1j} = \sum_{i=2}^n x_{i1} = 1, \quad (2.2)$$

$$\sum_{i=1}^n x_{ik} = \sum_{j=1}^n x_{kj} \leq 1; \forall k = 2, \dots, n, \quad (2.3)$$

$$\sum_{i=1}^n \sum_{j=1}^n t_{ij} x_{ij} + \sum_{i=1}^n d_i y_i \leq T_{max}, \quad (2.4)$$

$$2 \leq u_i \leq N; \forall i = 2, \dots, n, \quad (2.5)$$

$$u_i - u_j + 1 \leq (n - 1)(1 - x_{ij}); \forall i, j = 2, \dots, n, \quad (2.6)$$

$$x_{ij} \in \{0, 1\}; \forall i, j = 1, \dots, n, \quad (2.7)$$

$$y_i \in \{0, 1\}; \forall i = 1, \dots, n, \quad (2.8)$$

$$u_i \in \{1, 2, \dots, n\}; \forall i = 1, \dots, n. \quad (2.9)$$

La función objetivo (2.10) es la que maximiza la puntuación total coleccionada. La restricción (2.2) garantiza que el camino empiece y termine en el POI 1. La siguiente, (2.3), asegura la conectividad de los caminos y garantiza que cada POI sea visitado como mucho una vez. La tercera restricción (2.4) obliga a que nos ajustemos al tiempo que tenemos disponible. La cuarta (2.5) y la quinta (2.16) previenen la formación de subciclos. Es decir, sin ellas es posible que encuentre una solución óptima donde todos los POIs se visiten una vez pero sin la existencia de una ruta que recorra todos, dejando un subconjunto de POIs de la solución aislado (conectado entre sí cíclicamente). Estas restricciones para la eliminación de los subciclos están formuladas acorde con la formulación de *Miller-Tucker-Zemlin (MTZ)* en el TSP presentada en [4]. Las últimas restricciones, (2.7), (2.8) y (2.9) indican la naturaleza de las variables  $x_{ij}$  e  $y_i$  son binarias y las variables  $u_i$  son enteras.

Cuando la estancia del turista en el destino incluye varios días es necesario planificar la ruta de cada día. La extensión usual para estos casos del problema OP básico es el TOP. The *Team Orienteering Problem (TOP)* es una extensión del *OP* que optimiza la planificación de varios jugadores de equipo. El objetivo es determinar el conjunto de rutas, que están limitadas por un  $T_{max}$ , y maximizar el beneficio total obtenido por cada uno de los integrantes del equipo. Para este problema definiremos el siguiente conjunto de los caminos:  $U = \{1, 2, \dots, P\}$ . Donde  $p$  es un elemento del conjunto  $U$  y  $P$  es el número máximo de caminos que hay. En la planificación de rutas turísticas el índice  $p$  corresponde a cada uno de los días de estancia.

Para facilitar la comprensión al lector, las variables de decisión serían de la siguiente manera:

- $x_{ij}^p \in \{0, 1\}$ . Variable binaria que adquiere el valor 1 cuando la ruta  $p$  pasa del POI  $i$  al POI  $j$  por el camino  $p$ .
- $y_i^p \in \{0, 1\}$ . Variable binaria que adquiere el valor 1 cuando el POI  $i$  es visitado por el camino  $p$ .
- $u_i^p \in \{0, 1, \dots, n\}$ . Variable de posición. Nos indica en qué puesto va a ser visitado el POI  $i$  dentro de la ruta  $p$ .

El problema de optimización se formula, utilizando los parámetros y datos ya usados en el OP, de la siguiente manera:

Maximizar:

$$f(x, y, u) = \sum_{i=2}^n \sum_{p=1}^P b_i y_i^p \quad (2.10)$$

sujeto a:

$$\sum_{p=1}^P \sum_{j=2}^n x_{1j}^p = \sum_{p=1}^P \sum_{i=2}^n x_{i1}^p = P, \quad (2.11)$$

$$\sum_{p=1}^P y_i^p \leq 1; \forall i = 2, \dots, n, \quad (2.12)$$

$$\sum_{i=1}^n x_{ik}^p = \sum_{j=1}^n x_{kj}^p = y_k^p; \forall k = 2, \dots, n; \forall p = 1, \dots, P, \quad (2.13)$$

$$\sum_{i=1}^n \sum_{j=1}^n t_{ij} x_{ij}^p + \sum_{i=1}^n d_i y_i^p \leq T_{max}; \forall p = 1, \dots, P, \quad (2.14)$$

$$2 \leq u_i^p \leq n; \forall i = 2, \dots, n; \forall p = 1, \dots, P, \quad (2.15)$$

$$u_i^p - u_j^p + 1 \leq (n - 1)(1 - x_{ij}^p); \forall i, j = 2, \dots, n, \forall p = 1, \dots, P, \quad (2.16)$$

$$x_{ij}^p \in \{0, 1\}; \forall i, j = 1, \dots, n, \forall p = 1, \dots, P, \quad (2.17)$$

$$y_i^p \in \{0, 1\}; \forall i = 1, \dots, n, \forall p = 1, \dots, P, \quad (2.18)$$

$$u_i^p \in \{1, 2, \dots, n\}; \forall i = 1, \dots, n, \forall p = 1, \dots, P. \quad (2.19)$$

En este trabajo tratamos de abordar la planificación de las rutas durante la estancia del turista donde la satisfacción o beneficio asociado a cada punto de

interés varía con la propia experiencia del usuario. Tal es el caso, por ejemplo, de las playas o los centros comerciales. Por tanto, en este modelo, los coeficientes de la función objetivo (las puntuaciones o beneficios) varían con el índice  $p$  correspondiente al día de estancia. Sin embargo, no bastaría con sustituir los parámetros  $b_i$  por parámetros  $b_i^p$  porque los valores correspondientes a un día dependen de la solución adoptada en los días anteriores. Por tanto, el modelo es aplicable a cada día de la estancia pero no a la estancia completa. Por este motivo, adoptamos la estrategia de resolver el problema de cada día para calcular los valores de los parámetros  $b_i$  para el día siguiente, aplicando el modelo OP para cada día de forma sucesiva.

### 2.3. Procedimiento de solución exacta

Como programa para obtener la solución exacta del modelo se ha aplicado Gusek<sup>1</sup>. Gusek es un programa gratuito que permite modelar de forma compacta o explícita los modelos de programación lineal. Resuelve los modelos usando el método simplex revisado (movimiento por los vértices).

En este programa se trabajará con tres ficheros que cuentan con las siguientes extensiones:

- *.dat*: Fichero de los datos de entrada.
- *.mod*: Fichero que contiene el código del modelo.
- *.out*: Fichero que contiene los resultados.

El fichero *.out* es el resultante de la ejecución del fichero *.mod* con los datos del fichero *.dat* contiene los resultados. Este texto puede ser analizado con fines de depuración, o para entender los comportamientos o la ejecución de un programa en particular. En estos ficheros se encuentra un seguimiento del valor que adquiere cada variable y también se puede observar si se van cumpliendo las restricciones empleadas en el modelo. En el mismo podemos identificar los siguientes elementos:

- *Problem*: Nombre del archivo que contiene el modelo.
- *Rows*: Cantidad de filas de la matriz.
- *Columns*: Cantidad de columnas. Existe una columna por cada variable.
- *Non-zeros*: Cantidad de elementos en la matriz distintos de 0.
- *Status*: Tipo de solución.
- *Objective*: Valor de la función objetivo.

---

<sup>1</sup> <http://www.gnu.org/software/glpk/>

## 2.4. Procedimiento heurístico

Dado que el problema de la optimización de rutas turísticas no se puede resolver usando el programa Gusek cuando la estancia es de varios días pero si cuando se trata de la ruta de un sólo día se propone una estrategia mixta. Se resolverá TTDP formulado como un TOP usando de manera consecutiva el modelo OP, por lo que se obtendrá una solución aproximada a la que hallaríamos al aplicar el modelo TOP. Además, no se implementará directamente el TOP porque se pretende que las rutas dependan de las anteriores.

La experimentación se llevará a cabo con la ejecución de tres módulos: dos programas implementados en C, uno para la lectura de datos y otro para la modificación de los parámetros del OP según las soluciones del día anterior; y el de Gusek, que contiene el modelo del TTDP. En términos generales, el procedimiento para abordar el problema de la optimización de las rutas turísticas de una estancia de varios días comprende los siguientes pasos:

1. Leer los datos asociados a los POIs para obtener los valores de la matriz de distancias o tiempos entre los POIs y los vectores de beneficio y duración.
2. Ejecutar el programa en C para leer los datos de los puntos y crear un fichero `.dat` con el formato de entrada a Gusek.
3. Ejecutar el programa Gusek para obtener la ruta del día correspondiente día.
4. Analizar el fichero de salida con las soluciones óptimas de la primera ruta y ejecutar un el otro programa en C para actualizar los beneficios ( $b_i$ ) de las categorías de los POIs visitados. Este programa tendrá como resultado otro fichero con formato `.dat` para la siguiente ejecución del Gusek.
5. Repetir los pasos 3 y 4 hasta llegar al último día de la estancia.

