



Universidad  
Politécnica  
de Cartagena



Universidad  
Politécnica  
de Cartagena | Campus  
de Excelencia  
Internacional

*Grado en Administración y Dirección de Empresas*

*Año académico 2019/2020*

## ***TRABAJO FIN DE GRADO***

***“UNA APLICACIÓN DEL PROBLEMA DEL VIAJANTE  
DE COMERCIO A LA DISTRIBUCIÓN DEL DINERO EN  
EFECTIVO EN LA REGIÓN DE MURCIA”***



FACULTAD DE  
CIENCIAS DE LA  
EMPRESA

UPCT

*Autor: Laura Montoya Torres*

*Director: Roberto Javier Cañavate Bernal*

# Índice general

<b>Capítulo 1: Introducción</b> .....	<b>4</b>
1.1. El problema del viajante de comercio en la historia. ....	4
1.2. Aplicaciones prácticas del TSP. ....	10
<b>Capítulo 2: Modelo del TSP</b> .....	<b>12</b>
2.1. Descripción formal y modelo.....	12
<b>Capítulo 3: Métodos de resolución del TSP</b> .....	<b>15</b>
3.1 Método de resolución exacto: Método de Ramificación y Acotación. ....	15
3.2. Métodos de resolución heurísticos.....	18
3.2.1. Heurística del vecino más próximo.....	19
3.2.2. Heurístico de Inserción más Barata .....	20
3.2.3. Heurístico 2 - Intercambio .....	20
<b>Capítulo 4: Aplicación del TSP</b> .....	<b>23</b>
4.1. El ciclo del efectivo. ....	23
4.2. Descripción del problema. ....	26
4.3. Programa WinQSB.....	29
4.4. Solución del problema. ....	30
4.4.1. Branch and Bound.....	31
4.4.2. Heurístico del vecino más próximo .....	31
4.4.3. Heurístico de inserción más barata .....	32
4.4.4. Heurístico 2 - intercambio .....	33
<b>Capítulo 5: Conclusiones</b> .....	<b>34</b>
<b>ANEXO</b> .....	<b>36</b>
<b>BIBLIOGRAFÍA</b> .....	<b>39</b>

## Índice de figuras

Figura 1. Clasificación de los problemas de rutas (Peñalva, 2015).....	5
Figura 2. Siete puentes del río Pregel .....	5
Figura 3. Juego Icoséan.....	7
Figura 4. Evolución TSP a lo largo de los años .....	9
Figura 5. Progreso en el TSP en la historia.....	9
Figura 6. Ruta óptima de un recorrido con 532 nodos usando Concorde (Applegate, Bixby, Chvátal y Cook, 2006).....	10
Figura 7. Ejemplo de formación de subcircuitos en el TSP.....	13
Figura 8. Solución óptima (Gestión de Operaciones, 2020).....	16
Figura 9. Generación de cotas para X1 (Gestión de operaciones, 2020).....	17
Figura 10. Datos de un problema de viajante .....	21
Figura 11. Solución inicial para TSP .....	21
Figura 12. Sucursales del Banco de España.....	24
Figura 13. El ciclo del efectivo (Banco de España, 2020).....	25
Figura 14. Centros operativos Iberpay (Informe anual Iberpay, 2018).....	26
Figura 15. Solución del TSP con el método Branch and Bound.....	31
Figura 16. Solución del TSP con el método Heurístico del vecino más próximo .....	31
Figura 17. Solución del TSP con el método Heurístico de inserción más barata .....	32
Figura 18. Solución del TSP con el método Heurístico 2 - intercambio.....	33

## Índice de tablas

Tabla 1. Ubicación entidades bancarias .....	28
Tabla 2. Distancia en kilómetros entre las entidades de crédito .....	29

## Capítulo 1: Introducción.

El objetivo de este trabajo es realizar una aplicación práctica del problema del viajante de comercio para observar su utilidad en problemas cotidianos y sus múltiples aplicaciones. Para ello, se va a realizar un recorrido por su historia, modelo y algunos métodos de resolución, como se verá a continuación.

### 1.1. El problema del viajante de comercio en la historia.

Un problema de rutas es todo aquel problema de optimización que se plantea cuando existen unos clientes que demandan un servicio y se tiene que hallar la mejor ruta para ellos en algún sentido de optimización (menor distancia, menor coste, etc.). Esta clase de problemas tienen una enorme importancia, pues se pueden aplicar a muchas situaciones reales, en especial a problemas de logística y distribución, aunque también a muchos otros campos como la secuenciación de tareas. Estos problemas son tratados en la Investigación Operativa dentro del campo denominado Teoría de Grafos. El elemento esencial es, tal y como indica su nombre, el *grafo*, que es un par  $G = (N, A)$  compuesto por un conjunto de elementos ( $N$ ) llamados vértices o nodos, y un conjunto de elementos ( $A$ ) que reciben el nombre de arcos o aristas, los cuales representan una conexión directa entre dos nodos.

Los problemas de rutas se pueden dividir en dos tipos, según el lugar donde se produce la demanda, distinguiendo entre los problemas de rutas sobre nodos, y los problemas de rutas sobre arcos. La diferencia entre estos es que, en el primero, la ruta óptima debe visitar todos los vértices o nodos determinados y en el segundo, esta ruta debe pasar por todas las aristas o arcos del grafo establecidos por el problema. Dicho de otro modo, en los problemas sobre nodos, es habitual que los objetivos estén representados por nodos, mientras que, en los problemas sobre arcos, son trayectos que deben ser recorridos (Peñalva, 2015).

Demanda	Restricciones de capacidad	Nombre habitual del problema	Otras restricciones
Nodos	NO	Viajante de Comercio TSP	
	SÍ	Problema de rutas de vehículos VRP	Recogida/distribución
Arcos	NO	Una componente conexa (Problema del Cartero Chino CPP)	Ventanas de tiempo
		Varias componentes conexas (Problema del Cartero rural RPP)	Otras
	SÍ	Problema de rutas con capacidades CARP	

Figura 1. Clasificación de los problemas de rutas (Peñalva, 2015)

El origen de los problemas de rutas sobre arcos se remonta al siglo XVIII cuando los residentes de Königsberg, una ciudad de Prusia, comenzaron a discutir si era posible trazar una ruta que pasase solo una vez por los 7 puentes que cruzaban el río Pregel y volviese al punto inicial (Figura 2). Se propuso este problema al matemático Leonhard Euler, quien demostró que no había ninguna ruta que pudiese realizarlo en un artículo (Euler, 1736) en el que utilizó una abstracción del mapa formada por nodos (regiones terrestres) y arcos (puentes), dando origen al concepto de nodo mostrado anteriormente.

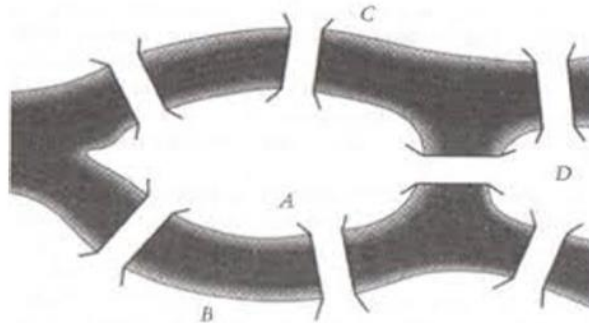


Figura 2. Siete puentes del río Pregel

Dentro de este tipo de problemas se encuentra el Problema del Cartero Chino (CPP, *Chinese Postman Problem*), introducido por el matemático Meigu Guan en 1960, en el que un cartero tiene que entregar la correspondencia en una zona antes de volver a la oficina, buscando que el trayecto efectuado suponga la distancia más corta posible. El CPP se puede resolver mediante un algoritmo, ya que es posible encontrar la solución óptima en un tiempo razonable,

por lo que es un problema técnicamente resuelto (Campos, 2020). Otro problema de arcos conocido es el denominado Problema del Cartero Rural (*RPP, Rural Postman Problem*) introducido en el año 1974, en el que se busca un camino que tenga la mínima distancia que recorra solo alguno de los arcos (Peñalva, 2015).

Por otro lado, dentro de los problemas de rutas sobre nodos encontramos el denominado Problema del Viajante de Comercio (*TSP, Travelling Salesman Problem*), el cual es uno de los problemas más estudiados dentro de la optimización combinatoria, que es un campo de las matemáticas aplicadas que trata de hallar el mejor valor de entre un número finito o numerable de soluciones viables. En este trabajo se trata el problema del viajante de comercio, consistente en minimizar la distancia recorrida por un viajante de comercio que debe visitar un número  $n$  de ciudades pasando sólo una vez por cada una de ellas y de forma que el punto o ciudad de origen sea el mismo que el de destino. Una definición informal de este problema podría ser: “si un viajante parte de una ciudad y las distancias a otras ciudades son conocidas, ¿cuál es la ruta óptima que debe elegir para visitar todas las ciudades y volver a la ciudad de partida?”. Si bien el enunciado original del problema hace referencia a un viajante de comercio y unas ciudades a visitar, es evidente que puede ser aplicado a muchas otras situaciones de características similares, tal y como mostraremos posteriormente. El TSP es un caso específico del Problema de rutas de Vehículos (*VRP, Vehicle Routing Problem*), en el que trata de obtener las rutas óptimas de un grupo de vehículos para servir a unos clientes. El TSP se diferencia del VRP en que el número de comerciantes que tienen que recorrer todas las ciudades no está restringido a uno, con lo que pueden ser varios comerciantes los que salen del almacén y cada uno tiene que visitar algunas de las ciudades, de forma que entre todos visiten el total de ciudades.