A continuación, se detalla la implementación de los pasos 2, 3 y 4 incluyendo sus respectivos códigos.

### 2.4.1. Lectura de datos de entrada

En esta subsección se expone el código que se ha realizado para la lectura de los datos expuestos en las tablas de la siguiente sección 3.1 y 3.2. Además, tiene la función de crear y escribir el primer fichero de entrada `.dat` con las cantidades leídas. Este programa solo será ejecutado una vez al principio para la primera iteración en Gusek.

Veamos las distintas partes que componen el código:

- Se declaran las variables y se pregunta el número  $n$  de POIs y el tiempo máximo  $t_{max}$  de que se dispone para cada una de las rutas.

```

/*****/
#include<stdio.h>
#include<stdlib.h>
#include <string.h>
#include <math.h>

FILE *datos, *entrada1;

int main(){
    int n,i,j;
    float tiempo;
    float x[100];
    float y[100];
    int b[100];
    float d[100];
    float dist[100][100];

    printf("Introduzca el numero de POIs a visitar:\n ");
    scanf("%d",&n);
    printf("Introduzca el tiempo que se dispone para la ruta
           :\n ");
    scanf("%f",&tiempo);
/*****/

```

- Se abre el fichero y se leen los tiempos de cada par de puntos ( $t_{ij}$ ), los beneficios ( $b_i$ ) de cada POI y la duración de su visita ( $d_i$ ).

```

/*****/
    datos = fopen("puntosGC.txt", "r");
    if(datos == NULL){
        printf("\nError al tratar de leer el archivo\n");
        return 1;
    }
    else{
        printf("\nEl archivo se creo correctamente\n");
    }
    for(i=1; i<=n; i++){
        for(j=1; j<=n; j++){
            fscanf(datos, "%f", &dist[i][j]);
        }
    }
    for(i=1; i<=n; i++){
        fscanf(datos, "%d %f", &b[i], &d[i]) ;
    }
    fclose(datos);

```



```

/*****/

```

- Los datos leídos se muestran por pantalla para comprobación.

```

/*****/
printf("\n \nLos valores de distancia leídos son: \n");
for(i=1; i<=n; i++){
    for(j=1; j<=n; j++){
        printf("%f ", dist[i][j]);
    }
    printf("\n");
}

printf("\n \nLos valores de beneficio y duracion de
visita leídos son: \n");
for(i=1; i<=n; i++){
    printf("%d %f \n", b[i], d[i]);
}
printf("\n");
/*****/

```

- Se crea el nuevo archivo con el formato de entrada para Gusek con parámetros leídos anteriormente.

```

/*****/
entrada1 = fopen("turistaHotel.dat", "w");
if(entrada1 == NULL){
    printf("\nError al tratar de leer el archivo");
}
else{
    printf("\nEl archivo se creo correctamente");
}
fprintf(entrada1, "param n:= %d;\nparam Tmax:=
%f;\nparam b:=\n", n, tiempo);
for(i=1; i<=n; i++)
    fprintf(entrada1, "%d %d\n", i, b[i]);
fprintf(entrada1, ";\n");
fprintf(entrada1, "param : A : t :=\n");
for(i=1; i<=n; i++)
    for(j=1; j<=n; j++)
        fprintf(entrada1, "%d %d %f\n", i, j, dist[i][j]);

fprintf(entrada1, ";\n");
fprintf(entrada1, "param d:=\n");

```

```

for(i=1; i<=n; i++)
    fprintf(entrada1, "%d %f\n", i,d[i]);
fprintf(entrada1, ";\nend;");
fclose(entrada1);

/*****

```

Figura 2.1: Primera entrada a Gusek (Ruta 7 horas).

```

param n:= 25;
param Tmax:= 7.000000;
param b:=
1 0
2 50
3 50
4 50
5 50
6 50
7 40
8 40
9 40
10 40
11 40
12 40
13 30
14 30
15 30
16 20
17 20
18 20
19 15
20 15
21 15
22 25
23 25
24 25
25 25
;

```

El programa crea un nuevo fichero *.dat* que será el fichero de entrada para la primera instancia. En este aparecerán los parámetros de entrada al programa tal y como se observa en la Figura 2.1. Cabe destacar que en esta Figura solo se muestra una parte de los datos, concretamente donde se encuentra el parámetro de beneficios que es el único que va a variar conforme pasen los días de la estancia.

#### 2.4.2. Modelo de optimización de rutas

En esta sección se encuentra el código Gusek del modelo *OP*, que se describe por partes.

- Se exponen los parámetros mencionados anteriormente y los conjuntos  $V$  (conjunto de POIs) y  $A$  (conjunto de arcos que van de cada POI a otro POI de  $V$ ).

```

/*****/
param n, integer, >= 3;
set V := 1..n;
set A, within V cross V;
param Tmax, integer, <=24;
param d{i in V};
param t{(i,j) in A};
param b{i in V};
/*****/

```

- Se exponen las variables binarias  $x_{ij}$ ,  $y_i$  y  $u_i$ .

```

/*****/
var x{(i,j) in A}, binary;
var y{i in V}, binary;
var u{i in V};
/*****/

```

- A continuación se añade la función objetivo. Se maximiza la suma del beneficio de los POIs visitados.

```

/*****/
maximize total: s{i in V} b[i]* y[i];
/*****/

```

- Se fija el POI 1, que corresponde con el hotel de donde partimos cada una de las rutas. Las siguientes restricciones nos permiten que el turista llegue y salga una sola vez del punto de interés  $i$ .

```

/*****/
s.t. origin: y[1]=1;
s.t. origin: y[1]=1;
s.t. leave{i in V}: sum{(i,j) in A} x[i,j] = y[i];
s.t. enter{i in V}: sum{(i,j) in A} x[j,i] = y[i];
/*****/

```

- Se añaden las restricciones para la eliminación de subciclos y para ajustarse al tiempo permitido, siendo este la suma del tiempo que se tarda en llegar de un POI a otro y la duración que supone visitarlo.

```

/*****/
s.t. isolatedNode{i in V, j in V: i==j}: x[i,j]=0;
s.t. subtours{i in V, j in V : i!= j and i!=1 and j!= 1}: u
    [j]>=u[i]+ x[i,j]-(n-2)*(1-x[i,j])+ (n-3)*x[j,i];
s.t. time: (sum{(i,j) in A} x[i,j]*t[i,j]) + (sum{i in V}
    d[i]*y[i]) <= Tmax;
/*****/

```

- Finalmente se ejecuta el resolutor del problema

```

/*****/
solve;
printf"El resultado es: \n";
display x;
end;
/*****/

```

### 2.4.3. Reducción de los beneficios.

Con respecto al paso 4 del procedimiento, en esta subsección se encuentra el código que lee los resultados del fichero *.out* y crea el nuevo fichero de entrada *.dat* con los beneficios actualizados según la experiencia de las rutas anteriores. Esta actualización se puede plantear de distintas formas que obedezcan a planteamientos realistas. En nuestra experiencia suponemos que los beneficios del turista al volver a visitar un punto de interés se reduciría si ya ha visitado otro punto similar. Esta reducción puede ser distinta dependiendo del tipo de recurso turístico del que se trate.

En la experiencia práctica partimos de la idea de que los POIs están clasificados en subconjuntos que llamamos “Categorías” correspondientes a POIs similares, en cuanto al interés que tienen para los turistas. De esta manera, estas categorías constituyen una partición del conjunto  $V$  de POIs, por lo que entre ellas tienen intersección vacía y su unión es igual a  $V$ . Además, se ha establecido que los POIs de una misma categoría poseen el mismo beneficio  $b_i$ . Por ejemplo: iglesias, castillos, ruinas antiguas, miradores, estadios, etc.

En el caso real que estudiamos en el capítulo siguiente los POIs están clasificados en las siguientes categorías: playas, monumentos naturales, parques temáticos, centros comerciales y museos.

En la tabla 2.1 se observan las preferencias que en este caso pondría el turista, dándole así las siguientes prioridades:

Playas > Monumentos naturales > Parques temáticos > Bodegas > Centro  
comercial > Museos

Tabla 2.1: Beneficios por categorías.

Categorías	Beneficio( $b_i$ )
Playas	50
Monumentos naturales	40
Parques temáticos	30
Bodegas	20
Centro comercial	15
Museos	25

Dado que la función objetivo maximiza los beneficios ( $b_i$ ), según este ejemplo queda claro que las playas posiblemente será la “Categoría” con mayor preferencia. En consecuencia, con este planteamiento lo que se pretende es la búsqueda de una ruta variada pero sin quitarle valor a las preferencias del turista. Para ello, implementaremos un programa que reduce el beneficio ( $b_i$ ) de los POIs de una categoría cuando es visitada. De esta forma, logramos que para la siguiente ruta se reduzca el  $b_i$  de la categoría visitada, favoreciendo así a las otras no visitadas, por lo que el turista disfrutará de una ruta de máximo interés pero menos monótona. En cuanto a la reducción, se ha fijado disminuir un 20% de su total. Por otra parte, es preciso aclarar que el  $b_i$  de los POIs visitados quedan reducidos a 0, por el hecho de que no queremos volver a visitarlos.

Este programa que se expone a continuación trabajará con tres ficheros: *turistaHotel.dat*, *turistaHotel.out* y *turistaHotel2.dat*. De los dos primeros extraeremos los datos necesarios para crear el tercer fichero con los datos listos para la siguiente iteración.

- En primer lugar, se abren los tres ficheros.

```

/*****

#include<stdio.h>
#include<stdlib.h>
#include <string.h>
#include <ctype.h>
#include<math.h>

FILE *fd, *fd1, *fd2;

int main(){

    int c;
    int i=1;

```

```

int b[100];
char linea[1024];
char direccion[] = "C:\\Users\\jorge\\Desktop\\TFG FINAL
\\turistaHotel.dat";
char direccion2[] = "C:\\Users\\jorge\\Desktop\\TFG FINAL
\\turistaHotel.out";
char direccion1[] = "C:\\Users\\jorge\\Desktop\\TFG FINAL
\\turistaHotel2.dat";

fd = fopen(direccion, "r");
if(fd == NULL){
    printf("Error al tratar de leer el archivo");
    return 1;
}
fd2 = fopen(direccion2, "rt");
if(fd == NULL){
    printf("Error al tratar de leer el archivo");
    return 1;
}

fd1 = fopen(direccion1, "w");
if(fd1 == NULL){
    printf("Error al tratar de leer el nuevo archivo");
}
else{
    printf("El archivo se creo correctamente");
}

```

```

/*****/

```

- A continuación se copian los mismos datos hasta el parámetro  $b_i$  que es el que se pretende modificar.

```

/*****/
while ((c = fgetc(fd)) != 'b' ){
    fputc (c, fd1);
}
/*****/

```

- Procedemos a la lectura de los datos del parámetro  $b_i$  y se convierten en números enteros para su operación.

```

/*****/
char v[270];
int beneficio[30];

```

```

printf ("\n");
for(i=1; i<=25; i++){
    while (fgetc(fd)!=' ');
    fgets(v,6,fd);
    beneficio[i] = atoi(v);
    printf ("El beneficio del punto %d es %d\n", i,
            beneficio[i]);
}
/*****/

```

- Se crea el nuevo archivo para la segunda instancia. Los beneficios de los POIs visitados adquieren el valor de 0 ya que no queremos volver a visitarlos, y el beneficio de los POIs no visitados es reducido en un 20% si se visitó un POI de su categoría, lo cual se consigue con el siguiente código.

```

/*****/
int contador = 1;
int benvisitados[30];
int solucion[30];
int visitados = 0;
int rebaja[30];

fprintf(fd1, "%s", "b:=\n");
while(fgets(linea, 1024, (FILE*) fd2)) {
    if (linea[7] == 'y'){
        int j=(int(linea[35])-48);
        solucion[contador]=j;
        visitados = visitados+j;
        if (solucion[contador] == 1){
            benvisitados[visitados] = beneficio[
                contador];
        }
        contador++;
    }
}

for(int i=1;i<=visitados;i++)
    printf("%d \n",benvisitados[i]);

for(i=1; i<=contador-1; i++){
    if (solucion[i] == 1)
        rebaja[i]=0;
    if ((solucion[i] == 0) & (beneficio[i]!=0)){
        rebaja[i] = beneficio[i];
        for (int p=1; p<=visitados; p++){

```

```

        if (beneficio[i] == benvisitados[p] ){
            if (i<7)
                rebaja[i] = round(beneficio[i]*0.5);
            else if (i<13)
                rebaja[i] = round(beneficio[i]*0.6);
            else if (i<16)
                rebaja[i] = round(beneficio[i]*0.9);
            else if (i<19)
                rebaja[i] = round(beneficio[i]*0.5);
            else if (i<22)
                rebaja[i] = round(beneficio[i]*0.4);
            else
                rebaja[i] = round(beneficio[i]*0.7);
        }
    }
}
fprintf(fd1, "%d %d\n", i, rebaja[i]);
}
while ((c = fgetc(fd))!= EOF )
    fputc (c,fd1);

fclose(fd);
fclose(fd1);
fclose(fd2);

return 0;
}
/*****

```

Con este programa hemos conseguido obtener la segunda entrada de datos. Este proceso junto con la ejecución del modelo en Gusek lo repetiremos hasta conseguir los días de planificación que correspondan a la estancia del turista.



## Experimentación

En este capítulo se describe la aplicación del modelo y procedimiento descrito anteriormente a un caso real en la isla de Gran Canaria. Se ha trabajado usando datos reales de tiempo que se han extraído de la web de *Google Maps* y de datos aproximados de los POIs escogidos.

### 3.1. Datos

En primer lugar, se han elegido 25 puntos de interés de la isla de Gran Canaria. Se ha cogido esta serie de POIs teniendo en cuenta que están repartidos geográficamente por toda la isla y que son atracciones turísticas frecuentemente visitadas. El lugar de alojamiento del turista será el Hotel Costa Meloneras, y se ha propuesto hacer las rutas para una estancia de 4 días. Se ha hecho el cálculo de dos itinerarios, uno con un tiempo máximo de 7 horas y otro de 8.

En la tabla 3.1 se muestran los datos que se han establecido para los puntos seleccionados. Consta de los siguientes parámetros:

- POIs que se pueden visitar.
- Categoría de cada POI.
- Beneficio de cada POI.
- Duración de la visita de cada POI.

Con respecto a estos dos últimos parámetros se han establecido explícitamente de una manera razonada. En el parámetro de beneficio ( $b_i$ ) se ha querido da mayor relevancia a las playas y a los monumentos naturales ya que son los más valorados en el turismo canario, y en el parámetro de duración ( $d_i$ ) se ha establecido un tiempo acorde con las horas que conlleva realizar la visita de ese punto.

Tabla 3.1: Datos: POIs Gran Canaria.

Nº POI	POIs	Categorías	Beneficio	Duración
1	Costa Meloneras	Hotel	0	0.0
2	Canteras	Playa	50	3.2
3	Dunas de Maspalomas	Playa	50	3.2
4	Puerto de Mogán	Playa	50	3.0
5	Playa de Tasarte	Playa	50	3.0
6	Agaete	Playa	50	3.0
7	Parque Tamadaba	Monumento natural	40	2.0
8	Roque Nublo	Monumento natural	40	2.3
9	Parque natural de Pílancones	Monumento natural	40	2.3
10	Yacimiento Arqueológico Risco Caído	Monumento natural	40	2.0
11	Casa cueva canaria	Monumento natural	40	2.0
12	Barranco de las vacas	Monumento natural	40	2.0
13	Aqualand	Parque temático	30	4.0
14	Palmitos Park	Parque temático	30	3.5
15	Hangar 37	Parque temático	30	3.0
16	Bodega los Berrazales	Bodega	20	1.4
17	Bodegas Mondalón	Bodega	20	1.4
18	Bodegas Bentayga	Bodega	20	1.4
19	Mogan Mall	Centro comercial	15	2.0
20	CC. Las terrazas	Centro comercial	15	2.0
21	CC. Alisios	Centro comercial	15	2.0
22	Museo Canario	Museo	25	2.0
23	Cueva Pintada de Gáldar	Museo	25	2.0
24	Mundo Aborigen	Museo	25	2.0
25	Poema del mar	Museo	25	2.0

En la Imagen 3.1 se muestra la ubicación en el mapa de la isla de Gran Canaria los POIs descritos en la tabla anterior 3.1. Cada color representa una categoría diferente, la etiqueta de color negro es el lugar donde se encuentra el hotel.

Además se ha contado con información que se ha extraído de *Google Maps* en cuanto al tiempo medio que se necesita para moverse de un POI a otro. Estos datos se recogen en la tabla 3.2.