El TSP es aparentemente sencillo en su planteamiento, sin embargo, es uno de los problemas más complejos de resolver. Su dificultad de resolución radica en que el número de soluciones posibles crece significativamente conforme aumenta el número de nodos ( $n$ ). Esto sucede porque pertenece a la clase de problemas NP-Duros, lo que quiere decir que no se conoce ningún algoritmo que resuelva estos problemas en un tiempo polinomial (Garey y Jonson, 1979).

Los primeros en tratar problemas matemáticos relacionados con el problema del viajante fueron los matemáticos William Rowan Hamilton y Thomas Penyngton Kirkman, siendo también los primeros en considerar los denominados caminos y ciclos *hamiltonianos*, los cuales reciben este nombre en honor al primero de ellos. El camino hamiltoniano es un camino que pasa por cada uno de los vértices de un grafo exactamente una vez (Sensagent, 2013), y los

ciclos hamiltonianos son caminos hamiltonianos cuyo último vértice visitado es adyacente al primero (Matemáticas discretas, 2020). Hamilton y Kirkman crearon el juego Icosean (Figura 3) que consistía en obtener una ruta con los 20 puntos del juego utilizando únicamente un conjunto de caminos y retornar al nodo inicial. Se puede ver el trabajo de estos matemáticos en el libro *Graph Theory 1736-1936* escrito por Biggs, Lloyd y Wilson en 1976.

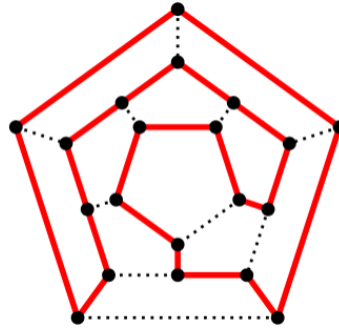


Figura 3. Juego Icosean

Se cree que el primero en estudiar la forma general del TSP fue Karl Menger en 1920 en Viena y Harvard, quien lo llamó “problema del mensajero”, pues lo encontró en los mensajeros postales. No obstante, se publicó en 1832 el libro *“El viajante de comercio: cómo debe ser y qué debe hacer para conseguir comisiones y triunfar en su negocio. Por un viajante de comercio veterano”* (Wiley and Sons, 1984), en el que, aunque trata principalmente otros aspectos de la profesión, se puede considerar la primera referencia bibliográfica, pues en el capítulo final definía el Problema del Viajante de Comercio de manera explícita. Sin embargo, fue Merrill M. Flood quien hizo popular el problema del viajante entre la comunidad matemática. En una entrevista que se le concedió, este recuerda haber oído hablar sobre ello a A. W. Tucker, quien anteriormente lo había escuchado de Hassler Whitney, en la Universidad de Princeton en 1931 (esta hipótesis no ha podido ser verificada por el propio Tucker al no poder acordarse de esto).

En los años 30 se empezó a trabajar propiamente en el problema del viajante. Merrill Flood, en la Universidad de Harvard, tuvo una función muy valiosa en la extensión del conocimiento de este problema. Este comenzó a interesarse por el TSP cuando comenzó a trabajar en la obtención de una ruta óptima para un autobús escolar. En el mismo momento, Karl Menger, en Viena, como ya hemos comentado anteriormente, expuso lo que se conocía como el problema del mensajero, que buscaba la ruta más corta que uniera una serie de puntos. Menger habló de una propiedad muy importante del TSP (Menger, 1931): es un problema NP-

duro. Asimismo, planteó lo que se llamaría más adelante como *algoritmo del vecino más próximo*, que consiste en empezar por el punto inicial y visitar cada vez el nodo más próximo sin regresar al punto de origen. Sin embargo, se dio cuenta de que, generalmente, este camino no era el más corto. Este algoritmo es heurístico, esto quiere decir que se obtiene una solución razonablemente buena, aunque no tiene por qué ser la solución óptima.

En las décadas de 1950 y 1960 fue el momento en que el problema logró mucha popularidad, en parte, gracias a la campaña de publicidad de *Procter and Gamble* (P&G), que consistía en un concurso en el que el premio se lo llevaría la persona que ayudara a obtener la ruta más corta entre 33 ciudades de Estados Unidos.

En estos años, por la similitud del problema del viajante con los problemas de optimización combinatoria hasta entonces conocidos, el problema de asignación y el del transporte, se realizaron un gran número de estudios de este problema.

Como he podido aprender durante mis estudios de grado en ADE, el problema de asignación trata de asignar  $n$  tareas a  $n$  agentes para lo cual se determina un coste de asignación de cada tarea a cada trabajador y se busca minimizar el coste total de asignaciones. Por otro lado, el problema del transporte busca distribuir de forma óptima unos recursos desde  $m$  orígenes a  $n$  destinos, conociendo los costes de transporte unitarios desde cada origen a cada destino.

Se puede destacar el estudio realizado en 1954 por Dantzig, Fulkerson y Johnson: “Soluciones de un problema del viajante de gran tamaño” (Dantzig, Fulkerson y Johnson, 1954: 393-410). En este artículo, el TSP fue clasificado como un problema de programación lineal entera y, además, se resolvió el problema para un total de 49 ciudades (las cuales se encontraban en cada estado de Estados Unidos). Este estudio se considera de gran importancia, pues lograron encontrar la solución a un problema del viajante con un gran número de ciudades y usaron un método completamente nuevo que contenía las técnicas más recientes de programación lineal. Esto dio lugar al método de los planos de corte, el cual, con el tiempo, se convertiría en el método de Gomory (Gomory, 1958:275-278) y más tarde al conocido actualmente como método de ramificación y acotación, uno de los más utilizados para solucionar problemas de programación lineal entera.

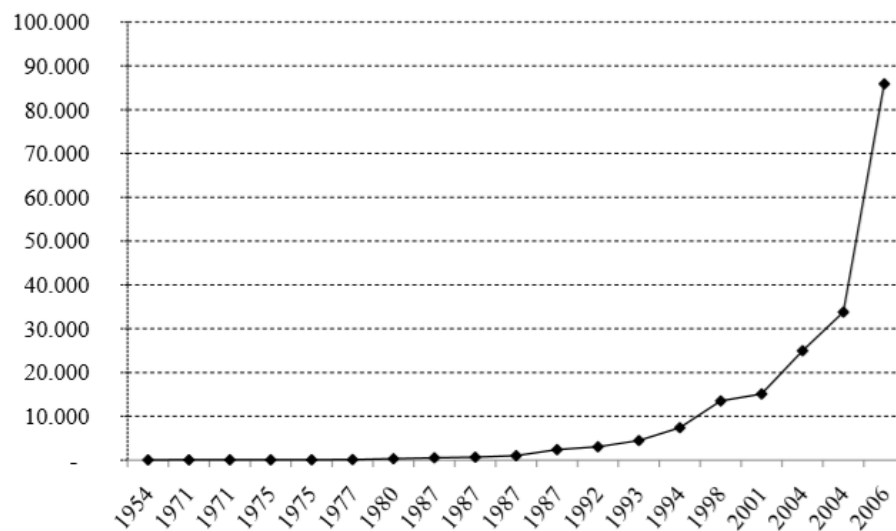
Posteriormente al estudio de Dantzig et al., surgieron otros estudios con nuevos algoritmos capaces de resolver problemas cada vez con un mayor número de ciudades, consiguiendo resolver problemas con hasta 85900 nodos hasta 2006. En la Figura 4, se puede ver una evolución de los tamaños máximos de problemas TSP resueltos desde 1954 hasta 2006.



Año	Autores	Nodos
1954	G. Dantzig, R. Fulkerson, S. Johnson	49
1971	M. Held, R.M. Karp	64
1975	P.M. Camerini, L. Fratta, F. Maffioli	67
1977	M. Grötschel	120
1980	H. Crowder and M. W. Padberg	318
1987	M. Padberg and G. Rinaldi	532
1987	M. Grötschel and O. Holland	666
1987	M. Padberg and G. Rinaldi	2392
1994	D. Applegate, R. Bixby, V. Chvátal, W. Cook	7397
1998	D. Applegate, R. Bixby, V. Chvátal, W. Cook	13509
2001	D. Applegate, R. Bixby, V. Chvátal, W. Cook	15112
2004	D. Applegate, R. Bixby, V. Chvátal, W. Cook	18512
2004	D. Applegate, R. Bixby, V. Chvátal, W. Cook, K. Helsgaun	24978
2004	W. Cook, Espinoza and Goycoolea	33810
2006	D. Applegate, R. Bixby, V. Chvátal, W. Cook	85900

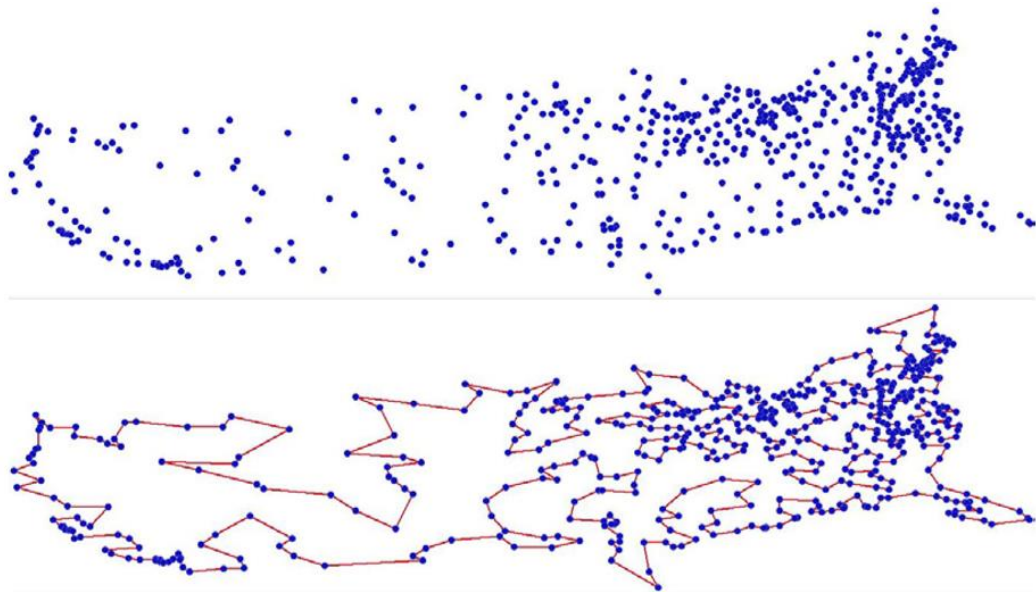
*Figura 4. Evolución TSP a lo largo de los años*

En la Figura 5 se muestran estos mismos datos en forma de gráfica para observar más claramente el rápido avance producido en los últimos años.



*Figura 5. Progreso en el TSP en la historia*

En el año 1990, Vasek Chvátal y William J. Cook, ayudados por David Applegate y Robert Bixby, consiguieron desarrollar un programa informático llamado Concorde, responsable de la resolución del problema TSP más grande resuelto hasta la fecha (85900 nodos), que provenía de una aplicación a circuitos integrados. En la figura 6, podemos contemplar la elección de la ruta óptima para una instancia del TSP de 532 nodos.



*Figura 6. Ruta óptima de un recorrido con 532 nodos usando Concorde  
(Applegate, Bixby, Chvátal y Cook, 2006)*

## 1.2. Aplicaciones prácticas del TSP.

Por su nombre se puede pensar que el problema del viajante tiene una aplicación muy concreta, es decir, la elección del recorrido eficiente para un vendedor. Sin embargo, el TSP se puede aplicar desde cualquier campo o problema en el que se necesite decidir el orden de ‘visita’ de los nodos para reducir costes, hasta, por ejemplo, la genética. A continuación, se enumeran algunas de las aplicaciones más utilizadas del TSP:

- **Logística:** las aplicaciones más directas de este problema se centran en la logística. Un recorrido eficiente para el transporte de personas, mercancías y vehículos alrededor de un conjunto de ciudades o clientes encaja en la filosofía del TSP. Entre las muchas aplicaciones dentro de este campo se pueden destacar: vendedores y turistas para determinar cuál es la mejor ruta con la que poder visitar una serie de lugares y regresar al punto de origen; organización de rutas escolares (una de las aplicaciones iniciales del TSP, estudiada por Merrill Flood); y la distribución de correo, ya que, a pesar de que este se ajusta más a un problema de rutas sobre arcos, también puede resolverse utilizando el TSP.

- **Industria:** no son tantas las aplicaciones en la industria como lo son en la logística, sin embargo, la aplicación del TSP a este campo ha permitido la reducción de costes en diversos ámbitos de la industria. Las aplicaciones más importantes son:

- *Secuenciación de tareas*: dada una máquina que tiene que llevar a cabo un conjunto de tareas sin importar el orden de estas, se puede aplicar el TSP para minimizar el tiempo total de ejecución de dichas tareas considerando que cada una de las tareas es un nodo y que el tiempo que se tarda en poner a punto la máquina para ejecutar la tarea  $j$  una vez efectuada la tarea  $i$  es un arco de longitud  $t_{ij}$ .
- *Producción de circuitos electrónicos*: su utilización se centra tanto en la determinación del orden óptimo para taladrar las placas, considerando como nodo el lugar en el que hay que taladrar y como costes la distancia entre cada punto, como en la elección de los caminos óptimos precisos para conectar los chips entre sí de forma que se minimice la cantidad de cable.

• **Organización de datos**: organizar datos en clústers o grupos de elementos con características muy parecidas, considerando la similitud entre un par de datos como un camino hamiltoniano con el objetivo de maximizarla.

Estas son algunas de las aplicaciones más comunes que se pueden encontrar del TSP, aunque también abarca muchos otros campos, cada día más dispares. Entre las últimas aplicaciones de este problema se pueden encontrar: la planificación de caminos de senderismo, minimizar el gasto de papel o patrones de corte en la industria cristalera.

## Capítulo 2: Modelo del TSP

### 2.1. Descripción formal y modelo.

Después de profundizar en la historia del problema del viajante, es necesario revisar alguna de sus múltiples formulaciones existentes. Como ya se ha explicado anteriormente, un grafo  $G = (N, A)$  consiste en un conjunto de elementos ( $N$ ) llamados vértices o nodos, y un conjunto de elementos ( $A$ ) que reciben el nombre de arcos o aristas, los cuales representan una conexión directa entre dos nodos. El TSP se aplica a grafos  $G = (N, A)$  completos, es decir, grafos en los que todas las aristas posibles están presentes<sup>1</sup>. Las ciudades a visitar se denotan con los vértices  $i = 2, \dots, n$  y el vértice 1 se considera la ciudad de origen y destino. A cada arco  $(i, j)$  se le asocia un valor no negativo  $c_{ij}$ , que representa el coste de viajar del vértice  $i$  al  $j$ . Si  $G$  es un grafo dirigido, es decir, la distancia de ida no tiene necesariamente que ser igual a la de vuelta, entonces la matriz de costes  $C$  es asimétrica y el problema recibe el nombre de TSP asimétrico (ATSP). Sin embargo, si  $G$  es un grafo no dirigido, es decir,  $c_{ij} = c_{ji}$  para todo  $(i, j) \in A$ , entonces la matriz de costes es simétrica y el problema se denomina TSP simétrico (STSP). Una de las variantes que se puede encontrar es el problema con múltiples viajantes (mTSP). En este último caso existen  $m$  vendedores localizados en un único nodo al que denominamos almacén, y tienen que ser visitados una serie de nodos o ciudades denominados nodos intermedios (Veiga Losada, 2013). En el mTSP los vendedores tienen que empezar y terminar en el almacén, de manera que cada nodo intermedio se visite solo una vez, minimizando el coste de los circuitos.

Existen diferentes formulaciones matemáticas del problema TSP. La primera de ellas consiste en definir las variables de decisión  $x_{ij}$  para todo  $(i, j) \in A$ , de forma que adquieran el valor 1 si el arco  $(i, j)$  forma parte de la solución y 0 en caso contrario. Con estas variables la función objetivo queda de la forma siguiente:

$$\text{minimizar } \sum_{i,j=0}^n c_{ij}x_{ij}$$

Sujeto a las siguientes condiciones:

---

<sup>1</sup> Si dos nodos no están unidos por una arista, entonces se añade una arista que los una con un peso lo suficientemente importante como para asegurar que no pertenecerá al recorrido óptimo.

$$(1) \quad \sum_{i=0}^n x_{ij} = 1 \quad j = 0, \dots, n$$

$$(2) \quad \sum_{j=0}^n x_{ij} = 1 \quad i = 0, \dots, n$$

La primera igualdad (1) quiere decir que a cada nodo sólo puede llegar un arco, mientras que la segunda (2), que solo puede salir un arco de cada nodo. Las restricciones mencionadas son necesarias, pero no suficientes, ya que no impiden la existencia de los denominados *subtours* o subcircuitos, como podemos ver en la figura 7.

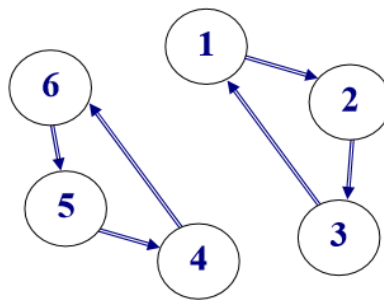


Figura 7. Ejemplo de formación de subcircuitos en el TSP

Obsérvese que  $x_{12} = x_{23} = x_{34} = x_{45} = x_{56} = x_{61} = 1$ , por lo que se cumplen las restricciones referidas anteriormente, esto es, de cada nodo entra y sale a lo sumo un arco, no obstante, este camino no es válido para el TSP.

Por consiguiente, tenemos que incorporar al problema alguna restricción más. Se puede observar en la figura 7, que en el subconjunto  $\{1,2,3\}$ , hay tres arcos que unen el subconjunto entre sí, al igual que en el subconjunto  $\{4,5,6\}$ . Observamos, en definitiva, que una solución del problema anteriormente formulado contendría subcircuitos cuando exista un subconjunto propio de nodos de modo que los arcos de la solución que conecten dichos nodos sean igual al número de nodos.

Por estas razones se añade la siguiente restricción a la formulación anterior:

$$(3) \quad \sum_{i,j}^n x_{ij} \leq |K| - 1 \quad 1 \leq i \neq j \leq n, K \subsetneq N$$

$$0 \leq x_{ij} \leq 1, \quad x_{ij} \in Z, \forall i, j$$

Esta restricción quiere decir que en todo subconjunto propio de nodos del problema el número de arcos que unen los nodos del subconjunto entre sí debe ser estrictamente menor que el número de nodos del subconjunto.