Tabla 3.2: Tiempo (horas) en recorrer de un POI a otro.

param	E	t	:=	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
1	0.000	0.817	0.167	0.350	1.100	1.183	1.317	1.150	1.000	1.717	0.767	0.500	0.133	0.317	0.233	1.350	0.683	1.483	0.250	0.533	0.833	0.700	1.000	0.300	0.800	0.300	0.800	
2	0.817	0.000	0.733	1.000	1.666	0.533	1.000	1.000	1.617	1.067	0.783	0.666	0.733	0.917	0.667	0.683	0.316	1.050	1.033	0.300	1.167	0.200	0.450	0.783	0.683	0.050	0.783	
3	0.167	0.733	0.000	0.417	1.167	1.200	1.400	1.117	1.067	0.700	0.767	0.466	0.167	0.350	0.216	1.317	0.683	1.467	0.283	0.533	0.733	0.683	1.067	0.267	0.783	0.067	0.783	
4	0.350	1.000	0.417	0.000	0.917	1.500	1.150	1.150	1.267	1.950	1.000	0.800	0.400	0.567	0.533	1.667	0.933	1.300	0.300	0.810	0.950	0.917	1.417	0.533	1.066	0.917	0.533	
5	1.100	1.666	1.167	0.917	0.000	1.217	0.850	1.467	2.000	1.800	1.733	1.400	1.133	1.300	1.183	1.367	1.650	1.816	0.933	1.517	1.667	1.667	1.367	1.267	1.700	1.367	1.267	
6	1.183	0.533	1.200	1.500	1.217	0.000	0.500	1.333	2.000	0.717	1.150	1.050	1.100	1.283	1.083	0.217	0.667	1.283	1.383	0.667	0.533	0.567	0.183	1.167	0.633	0.567	0.183	
7	1.317	1.000	1.400	1.150	0.850	0.500	0.000	1.500	2.100	1.183	1.617	1.566	1.350	1.500	1.400	0.650	1.133	1.516	1.133	1.133	1.000	1.033	0.650	1.500	1.066	1.500	1.066	
8	1.150	1.000	1.117	1.150	1.467	1.333	1.500	0.000	0.750	0.900	0.950	1.000	1.100	1.283	1.116	1.467	0.867	0.516	1.150	0.933	0.933	1.000	1.283	0.867	1.116	1.283	0.867	
9	1.000	1.617	1.067	1.267	2.000	2.000	2.100	0.750	0.000	1.600	1.717	1.283	1.000	1.200	1.000	2.250	1.617	1.183	1.083	1.460	1.600	1.600	2.000	1.183	1.583	1.600	2.000	
10	1.717	1.067	1.700	1.950	1.800	0.717	1.183	0.900	1.600	0.000	1.617	1.600	1.600	1.178	1.600	0.900	1.167	0.667	1.900	1.167	1.000	1.067	0.717	1.700	1.116	1.067	0.717	
11	0.767	0.783	0.767	1.000	1.733	1.150	1.617	0.950	1.717	1.617	0.000	0.366	0.717	0.883	0.616	1.333	0.700	1.167	0.867	0.567	0.717	0.700	1.083	0.767	0.883	0.767	0.883	
12	0.500	0.666	0.466	0.800	1.400	0.500	1.566	1.000	1.283	1.600	0.366	0.000	0.450	0.633	0.333	1.250	0.567	1.150	0.583	0.467	0.583	0.583	1.000	0.516	0.683	0.516	0.683	
13	0.133	0.733	0.167	0.400	1.133	1.100	1.350	1.100	1.000	1.600	0.717	0.450	0.000	0.183	0.200	1.317	0.683	1.450	0.283	0.550	0.700	0.683	1.050	0.250	0.766	0.683	0.250	
14	0.317	0.917	0.350	0.567	1.300	1.283	1.500	1.283	1.200	1.178	0.883	0.633	0.183	0.000	0.366	1.500	0.867	1.633	0.450	0.717	0.867	0.850	1.233	0.417	0.950	0.850	1.233	
15	0.233	0.667	0.216	0.533	1.183	1.083	1.400	1.116	1.000	1.600	0.616	0.333	0.200	0.366	0.000	1.267	0.583	1.300	0.333	0.467	0.583	0.600	0.966	0.250	0.633	0.600	0.966	
16	1.350	0.683	1.317	1.667	1.367	0.217	0.650	1.467	2.250	0.900	1.333	1.250	1.317	1.500	1.267	0.000	0.833	1.416	1.533	0.833	0.667	0.733	0.350	1.317	0.766	0.733	0.350	
17	0.683	0.316	0.683	0.933	1.650	0.667	1.133	0.867	1.133	1.67	0.700	0.567	0.683	0.867	0.583	0.833	0.000	0.950	0.833	0.250	0.250	0.300	0.617	0.733	0.366	0.617	0.733	
18	1.483	1.050	1.467	1.300	1.816	1.283	1.516	0.516	1.183	0.667	1.167	1.150	1.450	1.633	1.300	1.416	0.950	0.000	1.533	1.016	0.900	1.066	1.233	1.333	1.100	1.066	1.233	
19	0.250	1.033	0.283	0.300	0.933	1.383	1.133	1.150	1.083	1.900	0.867	0.583	0.283	0.45	0.333	1.533	0.833	1.533	0.000	0.716	0.900	0.867	1.283	0.416	0.933	0.867	1.283	
20	0.533	0.300	0.533	0.810	1.517	0.667	1.133	0.933	1.460	1.167	0.567	0.467	0.550	0.717	0.467	0.833	0.250	1.016	0.716	0.000	0.233	0.217	0.600	0.617	0.300	0.617	0.300	
21	0.833	1.167	0.733	0.950	1.667	0.533	1.000	0.933	1.600	1.000	0.717	0.583	0.700	0.867	0.583	0.667	0.250	0.900	0.900	0.233	0.000	0.233	0.500	0.733	0.233	0.500	0.733	
22	0.700	0.200	0.683	0.917	1.667	0.567	1.033	1.000	1.600	1.067	0.700	0.583	0.683	0.850	0.600	0.733	0.300	1.066	0.867	0.217	0.233	0.000	0.567	0.733	0.150	0.567	0.733	
23	1.000	0.450	1.067	1.417	1.367	0.183	0.650	1.283	2.000	0.717	1.083	1.000	1.050	1.233	0.966	0.350	0.617	1.233	1.283	0.600	0.500	0.567	0.000	1.067	0.516	0.567	0.000	
24	0.300	0.783	0.267	0.533	1.267	1.167	1.500	0.867	1.183	1.700	0.767	0.516	0.250	0.417	0.25	1.317	0.733	1.333	0.416	0.617	0.733	0.733	1.067	0.000	0.816	0.733	1.067	
25	0.800	0.050	0.783	1.066	1.700	0.633	1.066	1.116	1.583	1.116	0.833	0.683	0.766	0.950	0.633	0.766	0.366	1.100	0.933	0.300	0.233	0.150	0.516	0.816	0.000	0.516	0.816	



se puede observar si se van cumpliendo las restricciones empleadas en el modelo.

En la Figura 3.2 se observa los fragmentos de los ficheros de salida en los que se muestra la variable binaria  $y_i$ , que adquiere el valor 1 cuando el POI es visitado y 0 cuando no.

Figura 3.2: POIs visitados en el día 1.

(a) 7 horas			(b) 8 horas		
y[1]	*	1	y[1]	*	1
y[2]	*	0	y[2]	*	0
y[3]	*	1	y[3]	*	0
y[4]	*	0	y[4]	*	0
y[5]	*	0	y[5]	*	0
y[6]	*	0	y[6]	*	0
y[7]	*	0	y[7]	*	0
y[8]	*	0	y[8]	*	0
y[9]	*	0	y[9]	*	0
y[10]	*	0	y[10]	*	0
y[11]	*	1	y[11]	*	1
y[12]	*	0	y[12]	*	1
y[13]	*	0	y[13]	*	0
y[14]	*	0	y[14]	*	0
y[15]	*	0	y[15]	*	0
y[16]	*	0	y[16]	*	0
y[17]	*	0	y[17]	*	0
y[18]	*	0	y[18]	*	0
y[19]	*	0	y[19]	*	0
y[20]	*	0	y[20]	*	0
y[21]	*	0	y[21]	*	0
y[22]	*	0	y[22]	*	0
y[23]	*	0	y[23]	*	0
y[24]	*	0	y[24]	*	1
y[25]	*	0	y[25]	*	0

De la misma manera en la Figura 3.3 se puede ver la variable  $x_{ij}$  que es la que nos va indicar cómo será la ruta entre los POIs, que adquirieron valor 1 en la variable  $y_i$ .

Figura 3.3: Ruta del día 1

(a) 7 horas			(b) 8 horas		
x[1,11]	*	1	x[1,24]	*	1
x[3,1]	*	1	x[11,12]	*	1
x[11,3]	*	1	x[12,1]	*	1
			x[24,11]	*	1

La ruta geográficamente sería la mostrada en la Figura 3.4:

Tabla 3.3: Ruta (7 horas) día 1

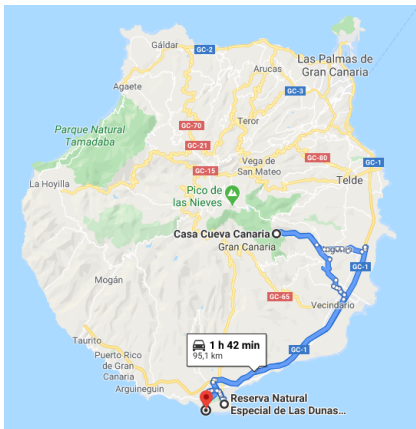
Ruta (7 horas) Día 1			
1	Costa Meloneras	Hotel	tiempo = 6.901h
11	Casa cueva canaria	Monumento natural	Función objetivo = 90
3	Dunas de Maspalomas	Playa	Memory used: 2.6 Mb (2685978 bytes)
1	Costa Meloneras	Hotel	Time used: 0.2 secs

Tabla 3.4: Ruta (8 horas) día 1

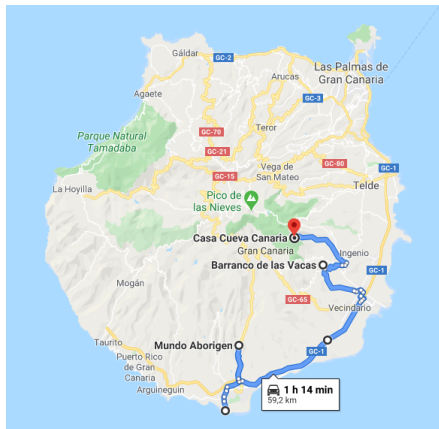
Ruta (8 horas) Día 1			
1	Costa Meloneras	Hotel	tiempo = 7.933h
24	Mundo Aborigen	Museo	Función objetivo = 105
11	Casa cueva Canaria	Monumento natural	
12	Barranco de las vacas	Monumento natural	Memory used: 3.2 Mb (3304771 bytes)
1	Costa Meloneras	Hotel	Time used: 0.9 secs

Figura 3.4: Mapa de las rutas del día 1

(a) 7 horas



(b) 8 horas



## Día 2

Iniciaremos un nuevo día con mismos POIs y sus correspondientes parámetros  $t_{ij}$  y  $d_i$ . Se ejecutará el modelo otra vez con la pequeña alteración de los  $b_i$ .

En la siguiente Figura 3.5a podemos ver cómo el beneficio de los POIs 3 y 11 ahora tienen valor 0 (aquellos POIs que visitamos el día 1). Asimismo, según la primera solución mostrada 3.4, se observa que se han visitado las categorías de

playa y monumento natural. En efecto, se aprecia que el  $b_i$  de las playas adquiere ahora el valor de 40 y el de los monumentos naturales de 32 cuando antes eran de 50 y 40 respectivamente. Para los  $b_i$  de la ruta de 8 horas ocurre lo mismo, en su caso, los POIs visitados han sido el 11, 12 y 24.

Figura 3.5: Parámetros de beneficios del día 2

(a) 7 horas	(b) 8 horas
param b:=	param b:=
1 0	1 0
2 40	2 50
3 0	3 50
4 40	4 50
5 40	5 50
6 40	6 50
7 32	7 32
8 32	8 32
9 32	9 32
10 32	10 32
11 0	11 0
12 32	12 0
13 30	13 30
14 30	14 30
15 30	15 30
16 20	16 20
17 20	17 20
18 20	18 20
19 15	19 15
20 15	20 15
21 15	21 15
22 25	22 20
23 25	23 20
24 25	24 0
25 25	25 20

De nuevo, en las Figuras 3.6 y 3.7 se hallarán los resultados.

Figura 3.6: POIs visitados en el día 2

(a) 7 horas			(b) 8 horas		
y[1]	*	1	y[1]	*	1
y[2]	*	0	y[2]	*	0
y[3]	*	0	y[3]	*	1
y[4]	*	1	y[4]	*	1
y[5]	*	0	y[5]	*	0
y[6]	*	0	y[6]	*	0
y[7]	*	0	y[7]	*	0
y[8]	*	0	y[8]	*	0
y[9]	*	0	y[9]	*	0
y[10]	*	0	y[10]	*	0
y[11]	*	0	y[11]	*	0
y[12]	*	1	y[12]	*	0
y[13]	*	0	y[13]	*	0
y[14]	*	0	y[14]	*	0
y[15]	*	0	y[15]	*	0
y[16]	*	0	y[16]	*	0
y[17]	*	0	y[17]	*	0
y[18]	*	0	y[18]	*	0
y[19]	*	0	y[19]	*	0
y[20]	*	0	y[20]	*	0
y[21]	*	0	y[21]	*	0
y[22]	*	0	y[22]	*	0
y[23]	*	0	y[23]	*	0
y[24]	*	0	y[24]	*	0
y[25]	*	0	y[25]	*	0

Figura 3.7: Rutas del día 2

(a) 7 horas			(b) 8 horas		
x[1,12]	*	1	x[1,4]	*	1
x[4,1]	*	1	x[3,1]	*	1
x[12,4]	*	1	x[4,3]	*	1

Por tanto, las soluciones para este segundo día son las mostradas en las Tablas 3.5 y 3.6:

Tabla 3.5: Ruta (7 horas) día 2.

Ruta (7 horas) Día 2			
1	Costa Meloneras	Hotel	tiempo = 6.65h
12	Barranco de las vacas	Monumento natural	Función objetivo = 72
4	Puerto de Mogán	Playa	Memory used: 2.8 Mb (2896229 bytes)
1	Costa Meloneras	Hotel	Time used: 0.3 secs

En la Figura 3.8 se vuelve a describir las rutas geográficamente.

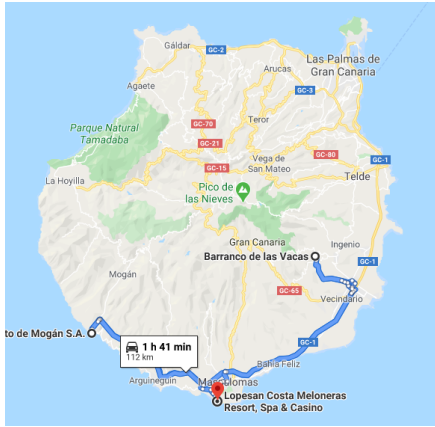


Tabla 3.6: Ruta (8 horas) día 2.

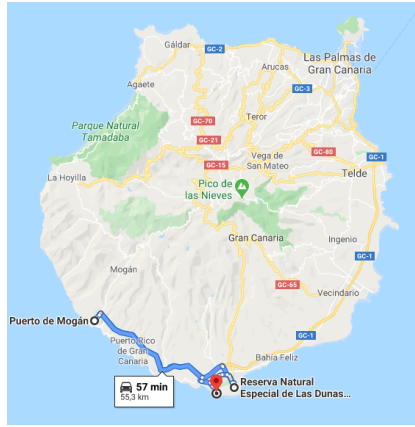
Ruta (8 horas) Día 2			
1	Costa Meloneras	Hotel	tiempo = 7.134h
4	Puerto de Mogán	Playa	Función objetivo = 100
3	Dunas de Maspalomas	Playa	Memory used: 2.7 Mb (2882218 bytes)
1	Costa Meloneras	Hotel	Time used: 0.3 secs

Figura 3.8: Mapa de las rutas del día 2

(a) 7 horas



(b) 8 horas



### Día 3

En el tercer día, se seguirán los mismos pasos que fueron aplicados en el segundo día.

Nuevamente los  $b_i$  de los POIs visitados en el día 2 pasan tener valor 0 junto a los del día 1 3.9.

Figura 3.9: Parámetro de beneficio día 3

(a) 7 horas	(b) 8 horas
param b:=	param b:=
1 0	1 0
2 32	2 40
3 0	3 0
4 0	4 0
5 32	5 40
6 32	6 40
7 26	7 32
8 26	8 32
9 26	9 32
10 26	10 32
11 0	11 0
12 0	12 0
13 30	13 30
14 30	14 30
15 30	15 30
16 20	16 20
17 20	17 20
18 20	18 20
19 15	19 15
20 15	20 15
21 15	21 15
22 25	22 20
23 25	23 20
24 25	24 0
25 25	25 20

Reiteradamente:

Figura 3.10: POIs visitados en el día 3

(a) 7 horas			(b) 8 horas		
y[1]	*	1	y[1]	*	1
y[2]	*	1	y[2]	*	0
y[3]	*	0	y[3]	*	0
y[4]	*	0	y[4]	*	0
y[5]	*	0	y[5]	*	0
y[6]	*	0	y[6]	*	1
y[7]	*	0	y[7]	*	1
y[8]	*	0	y[8]	*	0
y[9]	*	0	y[9]	*	0
y[10]	*	0	y[10]	*	0
y[11]	*	0	y[11]	*	0
y[12]	*	0	y[12]	*	0
y[13]	*	0	y[13]	*	0
y[14]	*	0	y[14]	*	0
y[15]	*	0	y[15]	*	0
y[16]	*	0	y[16]	*	0
y[17]	*	0	y[17]	*	0
y[18]	*	0	y[18]	*	0
y[19]	*	0	y[19]	*	0
y[20]	*	0	y[20]	*	0
y[21]	*	0	y[21]	*	0
y[22]	*	1	y[22]	*	0
y[23]	*	0	y[23]	*	0
y[24]	*	0	y[24]	*	0
y[25]	*	0	y[25]	*	0

Figura 3.11: Rutas del día 3

(a) 7 horas			(b) 8 horas		
x[1,2]	*	1	x[1,6]	*	1
x[2,22]	*	1	x[6,7]	*	1
x[22,1]	*	1	x[7,1]	*	1

Tabla 3.7: Ruta (7 horas) día 3.