Sin embargo, esta última condición no es eficiente, ya que da lugar a demasiadas restricciones, tantas como el número de subconjuntos propios dentro del conjunto de nodos del problema, por lo que se generarían  $2^{n+1} - 2$  restricciones. En el libro de Papadimitriou y Steiglitz se puede encontrar una formulación más práctica realizada por Tucker, en la que esencialmente cambia la condición (3).

$$\begin{aligned} & \text{minimizar} && \sum_{i,j=0}^n c_{ij}x_{ij} \\ & \text{sujeto a} && (1) \quad \sum_{i=0}^n x_{ij} = 1 && j = 0, \dots, n \\ & && (2) \quad \sum_{j=0}^n x_{ij} = 1 && i = 0, \dots, n \\ & && (3) \quad u_i - u_j + nx_{ij} \leq n - 1 && 1 \leq i \neq j \leq n \\ & && && 0 \leq x_{ij} \leq 1, && x_{ij} \in Z, \forall i, j \end{aligned}$$

Las variables  $u_i$ , con  $i = 1, \dots, n$ , se denominan condiciones de Miller-Tucker, pues fueron propuestas por Miller, Tucker y Zemlin en 1960. Se puede probar, véase el trabajo de Francisco José Veiga Losada, *El problema del viajante* (2013), que esta última formulación del problema con la nueva restricción (3) define el problema TSP, evitando la existencia de subcircuitos en la solución y disminuyendo drásticamente el número de restricciones frente a la restricción (3) propuesta anteriormente.

## Capítulo 3: Métodos de resolución del TSP

Existen multitud de métodos para resolver el TSP, tanto exactos como heurísticos. En este trabajo vamos a estudiar el método de Ramificación y Acotación como método exacto, y, como métodos heurísticos, el algoritmo del vecino más próximo, heurístico de inserción más barata y heurístico 2-intercambio.

### 3.1 Método de resolución exacto: Método de Ramificación y Acotación.

En el artículo de Dantzig, Fulkerson y Johnson de 1954 se podían observar pequeños rasgos que hacen referencia al método de Ramificación y Acotación, sin embargo, fueron Land y Doig, en 1960, quienes acuñaron por primera vez el término *Branch and Bound* (Ramificación y Acotación), y Little, Murty, Sweeney y Karel los primeros en ponerlo en práctica (Little, Murty, Sweeney y Karel, 1963).

Este método es un algoritmo que fue creado para resolver problemas de programación lineal entera, es decir, en cuya solución óptima algunos o todos los valores de las variables deben ser números enteros (Gestión de Operaciones, 2020). Esta técnica nos permite considerar todas las posibles soluciones óptimas, evitando examinarlas una a una. La base del algoritmo es la aplicación reiterada de los siguientes pasos (se considera un problema de minimización, siendo el caso de maximización idéntico cambiando “cotas inferiores” por “cotas superiores”):

- **Ramificar**, es decir, subdividir el problema en al menos dos subproblemas separados.
- **Acotar** cada uno de los subproblemas obtenidos, esto es, obtener cotas inferiores de cada una de las ramificaciones.

Habitualmente, las cotas inferiores de la fase de acotación se obtienen resolviendo el problema de programación lineal que se obtiene al relajar las restricciones de integridad en el subproblema dado.

En general, el algoritmo de Ramificación y Acotación podría resumirse en resolver en primer lugar el problema relajado y en el caso el valor de alguna variable de la solución no fuese entero, pero tuviera que serlo, añadir restricciones para que este valor sea entero (Infantes, 2018).

Veamos este método de resolución con más claridad en un ejemplo (Gestión de Operaciones, 2020), considerando el modelo de programación entera siguiente:

$$\begin{aligned}
 \text{Max} \quad & 4X_1 + 6X_2 \\
 \text{s.a.} \quad & 2X_1 + 4X_2 \leq 12 \\
 & 4X_1 + 3X_2 \leq 16 \\
 & X_1, X_2 \geq 0 \text{ Enteros}
 \end{aligned}$$

Este modelo se resuelve como si fuera de programación lineal y, si en la solución que obtenemos las variables tuviesen valores enteros, es decir, cumpliesen las restricciones de integridad, esta sería la solución óptima.

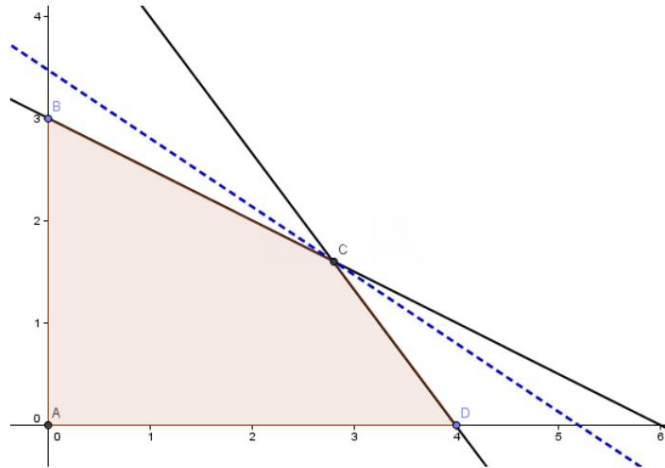


Figura 8. Solución óptima (Gestión de Operaciones, 2020)

En este grafo se puede ver la solución del problema,  $X_1 = 2,8$  y  $X_2 = 1,6$  con un valor óptimo de  $V = 20,8$ . En esta solución, los valores de la solución correspondientes a las variables  $X_1$  y  $X_2$  no cumplen las condiciones de integridad, por lo que se establecen unas restricciones adicionales. Se escoge una de las dos variables cuyo valor en la solución no sea exacto para generar estas restricciones. Conviene señalar que resulta indiferente en este caso cuál de las variables sea escogida, ya que no es posible determinar a priori cuál de las opciones supondrá alcanzar la solución de la forma más rápida posible.

En este ejemplo se ha escogido en primer lugar la variable  $X_1$  y se ramifica creando dos nuevas restricciones, una obligando a que la variable  $X_1$  sea menor o igual que 2 (valor entero inferior más próximo al valor de  $X_1$  en la solución óptima del problema a ramificar) y otra estableciendo que la variable  $X_1$  debe ser mayor o igual que 3 (valor entero superior más próximo al valor de  $X_1$  en la solución óptima del problema a ramificar).



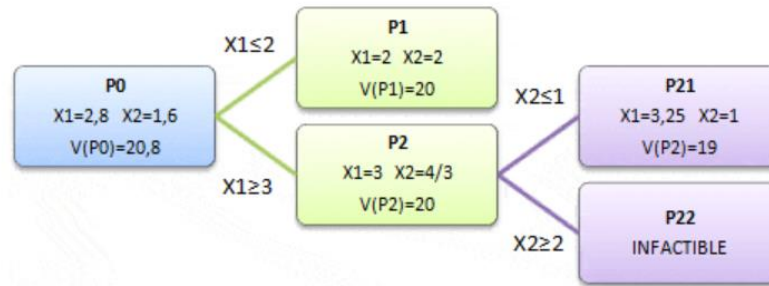


Figura 9. Generación de cotas para  $X1$  (Gestión de operaciones, 2020)

Como se puede observar, la solución obtenida de la relajación del problema P1 cumple las condiciones de integridad, aunque el valor óptimo es inferior, lo que es normal al haber añadido restricciones. Dado que la solución obtenida de P1 cumple todas las restricciones de integridad, dicho problema no deberá volver a ramificarse. A continuación, se resuelve la versión relajada del subproblema P2. Como puede observarse, la solución obtenida contiene valores fraccionarios de la variable  $X2$ , con lo que se debe ramificar el subproblema P2 añadiendo restricciones respecto de la variable  $X2$ . De este modo, se obtienen los subproblemas P21 y P22, siendo las soluciones óptimas de sus respectivas relajaciones las indicadas en la Figura 9. En la solución P21 también hay variables con valores fraccionarios y el valor óptimo es inferior a la solución óptima que se obtuvo al principio (P1), por lo que se puede asegurar que este subproblema no contiene la solución óptima del problema original. La solución de la relajación del subproblema P22 no es factible, con lo que tampoco contiene la solución óptima del problema original y no se continúa ramificando. Por lo tanto, se concluye que la solución óptima del problema original es la obtenida en el subproblema P1,  $X1=2$  y  $X2=2$ , con valor óptimo  $V=20$ .

Al ir avanzando en el árbol de búsqueda se compara en todo momento la solución obtenida en cada subproblema con la mejor solución obtenida hasta el momento, la denominada solución titular. Si la nueva solución óptima es mejor que la titular, esta se convierte en la nueva titular. En caso contrario, se descarta. Además, el problema al que corresponda la solución titular deberá ser ramificado si contiene valores fraccionarios de alguna variable que debiera tomar valores enteros.

El TSP se puede ser resuelto utilizando una versión adaptada del algoritmo Branch and Bound. La base de este sistema radica en observar que si se eliminan en el TSP las restricciones para evitar subcircuitos, el problema resultante se puede interpretar como un problema de asignación. De este modo, el proceso comienza resolviendo el problema de asignación

asociado al TSP original. Como es bien conocido los problemas de asignación se pueden resolver de forma exacta y eficiente a través del Algoritmo Húngaro. Una vez se obtiene la solución del problema de asignación asociado, pueden darse dos situaciones: que la solución obtenida no contenga subcircuitos, con lo cual sería la solución óptima del TSP original ya que verificaría las restricciones eliminadas, o que la solución sí contenga subcircuitos, en cuyo caso sería necesario ramificar el problema de asignación. El proceso de ramificación sería el siguiente: se escoge uno de los subcircuitos de la solución y para cada uno de los arcos contenidos en el subcircuito, se genera un nuevo subproblema (rama) en el que la variable asociada al arco sea igual a cero, es decir, impidiendo que dicho arco se incluya en la solución. Por ejemplo, si el subcircuito  $A \rightarrow B \rightarrow C \rightarrow A$  fuera parte de la solución, entonces se tendrían las siguientes variables  $X_{AB} = X_{BC} = X_{CA} = 1$  en la solución del problema de asignación. Pues bien, en este caso para romper el circuito se crearían tres subproblemas: uno con la restricción añadida  $X_{AB} = 0$ , otro con la restricción  $X_{BC} = 0$  y otro con  $X_{CA} = 0$ . Todos ellos sería a su vez problemas de asignación que podrían resolverse también de forma eficiente.