Ruta (7 horas) Día 3			
1	Costa Meloneras	Hotel	tiempo = 6.917h
2	Canteras	Playa	Función objetivo = 57
22	Museo canario Las Palmas	Monumento natural	Memory used: 3.6 Mb (3769537 bytes)
1	Costa Meloneras	Hotel	Time used: 2.5 secs

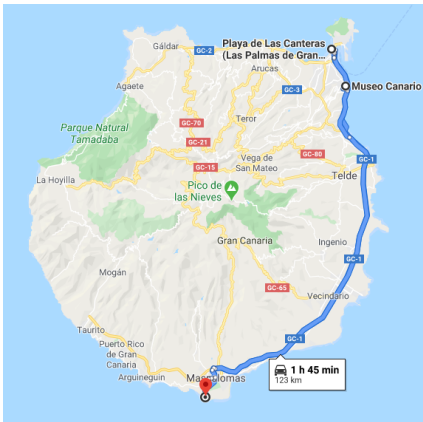
En la Figura 3.12 se observa la ruta geográfica.

Tabla 3.8: Ruta (8 horas) día 3

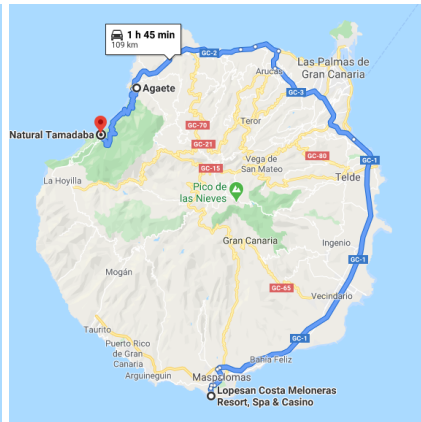
Ruta (8 horas) Día 3			
1	Costa Meloneras	Hotel	tiempo = 8h
6	Agaeete	Playa	Función objetivo = 72
7	Parque Tamadaba	Monumento natural	Memory used: 2.5 Mb (2573046 bytes)
1	Costa Meloneras	Hotel	Time used: 0.5 secs

Figura 3.12: Mapa de las rutas del día 3

(a) 7 horas



(b) 8 horas



### Día 4

Para terminar, los últimos resultados.

En la Figura 3.13 se puede examinar la reducción de los beneficios.

En las siguientes Figuras 3.14, 3.15 observamos la salida de la nueva iteración.

Figura 3.13: Parámetros de beneficios del día 4.

(a) 7 horas	(b) 8 horas
param b:=	param b:=
1 0	1 0
2 0	2 32
3 0	3 0
4 0	4 0
5 26	5 32
6 26	6 0
7 26	7 0
8 26	8 26
9 26	9 26
10 26	10 26
11 0	11 0
12 0	12 0
13 30	13 30
14 30	14 30
15 30	15 30
16 20	16 20
17 20	17 20
18 20	18 20
19 15	19 15
20 15	20 15
21 15	21 15
22 0	22 20
23 20	23 20
24 20	24 0
25 20	25 20

Figura 3.14: POIs visitados en el día 4

(a) 7 horas	(b) 8 horas
y[1] *	y[1] *
y[2] *	y[2] *
y[3] *	y[3] *
y[4] *	y[4] *
y[5] *	y[5] *
y[6] *	y[6] *
y[7] *	y[7] *
y[8] *	y[8] *
y[9] *	y[9] *
y[10] *	y[10] *
y[11] *	y[11] *
y[12] *	y[12] *
y[13] *	y[13] *
y[14] *	y[14] *
y[15] *	y[15] *
y[16] *	y[16] *
y[17] *	y[17] *
y[18] *	y[18] *
y[19] *	y[19] *
y[20] *	y[20] *
y[21] *	y[21] *
y[22] *	y[22] *
y[23] *	y[23] *
y[24] *	y[24] *
y[25] *	y[25] *

Figura 3.15: Rutas del día 4

(a) 7 horas			(b) 8 horas		
x[1,15]	*	1	x[1,15]	*	1
x[15,24]	*	1	x[15,17]	*	1
x[24,1]	*	1	x[17,20]	*	1
			x[20,1]	*	1

El resultado generado ha sido:

Tabla 3.9: Ruta (7 horas) día 4

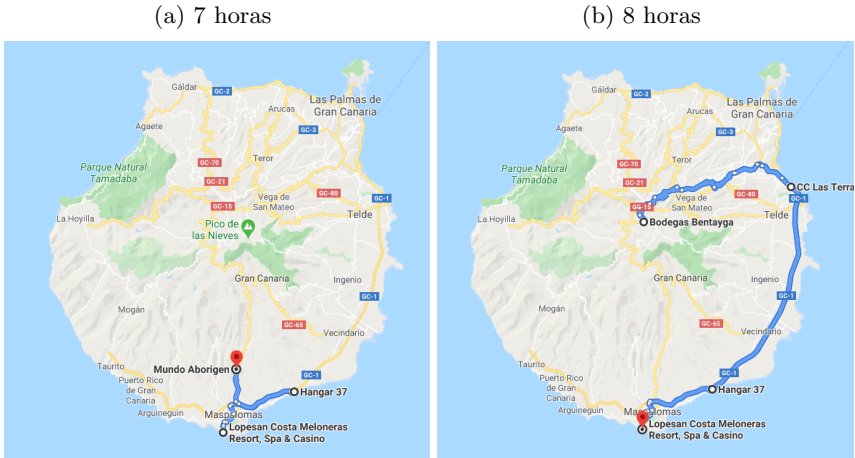
Ruta (7 horas) Día 4			
1	Costa Meloneras	Hotel	tiempo = 5.783h
15	Hangar 37	Parque temático	Función objetivo = 50
24	Mundo aborigen	Museo	Memory used: 3.2 Mb (3338796 bytes)
1	Costa Meloneras	Hotel	Time used: 2.1 secs

Tabla 3.10: Ruta (8 horas) día 4

Ruta (8 horas) Día 4			
1	Costa Meloneras	Hotel	tiempo = 7.999h
15	Hangar 37	Parque temático	Función objetivo = 65
17	Bodegas Bentayga	Bodega	
20	CC. Las Terrazas	Centro comercial	Memory used: 3.1 Mb (3298031 bytes)
1	Costa Meloneras	Hotel	Time used: 1.3 secs

Por último, se muestra en la Figura 3.18 la ruta geográficamente del día 4.

Figura 3.16: Mapa de las rutas del día 4



Además se ha realizado otra experiencia diseñando las rutas con la reducción de los beneficios de las categorías de forma diferenciada. En la experiencia anterior se disminuían los beneficios de todas las categorías en un 20 %, pero ahora se efectuarán de la siguiente manera:

- Cuando se visiten las “Playas” un 50 %.
- Cuando se visiten los “Monumentos naturales” un 60 %.
- Cuando se visiten los “Parques temáticos” un 90 %.
- Cuando se visiten los “Bodega” un 50 %.
- Cuando se visiten los “Centros comerciales” 40 %.
- Cuando se visiten los “Museos” un 70 %.

Se ha realizado la experiencia obteniendo las rutas de los 4 días sólo con una duración máxima de 8 horas. Los resultados obtenidos muestran que las rutas son muy similares al caso anterior con tiempos de cómputos muy similares y las rutas sólo se diferencian en la del día 4.