A partir de este paso, el procedimiento continúa de manera similar al Branch and Bound para problemas lineales: cada vez que la solución de un subproblema no contiene subcircuitos, entonces dicha solución sería una solución factible del TSP y ya no es necesario volver a ramificar este subproblema. Si se denomina *solución incumbente* a la mejor solución factible del TSP encontrada hasta el momento, entonces cada que vez que se obtenga una solución factible del TSP se debe comparar su coste con el de la solución incumbente: si el coste es mejor que el de la incumbente, dicha solución factible se convierte en la nueva incumbente; en caso contrario, se descarta ya que no puede ser óptima.

Sin embargo, si la solución del subproblema contiene subcircuitos, entonces debe ser ramificado solo si su coste es mejor que la incumbente, ya que en caso contrario (coste peor que la incumbente) se descarta el subproblema al ser evidente que ninguna ramificación de dicho subproblema puede contener la solución óptima.

### **3.2. Métodos de resolución heurísticos.**

«Un heurístico es una técnica que busca buenas soluciones con un tiempo de computación razonable sin garantizar la optimalidad», según afirma C.R. Reeves. Los algoritmos heurísticos son un conjunto de técnicas que nos permiten resolver los problemas de

manera eficiente, aunque no se puede asegurar que la solución obtenida resulte ser la solución óptima del problema.

Existen una gran cantidad de algoritmos para el TSP, pero nos centraremos en los que se usan en WinQSB, ya que es el software que utilizaremos para la resolución del problema. Por consiguiente, se presentan dos tipos de algoritmos heurísticos para el TSP: constructivos y de mejora (o búsqueda) local. Los primeros son para construir una solución factible, mientras que en los segundos se parte de la solución factible y se intenta mejorar. Los algoritmos 3.2.1 y 3.2.2 son “de construcción” que, como su nombre indica, consisten en ir construyendo una solución para el problema escogiendo la mejor opción en cada repetición. Estos suelen ser modelos deterministas. El algoritmo 3.2.3 es de búsqueda local, que es un procedimiento que parte de una solución del problema y paso a paso, con cada iteración, va mejorando la solución hasta que no se pueda obtener con el mismo procedimiento otra con mejor valor.

### 3.2.1. Heurística del vecino más próximo

En 1931, Menger definió este método, observando que las soluciones generalmente no eran óptimas. Este método es importante debido a su sencillez, por lo que es una de las heurísticas más utilizadas para el TSP.

Según esta heurística, desde la ciudad de origen se busca la ciudad más próxima y se añade a la ruta. Una vez realizado esto, se busca la ciudad más cercana a esta última, sin haber sido incorporada, y se añade a la ruta. Este proceso se repite hasta haber recorrido todos los nodos o ciudades y, una vez pasado por todos, se regresa a la ciudad inicial desde el último punto visitado (Martí, 2003).

Conforme se va formando la solución del problema, suele comenzar bien, realizando trayectorias a las ciudades más cercanas con una distancia corta. Sin embargo, a medida que continúa y en especial al finalizar, esta solución puede ir empeorando, ya que cada vez pueden aparecer aristas o trayectos más largos entre nodo y nodo, lo que se conoce como “miopía” del procedimiento, pues cada vez se va escogiendo la mejor opción sin observar si esto nos puede llevar a problemas posteriores.

Como describe el algoritmo, para hallar la solución se requiere un número de operaciones de orden  $O(n^2)$  y se conoce una cota del error máximo de la solución obtenida con esta heurística:

$$\frac{\text{longitud de la ruta 'vecino cercano'}}{\text{longitud de la ruta óptima}} \leq \frac{1}{2} [\log_2(n)] + \frac{1}{2}$$

### 3.2.2. Heurístico de Inserción más Barata

Esta heurística es conocida porque es simple y sencilla de implementar, las soluciones que obtenemos son aceptables y fácilmente puede manejar restricciones complicadas (Alssager, Ali Othman, Ayob, 2017). En un principio fue introducido para resolver el problema del viajante de comercio, y hoy en día se utiliza para resolver problemas de rutas de vehículos e, incluso, extensiones de este. Su funcionamiento es el siguiente:

- 1) Primero hay que seleccionar dos nodos, diremos que son  $i$  y  $j$ , entre los cuales la distancia total de ida y vuelta es la menor. El subtour inicial sería similar a  $i-j-i$ .
- 2) Se selecciona un nodo distinto a los que componen el subtour,  $k$ , que incremente en la menor cantidad (más barata) la longitud del subtour. Es decir, el nodo  $k$  tiene la menor cantidad de  $C_{ik}+C_{kj}-C_{ij}$  para todos los  $(i, j)$  en el subtour.
- 3) Se inserta el nodo  $k$  entre  $i$  y  $j$ . Volver a realizar el paso 2) hasta que todos los nodos del grafo estén incluidos en el subtour.

En el caso de que se trate de un problema de maximización, se escogerá el nodo  $k$  que corresponde a la mayor cantidad  $C_{ik}+C_{kj}-C_{ij}$ .

El número de cálculos que requiere este algoritmo es de orden  $O(n^2 \log_2(n))$ . La cota del error máximo que se puede cometer con este método es:

$$\frac{\text{longitud de la ruta 'inserción más cercana'}}{\text{longitud de la ruta óptima}} \leq 2$$

### 3.2.3. Heurístico 2 - Intercambio

Este heurístico es conocido como de búsqueda local. Se parte de una solución completa anterior generada habitualmente por un heurístico de construcción como el heurístico del vecino más próximo o el de inserción más barata y se intenta mejorar cambiando 2 aristas de la solución. Para ello, en primer lugar, se toma de una solución completa del problema dos arcos no adyacentes,  $(i, j)$  y  $(k, l)$ . Si se eliminan estos arcos, se formarán dos partes separadas, digamos A y B. Finalmente se reconecta el circuito con una de las dos opciones: 1) dando la vuelta a los arcos en A y conectando  $(l, j)$  y  $(k, i)$ ; o 2) dando la vuelta a los arcos en B y conectando  $(i, k)$  y  $(j, l)$ . Si esta nueva ruta es mejor que la anterior, esto es, tiene un coste menor si la función objetivo está minimizando o si es mayor si está maximizando, la ruta mejorada pasa a ser la solución actual. Si esta nueva ruta no es mejor que la anterior, se busca algún otro intercambio que no se haya realizado. Este proceso se repite hasta que ninguna otra

transformación del conjunto de transformaciones produzca una mejora. En el ejemplo mostrado en la figura 10 se muestran los datos de un problema de viajante de comercio (TSP).

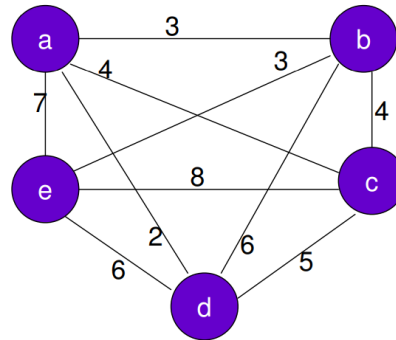


Figura 10. Datos de un problema de viajante

Una solución inicial al problema TSP se encuentra en la figura 11.

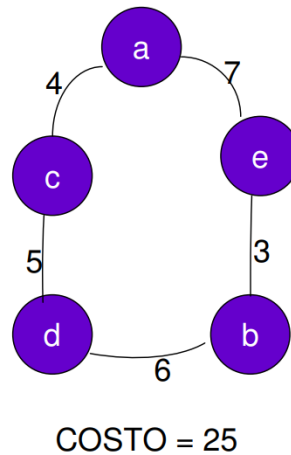


Figura 11. Solución inicial para TSP

A partir de la solución inicial, se generan unos posibles circuitos intercambiando 2 arcos no adyacentes. Por ejemplo, al cambiar el arco (a, e) y el arco (b, d) tendríamos 2 posibles opciones:

Opción 1: a – b – e – d – c – a.

Opción 2: a – d – b – e – c – a.

Teniendo en cuenta el coste de los arcos de la figura 10, en este caso, si el coste de los nuevos arcos de la opción 1 (a, b) y (e, d) es menor que el original (a, e) y (b, d), esta pasa a ser la solución original, ya que está mejorada. Eliminaremos los arcos (a, e) y (b, d) y los sustituiremos por (a, b) y (e, d).

Sin embargo, este método heurístico, como cualquiera de ellos, no nos garantiza obtener la solución óptima, porque es un algoritmo del tipo búsqueda local en el que no está permitido empeorar, por lo que llegaremos a óptimos locales, pero no podemos asegurar que sean también óptimos globales.

## **Capítulo 4: Aplicación del TSP.**

### **4.1. El ciclo del efectivo.**

Como ya se ha comentado con anterioridad, el TSP se puede aplicar a muchos campos o situaciones en las que se necesite elegir la mejor ruta. En este trabajo, el TSP se ha aplicado a la distribución del dinero en efectivo en la Región de Murcia, desde que se recoge en los depósitos auxiliares, es decir, donde el Banco de España tiene depósitos de billetes a lo largo del territorio español para conseguir una circulación de billetes más eficaz y una mejor condición de estos, hasta las entidades de crédito. Dicho de otra forma, esta distribución del dinero en efectivo se refiere al reparto de los billetes en euros a las entidades de crédito de la Región de Murcia desde estos depósitos. Para entender esta aplicación del TSP, vamos a explicar el contexto en el que se sitúa esta distribución del dinero en efectivo.

El Banco de España es uno de los bancos centrales nacionales que pertenece al Eurosistema. Tiene muchas funciones, entre las que se puede destacar: definir y ejecutar la política monetaria, operaciones de cambio de divisas, supervisar la solvencia, supervisar el buen funcionamiento del sistema financiero, etc. Una de las funciones que más nos interesa para este trabajo de las que realiza el Banco de España es la emisión de billetes y monedas en euros en nombre del Estado español (Banco de España, 2020).

Las entidades de crédito son las encargadas de solicitar al Banco la cuantía monetaria que necesitan para cubrir su demanda. El Banco, de forma gratuita y sin ofrecer restricciones atiende las peticiones, y una vez ingresada la cantidad correspondiente por las entidades realiza la emisión de billetes y monedas, los cuales distribuye a una red de sucursales y depósitos auxiliares, para que después circulen a su siguiente punto de distribución, como los almacenes de las entidades de crédito, sus oficinas, a los cajeros expendedores y a las sedes de empresas. Los depósitos auxiliares fueron creados en 2005 con el objetivo de que el movimiento del efectivo fuese más fluido y eficaz entre el banco y las entidades de crédito, y que dicho efectivo se encuentre en condiciones óptimas de conservación. Se denomina auxiliar porque se trata de un servicio adicional al servicio de caja del banco, que también cuenta con sus propias sucursales (Los Depósitos Auxiliares del Banco de España, 2005). El Banco de España tiene su sede en Madrid, la capital de España, y, a lo largo del territorio español, dispone de 15 sucursales, como se puede ver en la imagen siguiente (figura 12), las cuales hacen la función del Banco de España en cada una de sus ubicaciones. En el caso de la sucursal en Murcia, podemos destacar la realización de actividades como: operaciones de cambio de pesetas por

euros, recogida de billetes y monedas falsos y de billetes deteriorados, registro de documentación, entre otras.



*Figura 12. Sucursales del Banco de España*

El transporte de los billetes y monedas desde los depósitos auxiliares hasta las entidades de crédito es realizado por las empresas de transporte de fondos. Estas empresas deben de seguir unos estrictos requerimientos de seguridad y vigilancia para poder realizar el transporte de fondos, los cuales se pueden encontrar en la Orden INT/314/2011, de 1 de febrero, sobre empresas de seguridad privada.

El público para sus operaciones diarias necesita efectivo, que obtiene retirándolo de las ventanillas de las entidades de crédito y, en especial, de los cajeros automáticos. Además, los comerciantes y la gente depositan efectivo en estas entidades a través de los mismos medios por el que lo retiran.

Las compañías de transporte de fondos se encargan de entregar al Banco de España el dinero que recogen de las entidades de crédito, es decir, el efectivo que el público ingresa en la entidad. Una vez que el Banco de España recibe estos billetes y monedas, realiza una revisión de estos con el objetivo de mantener en buen estado la circulación de efectivo y de retirar los que sean falsos o estén en mal estado (Banco de España, 2020).

A todo este proceso de retirada y entrega de efectivo se le denomina el ciclo del efectivo. En la figura siguiente (Figura 13) se puede apreciar un esquema de lo relatado con anterioridad.





Figura 13. El ciclo del efectivo (Banco de España, 2020)

Las entidades de crédito eran las encargadas de realizar el transporte de los billetes desde el Banco de España hasta donde estuviesen situadas. Sin embargo, a partir de los años 80 esto cambió, siendo empresas de seguridad las encargadas de este transporte. Estas empresas crearon bases operativas para guardar el dinero en efectivo, seleccionar grandes entregas de efectivo y centralizar el transporte de fondos dentro de los límites respectivos. En términos jurídicos, el Banco de España cuenta con un contrato firmado entre este y una entidad representante de un conjunto de entidades de crédito, denominada entidad gestora, que se encarga de la custodia y gestión de los fondos.

Dentro del sistema de depósitos auxiliares (SDA) encontramos la empresa Iberpay, que es la entidad gestora de los SDA. Cada entrega y recepción de efectivo lo realiza la entidad gestora en nombre del Banco de España. El Banco de España le impuso la gestión de los SDA y de mejorar el flujo de efectivo, ya que se encarga de su logística (Iberpay, 2020).

Iberpay delega en empresas especializadas la seguridad, el transporte, custodia y tratamiento de los billetes, empresas como Prosegur o Loomis. Estas empresas deben seguir unas medidas de vigilancia y seguridad muy estrictas, como puertas de acceso blindadas, ventanas protegidas, sistemas de captación y registro de imágenes, entre otros. reguladas en la Orden INT/314/2011, de 1 de febrero, sobre empresas de seguridad privada. Además, en cuanto al transporte de efectivo se realiza en furgones blindados, cuyas características deben ajustarse

a lo establecido en la ley mencionada anteriormente, y en el horario de entrega y recepción de efectivo establecido, de 8:00 a 17:30 horas.

#### 4.2. Descripción del problema.

Como ya se ha comentado anteriormente, nos vamos a centrar en la distribución del dinero en efectivo desde que el camión sale del centro operativo de la Región de Murcia gestionado por Iberpay, hasta las diferentes entidades de crédito de esta región, regresando, una vez terminada la ruta, al centro operativo inicial. Se ha supuesto para este trabajo que el traslado del dinero se realiza de esta forma porque es lo que ocurre de media de acuerdo con los datos de Iberpay. Por esto, el problema va a consistir en la obtención de la mejor ruta para un único camión que pasa por todas las entidades posibles (se explicará más adelante) y regresando a su punto inicial.

Los datos relativos a este problema se han obtenido del Banco de España y de la empresa Iberpay (2018). Esta empresa tiene, además de su sede en Madrid, otros 46 centros operativos repartidos por toda España como puede apreciarse en la figura 14.

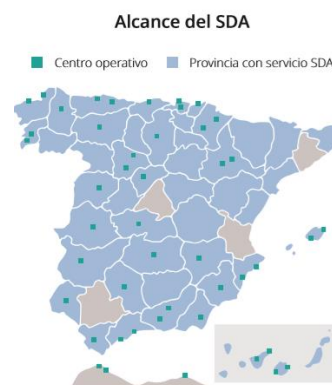


Figura 14. Centros operativos Iberpay (Informe anual Iberpay, 2018)

El proceso de distribución de efectivo es largo, por lo que se ha decidido estudiar solo la parte del problema correspondiente a su distribución desde el centro operativo de la Región de Murcia para su posterior distribución a las entidades de crédito de la comunidad autónoma. Debido a motivos de seguridad la empresa, con la que hemos intentado contactar, ha decidido no proporcionarnos la ubicación exacta de este centro operativo, aunque si sabemos que se encuentra en Murcia capital, razón por la cual se ha tomado un punto al azar de la ciudad de Murcia como la posible ubicación del centro operativo de Iberpay en la región. Este hecho no supone ninguna disminución en la validez de los resultados que se proporcionan, ya que lo que

se pretende con este trabajo es mostrar la utilidad de la aplicación de las técnicas TSP a la operativa habitual de Iberpay y no a un caso concreto y determinado de dicha operativa.

Según los datos anuales de Iberpay, a 31 de diciembre de 2018 había 46 centros operativos en España. Además, este informe muestra el número de operaciones realizadas por cada uno de estos centros, contemplando que en cada operación se considera la entrega o retirada de dinero de una entidad bancaria. El número de operaciones de distribución y retirada de billetes, es 1.814 y 2.218 operaciones, respectivamente (Informe Anual Iberpay, 2018). Al sumar las operaciones realizadas y dividiéndolo entre el número de días laborables de 2018, que fue de 249 días, se obtiene que el número medio de operaciones realizadas diariamente es de 16, por lo que consideraremos que distribuye efectivo a 8 entidades y recoge efectivo de otras 8. A pesar de que los datos se han tomado del año 2018 por no disponerse de información más reciente, de nuevo este hecho no resta generalidad al trabajo que, como ya se ha comentado, su objetivo es mostrar la aplicabilidad a la práctica diaria de la empresa y no determinar la solución de una instancia concreta.

Además, según el número de billetes que entregan o recogen en cada operación (20.263 billetes) (Informe Anual Iberpay, 2018), y, conforme a las normas del Banco de España y la Orden INT/314/2011, de 1 de febrero, sobre empresas de seguridad privada sobre el modo de trasladar el dinero y sobre el tamaño y características del furgón donde se transporta, respectivamente, llegamos a la conclusión de que se pueden realizar estas 16 operaciones diarias de distribución y entrega de dinero sin ningún problema. Esto es así porque, para su distribución, los billetes se deben agrupar en centenares (100 billetes), en paquetes de 10 centenares, y, por último, en bolsas de 10 paquetes (cada bolsa contiene un total de 10.000 billetes). Contando, además, con que las dimensiones de la bolsa son de 82 cm de largo por 52 cm de ancho y que se distribuye efectivo a 8 entidades y se recoge de otras 8, se tendrían que introducir alrededor de 16 bolsas de billetes en el furgón, por lo que en el espacio del furgón reservado para el dinero resultaría suficiente.