Figura 3.17: Solución del día 4

```

y[1]      *      1
y[2]      *      0
y[3]      *      0
y[4]      *      0
y[5]      *      0
y[6]      *      0
y[7]      *      0
y[8]      *      1
y[9]      *      1
y[10]     *      0
y[11]     *      0
y[12]     *      0
y[13]     *      0
y[14]     *      0
y[15]     *      0
y[16]     *      0
y[17]     *      0
y[18]     *      0
y[19]     *      0
y[20]     *      0
y[21]     *      0
y[22]     *      0
y[23]     *      0
y[24]     *      0
y[25]     *      0
    
```

Tabla 3.11: Ruta (8 horas) día 4

Ruta (8 horas) Día 4			
1	Costa Meloneras	Hotel	tiempo = 7.5h
8	Roque Nublo	Monumento natural	Función objetivo = 56
9	Parque natural de Pílancones	Monumento natural	Memory used: 2.4 Mb (2473531 bytes)
1	Costa Meloneras	Hotel	Time used: 0.1 secs







---

## Conclusiones

En este trabajo se ha conseguido mostrar cómo las técnicas de programación matemática se pueden combinar con otras técnicas algorítmicas para desarrollar procedimientos que pueden dar respuestas a problemas de planificación de rutas turísticas. Se ha elaborado un modelo de optimización para que el turista pueda mejorar su toma de decisiones a la hora de organizar sus vacaciones. Dicho modelo se convierte en un instrumento muy útil para el sector turístico, ya que permite generar una diversidad de rutas de manera personalizada según el tiempo disponible y la importancia que se le asigne a cada POI, o categoría de POIs. Con ello, se consigue un trato individualizado hacia el turista permitiendo que este pueda quedar más satisfecho y tenga una mejor experiencia de viaje, siendo además una propuesta de ruta diferente a las que hay prefijadas con un gusto generalizado. Bien es cierto, que los expertos del turismo se ayudan de estudios de mercado y de encuestas de satisfacción, por lo que complementar el modelo con todo ello puede ser un tanto laborioso pero podría enriquecer los estudios desarrollados por los especialistas.

Por otro lado, algunas de las propuestas que se podría extender a este trabajo consisten en añadir al modelo y a los programas algunas ventanas de tiempo para los puntos de interés. También se propone incluir información meteorológica para descartar ciertos puntos con el tiempo meteorológico desfavorable. Otras serían establecer un límite de presupuesto para el viaje, que cada jornada la ruta tuviese que pasar a mediodía por un restaurante, etc. De esta manera, se conseguiría perfeccionar más el modelo, de modo que cuanto más información poseamos más se puede alcanzar una solución más ajustada a la realidad.



---

## Lista de Figuras

2.1. Primera entrada a Gusek (Ruta 7 horas).....	14
3.1. Mapa de POIs Gran Canaria. ....	24
3.2. POIs visitados en el día 1. ....	25
3.3. Ruta del día 1 . ....	25
3.4. Mapa de las rutas del día 1 . ....	26
3.5. Parámetros de beneficios del día 2 . ....	27
3.6. POIs visitados en el día 2 . ....	28
3.7. Rutas del día 2.....	28
3.8. Mapa de las rutas del día 2 . ....	29
3.9. Parámetro de beneficio día 3 . ....	30
3.10. POIs visitados en el día 3 . ....	31
3.11. Rutas del día 3.....	31
3.12. Mapa de las rutas del día 3 . ....	32
3.13. Parámetros de beneficios del día 4. ....	33
3.14. POIs visitados en el día 4 . ....	33
3.15. Rutas del día 4.....	34
3.16. Mapa de las rutas del día 4 . ....	35
3.17. Solución del día 4 . ....	36
3.18. Mapa de la ruta del día 4.....	37



---

## Lista de Tablas

2.1. Beneficios por categorías. ....	17
3.1. Datos: POIs Gran Canaria. ....	22
3.2. Tiempo (horas) en recorrer de un POI a otro. ....	23
3.3. Ruta (7 horas) día 1. ....	26
3.4. Ruta (8 horas) día 1. ....	26
3.5. Ruta (7 horas) día 2. ....	28
3.6. Ruta (8 horas) día 2. ....	29
3.7. Ruta (7 horas) día 3. ....	31
3.8. Ruta (8 horas) día 3. ....	32
3.9. Ruta (7 horas) día 4. ....	34
3.10. Ruta (8 horas) día 4. ....	34
3.11. Ruta (8 horas) día 4. ....	36





---

## Bibliografia

- [1] GAVALAS, D., KONSTANTOPOULOS, C., MASTAKAS, K., PANTZIOU, G. *A survey on algorithmic approaches for solving tourist trip design problems* Journal of Heuristics, 20(3) (2014), Pages 291-328.
- [2] LAPORTE, G., MARTELLO, S. *The selective travelling salesman problem* Discrete Applied Mathematics Volume 26, Issues 2–3, March 1990, Pages 193-207.
- [3] MAKHORIN, A. *GNU Linear Programming Kit* Manual. Version 4.45.2010.
- [4] MILLER, C., TUCKER, A., ZEMLIN, R. *Integer programming formulations and travelling salesman problems*. Journal of the AMC 7, Pages 326-329.
- [5] RAMESH, R., YOON, Y., KARWAN, M. *An optimal algorithm for the orienteering tour problem*. ORSA Journal on Computing, 4 (1992), Pages 155-165.
- [6] SCHILDE, M., DOERNER, K.F., HARTL, R.F., KIECHLE, G. *Metaheuristics for the bi-objective orienteering problem* Volume 3, Issue 3, September 2009, Pages 179–201 Volume 209, Issue 1, 16 February 2011, Pages 1-10.
- [7] TSILIGIRIDES, T. *Heuristic methods applied to orienteering*. Journal of the Operational Research Society, 35 (9) (1984), Pages 797-809.
- [8] VANSTEENWEGEN, P., SOUFFRIAU, W., VAN OUDHEUSDEN, D. *The orienteering problem: A survey*. European Journal of Operational Research Volume 209, Issue 1, 16 February 2011, Pages 1-10.





# Optimitation of tourist routes

Jorge Brito Dávila

Facultad de Ciencias · Sección de Matemáticas

Universidad de La Laguna

alu0100890495@ull.edu.es

## Abstract

It is a priority for companies in the tourism sector to meet the specific needs of tourists by programming personalized activities with the definition of tourist routes for budgeting and adequate administration of resources and time, seeking to meet their demands for quality and flexibility. Through the use of mathematical tool combined with some algorithm techniques we can offer solutions applied to the service economy and specifically to the planning of these tourist routes. We will put into operation a mathematical model with the design of an algorithm after defining variables that describe the interests of the tourist, showing the results and conclusions of the model.

## 1. Introduction

The problem of the choice of activities by the tourist traveler has become an interesting topic of research in recent years. This information is used to plan the trip in such a way that it has the highest possible performance. That is, tourists can visit the most interesting places taking advantage of all the time they have to make a route, organizing their own itineraries and responding to your own interests on time. In this project we will propose a model that finds the best route every day according to the tourist's preferences.

## 2. Optimization of tourist routes

Most of the tourist route optimization problems or *Tourist Trip Design Problems (TTDP)* are derived from the design and planning of tourist routes stem from the basic problem known as *The Orienteering Problem (OP)*. In general, this problem is based on a set of control points that have a certain score, and what is involved is to go through the control points in such a way that when returning to the initial point the score is maximized. All this assuming you have a time limit that prevents you from going through all the points. So, lets take  $n$  number of collection points and a set of points  $V = \{1, \dots, n\}$ . Let the available time be  $T_{max}$ , the score of each point be  $b_i$ , the duration of the visit to each point be  $d_i$ , the time to need to go from point  $i$  to  $j$  be  $t_{ij}$ . By last, the following variables:  $x_{ij}$ , indicates if the route goes from point  $i$  to  $j$ ;  $y_i$ , indicates if the point  $i$  has been visited; and  $u_i$ , indicates the position of the point  $i$  in the route. After thah, the objective is to maximize the following function subject to the following restrictions.

Objective function:

$$\max \sum_{i=1}^n \sum_{j=1}^n b_i y_i \quad (1)$$

subject to:

$$\sum_{j=2}^n x_{1j} = \sum_{i=2}^n x_{i1} = 1, \quad (2)$$

$$\sum_{i=1}^n x_{ik} = \sum_{j=1}^n x_{kj} \leq 1; \forall k = 2, \dots, n, \quad (3)$$

$$\sum_{i=1}^n \sum_{j=1}^n t_{ij} x_{ij} + \sum_{i=1}^n d_i y_i \leq T_{max}, \quad (4)$$