En la Región de Murcia, según los datos facilitados por la página web Iahorro (Iahorro, 2020), hay un total de 516 oficinas bancarias entre las distintas localidades de la región. Se ha escogido al azar 16 oficinas bancarias de las 516 existentes para la entrega y recogida de efectivo. He utilizado la herramienta Excel para esta elección al azar, poniendo en una columna los números de 1 al 516 y, a continuación, una columna con valores aleatorios creados con la fórmula =ALEATORIO(), para finalmente ordenar en función de los valores aleatorios creados y elegir las sucursales correspondientes a los 16 mayores valores de la columna de valores

aleatorios. Para más claridad, a las entidades las llamaremos por el nombre de la ciudad y un número, en función de si se repite la ciudad y por orden alfabético de la calle. La ubicación de estas entidades se detalla en la tabla 1.

*Tabla 1. Ubicación entidades bancarias*

<b>Ciudad</b>	<b>Oficina bancaria</b>	<b>Dirección</b>	<b>Código postal</b>
Dep. Aux. Iberpay	Murcia Capital	Murcia (ub. Google maps)	
Águilas 1	Bankia 3	C/ Rey Carlos III, 8	30880
Águilas 2	CaixaBank 3	C/ Iberia, 1	30880
Águilas 3	Santander 2	Plaza de España, 10	30880
Alhama de Murcia	CaixaBank 2	Calle Feria, 23	30840
Cartagena 1	CaixaBank 1	Alameda de San Antón, 37	30205
Cartagena 2	Kutxabank 3	Juan Fernandez, 8	30201
Ceutí	CaixaBank 1	Plaza José Virgili, 5	30562
Lorca	CaixaBank 11	C/ Ortega Melgares, 132	30800
Molina de Segura	CaixaBank 1	Av. Gutierrez Mellado, 48	30500
Murcia 1	Bankia 39	C/ Mayor, 103	30100
Murcia 2	BBVA 6	C/ Floridablanca, 53	30002
Murcia 3	CaixaBank 38	C/ Trapería, 14	30001
Murcia 4	CaixaBank 41	C/ Trapería, 19	30001
Murcia 5	Sabadell 17	AV. Marina Española, S/N (oficina 1243)	30008
San Javier	Bankia 2	Plaza de España, 15	30730
Santomera	CaixaBank 2	C/ Maestro Puig Valera, 65	30140

Mediante Google Maps se han calculado una a una las distancias entre las entidades de crédito entre sí y también con el centro operativo, lo que nos ha permitido obtener la matriz de distancias (kilómetros) (tabla 2) con la que obtener un tour óptimo del TSP. Se ha calculado así porque es una forma bastante rápida, sencilla y actualizada. Entre dos puntos no hay una única distancia, es decir, se podría ir en línea recta utilizando la menor distancia posible, sin embargo, para los trayectos en coche esto no sirve, pues tienen que circular por carreteras y la distancia es diferente. Google Maps da varias opciones de rutas por carretera para elegir la que más nos convenga. En este caso he escogido la que recorre menos kilómetros, sin observar si es de peaje o no, pues el objetivo es minimizar los kilómetros a recorrer. Hay que tener en cuenta que, como se ha comentado, la ruta es por carretera, por lo que depende de estas, es decir, no todas las carreteras son de doble sentido o en línea recta, sobre todo en la ciudad, por lo que esta matriz de distancias no es simétrica. Para poner un ejemplo, se puede observar que de Águilas 2 a Águilas 1 hay 600 metros de distancia, mientras que de Águilas 1 a Águilas 2 hay 1 kilómetro, pues en este caso juega un gran papel las carreteras dentro de la ciudad.

Tabla 2. Distancia en kilómetros entre las entidades de crédito

Desde / Hasta	Dep. Aux. Iberpay	Águilas 1	Águilas 2	Águilas 3	Alhama de Murcia	Cartagena 1	Cartagena 2	Ceuti	Lorca	Molina de Segura	Murcia 1	Murcia 2	Murcia 3	Murcia 4	Murcia 5	San Javier	Santomera
Dep. Aux. Iberpay		110	109	110	35,8	52,3	55	23,9	68,7	16,3	5,9	1,8	1,7	1,9	1,6	46,1	12,5
Águilas 1	100		1	0,063	67,8	71,2	72,5	102	38,2	100	103	101	111	112	102	100	110
Águilas 2	109	0,6		0,65	72,9	76,3	77,6	111	38,3	109	112	110	117	117	107	105	115
Águilas 3	100	0,75	0,95		67,7	71,2	72,5	102	38,1	100	103	101	111	112	102	100	110
Alhama de Murcia	35,3	72,3	72,1	72,4		59,2	61,9	37,4	35,2	35,3	41,4	36,1	38,1	38,3	36,8	68,3	45,4
Cartagena 1	55,6	76,3	76,1	76,3	60,8		1,3	74,4	83,9	66,8	54,3	50,5	52,2	52,4	54,7	30,6	68,6
Cartagena 2	54,4	77	76,8	77	62,2	1,4		75,8	90,9	68,2	55,7	51,9	53,5	53,8	56	29,5	70
Ceuti	23,5	108	107	108	37,4	72,6	75,3		70,3	8,7	24,3	24,4	26,3	26,6	21,3	66,4	27,9
Lorca	67,9	37,8	37,6	37,9	35,7	87,3	90	70,1		68	71,2	68,8	70,8	71	69,5	101	78,1
Molina de Segura	16,4	113	109	114	43,2	65,4	68,2	8,1	67,9		18,4	17,2	19,2	19,4	14,1	59,3	20,7
Murcia 1	5,3	120	120	120	43,3	56,4	59,1	24,9	74,2	17,3		7,4	5,9	6,2	4,5	40,9	8,6
Murcia 2	3,4	113	113	114	36,6	49,4	52,2	24,8	69,9	17,2	7,4		1,4	1,6	4,4	43,3	19
Murcia 3	3,8	107	115	116	37	51,5	54,2	25,2	69,9	17,6	7,5	1,5		0,3	4,1	45,3	19,4
Murcia 4	3,5	115	107	115	36,7	51,2	53,9	24,9	69,6	17,2	7,2	1,2	1,3		3,8	45	19
Murcia 5	1,6	108	111	111	37,4	53,9	56,6	21,3	70,3	13,7	5,1	4,1	2,2	2,4		47,7	11,7
San Javier	45,3	104	104	104	68,5	30,7	29,8	66,7	101	59	40,5	42,7	44,4	44,6	46,9		40,3
Santomera	12,4	119	119	119	45,2	67,5	70,2	27,3	78,1	19,7	8	19,3	13	13,2	11,6	40,8	

### 4.3. Programa WinQSB.

Si se intentara resolver el problema directamente a partir de la formulación lineal correspondiente a esta situación (ver Anexo) se podría comprobar cómo la extensión tanto de la función objetivo como de las restricciones sería abrumadora, razón por la cual emplearemos el software WinQSB, ya conocido por los estudiantes de Investigación Operativa de la titulación.

WinQSB es un sistema interactivo que nos permite resolver distintos tipos problemas de investigación operativa gracias a las múltiples herramientas que tiene. Está compuesto por una serie de módulos para diferentes áreas de la Investigación Operativa, de los cuales destacaremos los siguientes:

- Linear programming (LP) and integer linear programming (ILP): con este módulo se puede resolver el problema de programación lineal gráficamente o mediante el método del Simplex. Utiliza, además, el método de Ramificación y Acotación (Branch and Bound) para resolver problemas de programación lineal entera.
- Linear goal programming (GP) and integer lineal goal programming (IGP): resuelve problemas de programación multiobjetivo con restricciones lineales.
- Quadratic programming (QP) and integer quadratic programming (IQP): resuelve modelos de función objetivo cuadrática y restricciones lineales. Utiliza el método del Simplex adaptado, y el de Ramificación y Acotación para los IQP.
- Network modeling (NET): resuelve el problema del transbordo, del transporte, el de asignación, del camino más corto, árbol generados, flujo máximo, y el problema del agente viajero.
- Nonlinear programming (NLP): resuelve problemas no lineales irrestringidos y con restricciones.

- PERT/CPM: resuelve problemas en los que se tienen que realizar varias tareas en cada proyecto con relaciones de precedencia.

#### **4.4. Solución del problema.**

Para solucionar este problema se va a utilizar dentro del programa WinQSB el módulo Network modeling, pues como se ha comentado, es el que resuelve el problema del viajante de comercio, además de resolver otros problemas comentados anteriormente. Una vez seleccionado el tipo de problema que queremos resolver, hay que elegir el criterio objetivo, es decir, si la función es a maximizar o minimizar. Después, el formato en el que queremos introducir los datos: en forma de matriz, en forma de modelo gráfico o si se trata de una matriz en la que los datos son simétricos. Por último, hay que establecer el número de nodos que tiene el problema y existe la opción de poner un título a este. Una vez hecho esto el programa conduce a la ventana de introducción de los datos correspondientes a la matriz de distancias.