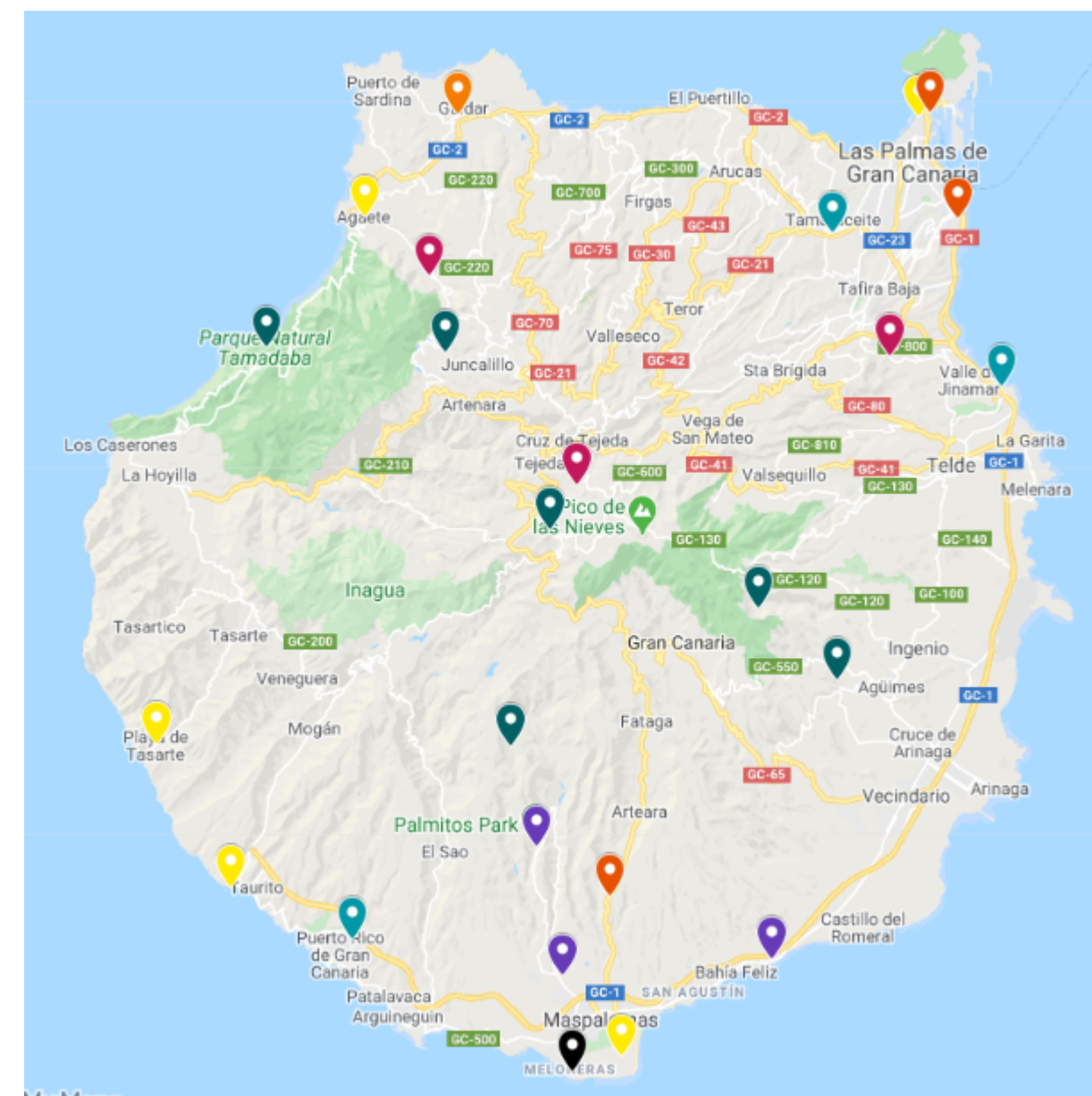
$$2 \leq u_i \leq N; \forall i = 2, \dots, n, \quad (5)$$

$$u_i - u_j + 1 \leq (n-1)(1 - x_{ij}); \forall i, j = 2, \dots, n, \quad (6)$$

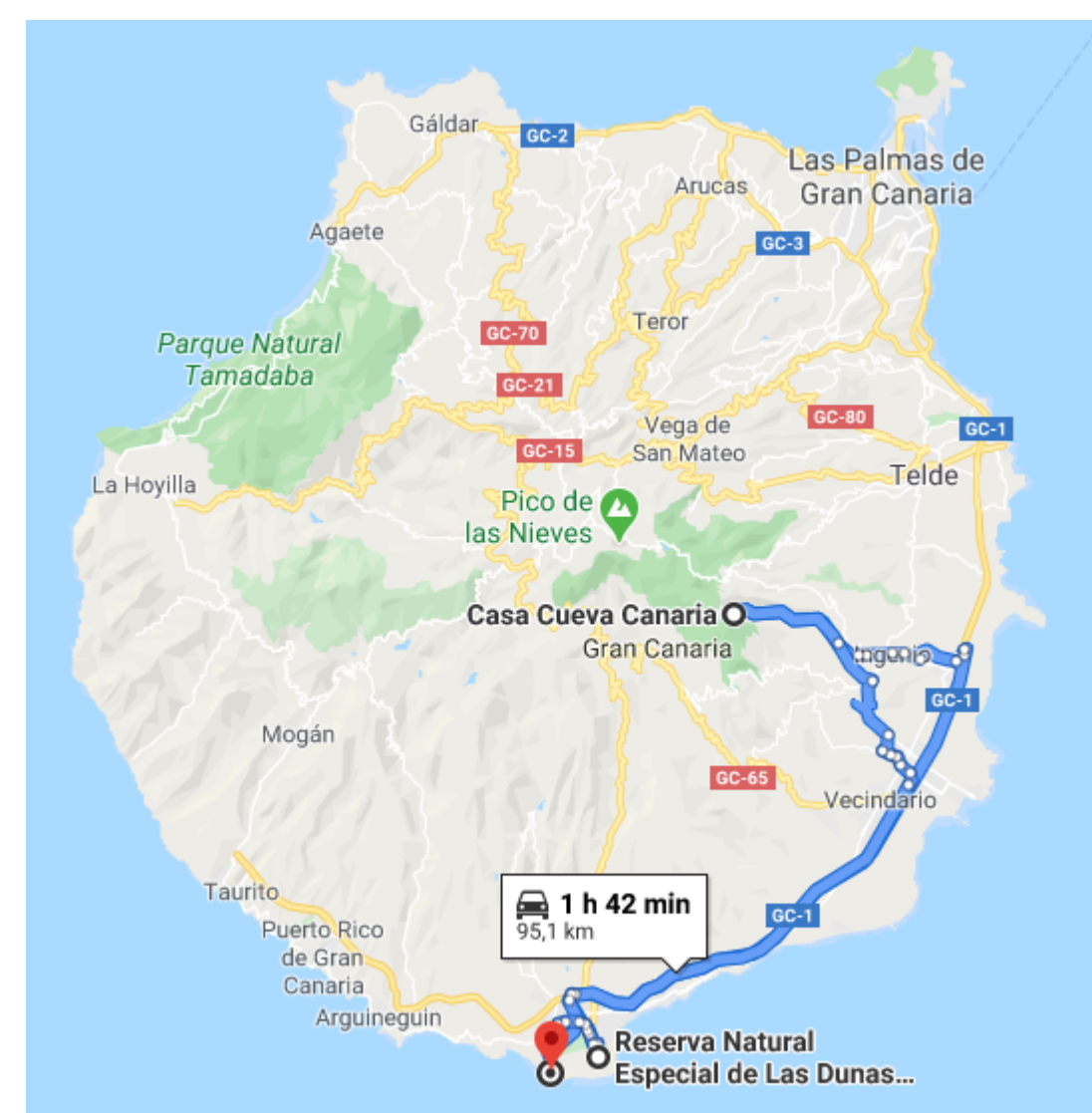
$$x_{ij} \in \{0, 1\}; \forall i, j = 1, \dots, n. \quad (7)$$

## 3. Experiment

The presented model has been experimented with a set of data obtained from the google maps website. Here, we used a series of points of interest (POIs) on the island of Gran Canaria.



What is intended is to find good routes based on the location and preferences of these given POIs. For this, what will be done is to consecutively apply the *Orienteering problem (OP)* model and thus generate as many routes as days the tourist is present on the island. In this way, when implementing the OP model for several times, what we obtained is something similar to applying the *Team orienteering problem (TOP)* model. In our case we will not implement TOP because the quality of each route that is calculated will depend on the POIs visited in the previous one.



## References

- [1] GAVALAS, D., KONSTANTOPOULOS, C., MASTAKAS, K., PANTZIOU, G. *A survey on algorithmic approaches for solving tourist trip design problems* Journal of Heuristics, 20(3) (2014), Pages 291-328.
- [2] LAPORTE, G., MARTELLO, S. *The selective travelling salesman problem* Discrete Applied Mathematics Volume 26, Issues 2&3, March 1990, Pages 193-207.
- [3] RAMESH, R., YOON, Y., KARWAN, M. *An optimal algorithm for the orienteering tour problem.* ORSA Journal on Computing, 4 (1992), Pages 155-165.
- [4] TSILIGIRIDES, T. *Heuristic methods applied to orienteering.* Journal of the Operational Research Society, 35 (9) (1984), Pages 797-809.
- [5] VANSTEENWEGEN, P., SOUFFRIAU, W., VAN OUDHEUSDEN, D. *The orienteering problem: A survey.* European Journal of Operational Research Volume 209, Issue 1, 16 February 2011, Pages 1-10.