Una vez dentro del programa, en el módulo Network modeling, hay que especificar qué problema queremos resolver, en nuestro caso, queremos minimizar el Travelling Salesman Problem con los datos en forma de matriz (17 nodos), pero con datos no simétricos. Este módulo utiliza como métodos de resolución exactos el Branch and Bound, y como métodos heurísticos puede emplear el Heurístico del Vecino más Próximo, Heurístico de Inserción más Barata o el Heurístico de Mejora de Dos Intercambios. En nuestro caso hemos decidido resolver el problema con todos los métodos posibles para observar las diferencias entre el método exacto (Branch and Bound) y los algoritmos heurísticos. Para ello, una vez introducidos los datos, seleccionaremos la opción resolver el problema, escogiendo el método correspondiente y resolviendo a continuación. La primera solución obtenida es la siguiente:

#### 4.4.1. Branch and Bound

11-30-2020	From Node	Connect To	Distance/Cost		From Node	Connect To	Distance/Cost
1	Iberpay	Murcia 5	1,6	10	Alhama de	Lorca	35,2
2	Murcia 5	Murcia 3	2,2	11	Lorca	Águilas 2	37,6
3	Murcia 3	Murcia 4	0,3	12	Águilas 2	Águilas 1	0,6
4	Murcia 4	Murcia 2	1,2	13	Águilas 1	Águilas 3	0,063
5	Murcia 2	Murcia 1	7,4	14	Águilas 3	Cartagena 1	71,2
6	Murcia 1	Santomera	8,6	15	Cartagena 1	Cartagena 2	1,3
7	Santomera	Molina de	19,7	16	Cartagena 2	San Javier	29,5
8	Molina de	Ceutí	8,1	17	San Javier	Iberpay	45,3
9	Ceutí	Alhama de	37,4				
	<b>Total</b>	<b>Minimal</b>	<b>Traveling</b>	<b>Distance</b>	<b>or Cost</b>	<b>=</b>	<b>307,26</b>
	<b>(Result</b>	<b>from</b>	<b>Branch</b>	<b>and</b>	<b>Bound</b>	<b>Method]</b>	

Figura 15. Solución del TSP con el método Branch and Bound

Recordemos que el algoritmo de ramificación y acotación nos proporciona una solución exacta, que podemos ver en la figura 15. La ruta es la que se muestra en la tabla siguiendo el orden de las filas, esto es, empezando en Iberpay, continuando a Murcia 5, después a Murcia 3, y así pasando por los demás nodos hasta regresar a Iberpay. Tiene una longitud total de 307,26 kilómetros, por lo que el furgón recorrería esa distancia. Como se verá a continuación, esta ruta no es la óptima, pues existe una ruta con una longitud menor, por lo que se ha comprobado que el método Branch and Bound en WinQSB no es fiable.

#### 4.4.2. Heurístico del vecino más próximo

11-30-2020	From Node	Connect To	Distance/Cost		From Node	Connect To	Distance/Cost
1	Iberpay	Murcia 5	1,6	10	Alhama de	Lorca	35,2
2	Murcia 5	Murcia 3	2,2	11	Lorca	Águilas 2	37,6
3	Murcia 3	Murcia 4	0,3	12	Águilas 2	Águilas 1	0,6
4	Murcia 4	Murcia 2	1,2	13	Águilas 1	Águilas 3	0,063
5	Murcia 2	Murcia 1	7,4	14	Águilas 3	Cartagena 1	71,2
6	Murcia 1	Santomera	8,6	15	Cartagena 1	Cartagena 2	1,3
7	Santomera	Molina de	19,7	16	Cartagena 2	San Javier	29,5
8	Molina de	Ceutí	8,1	17	San Javier	Iberpay	45,3
9	Ceutí	Alhama de	37,4				
	<b>Total</b>	<b>Minimal</b>	<b>Traveling</b>	<b>Distance</b>	<b>or Cost</b>	<b>=</b>	<b>307,26</b>
	<b>(Result</b>	<b>from</b>	<b>Nearest</b>	<b>Neighbor</b>	<b>Heuristic]</b>		

Figura 16. Solución del TSP con el método Heurístico del vecino más próximo

El Heurístico del Vecino más Próximo nos da exactamente la misma solución que el método Branch and Bound, obteniendo la ruta que se muestra en la figura 16. Este método nos da una solución factible pero no es óptima. Si el método exacto fuese fiable (véase apartado 4.4.4 para una explicación de esta duda), podríamos afirmar que esta solución es la óptima, sin embargo, si solo hubiésemos resuelto el problema con este método no podríamos confirmar si esta ruta es la óptima o simplemente una más de las muchas factibles.

#### 4.4.3. Heurístico de inserción más barata

11-30-2020	From Node	Connect To	Distance/Cost		From Node	Connect To	Distance/Cost
1	Águilas 1	Águilas 3	0,063	10	Santomera	Molina de	19,7
2	Águilas 3	Cartagena 1	71,2	11	Molina de	Ceutí	8,1
3	Cartagena 1	Cartagena 2	1,3	12	Ceutí	Murcia 5	21,3
4	Cartagena 2	San Javier	29,5	13	Murcia 5	Iberpay	1,6
5	San Javier	Murcia 2	42,7	14	Iberpay	Alhama de	35,8
6	Murcia 2	Murcia 3	1,4	15	Alhama de	Lorca	35,2
7	Murcia 3	Murcia 4	0,3	16	Lorca	Águilas 2	37,6
8	Murcia 4	Murcia 1	7,2	17	Águilas 2	Águilas 1	0,6
9	Murcia 1	Santomera	8,6				
	<b>Total</b>	<b>Minimal</b>	<b>Traveling</b>	<b>Distance</b>	<b>or Cost</b>	<b>=</b>	<b>322,16</b>
	<b>(Result</b>	<b>from</b>	<b>Cheapest</b>	<b>Insertion</b>	<b>Heuristic)</b>		

Figura 17. Solución del TSP con el método Heurístico de inserción más barata

La solución que nos da el Heurístico de Inserción más barata tiene la particularidad de que el primer trayecto que muestra no parte de Iberpay como era de esperar. Esto no supone ningún problema ya que al ser la solución mostrada un ciclo hamiltoniano, todos los nodos pueden considerarse el punto de partida. Por tanto, la interpretación de esta solución es sencilla: la ruta comienza en la fila número 14, donde se parte de Iberpay, y continuaría de forma secuencial hasta la fila número 13, donde se regresa de nuevo a Iberpay. Se observa que el coste es más alto que el obtenido con los métodos de resolución anteriores, 322,16 kilómetros, por lo que es una solución aceptable pero no óptima.



#### 4.4.4. Heurístico 2 - intercambio

11-30-2020	From Node	Connect To	Distance/Cost		From Node	Connect To	Distance/Cost
1	Murcia 2	Iberpay	3,4	10	Águilas 3	Cartagena 1	71,2
2	Iberpay	Murcia 5	1,6	11	Cartagena 1	Cartagena 2	1,3
3	Murcia 5	Molina de	13,7	12	Cartagena 2	San Javier	29,5
4	Molina de	Ceutí	8,1	13	San Javier	Santomera	40,3
5	Ceutí	Alhama de	37,4	14	Santomera	Murcia 1	8
6	Alhama de	Lorca	35,2	15	Murcia 1	Murcia 3	5,9
7	Lorca	Águilas 2	37,6	16	Murcia 3	Murcia 4	0,3
8	Águilas 2	Águilas 1	0,6	17	Murcia 4	Murcia 2	1,2
9	Águilas 1	Águilas 3	0,063				
	Total	Minimal	Traveling	Distance	or Cost	=	295,36
	(Result	from	Two-way	Exchange	Improvement	Heuristic)	

Figura 18. Solución del TSP con el método Heurístico 2 - intercambio

La solución que nos proporciona el Heurístico 2 – intercambio carece de sentido lógico al tener un valor objetivo mejor que la supuesta solución óptima del problema, obtenida mediante el algoritmo de ramificación y acotación. Este contrasentido nos ha hecho barajar la posibilidad de que se debiera a errores propios, pero tras haber realizado numerosas revisiones, algunas de ellas independientes entre sí, hemos comprobado que no hay error por nuestra parte y, efectivamente, los resultados proporcionados por WinQSB son los mostrados en las figuras anteriores. En nuestra búsqueda de una explicación para esta situación tan contradictoria, hemos incluso probado con otras matrices de distancias en las que se han eliminado los decimales para ver si el fallo provenía de la gestión de estos por parte del software, pero los resultados siguen siendo los mismos: el resultado que proporciona WinQSB con el Heurístico 2-intercambio es (inexplicablemente) mejor que el presenta con el método exacto Branch and Bound. Por tanto, la única explicación para este hecho es que, desgraciadamente, el algoritmo de Branch and Bound para WinQSB contiene algún fallo de programación que le impide alcanzar la solución óptima del problema.

En todo caso, dado que la solución proporcionada por el Heurístico 2-intercambio es válida, la conclusión respecto a la aplicación práctica estudiada es que la solución del Branch and Bound para ella no es óptima y que la que ofrece el Heurístico 2-intercambio no podemos confirmar si es óptima ya que el algoritmo utilizado es eso, heurístico.

## Capítulo 5: Conclusiones

El problema del viajante de comercio (o Travelling Salesman Problem en inglés) consiste en minimizar la distancia recorrida por un viajante de comercio que debe visitar un número  $n$  de ciudades pasando sólo una vez por cada una de ellas y de forma que el punto o ciudad de origen sea el mismo que el de destino. El TSP se ha estudiado a lo largo del tiempo, dando cabida a rutas con distancias cada vez mayores. Sin embargo, hoy en día es un problema que se puede aplicar a muchos ámbitos, no sólo a comerciantes. En este trabajo, el TSP se ha aplicado a la distribución del dinero en efectivo en la Región de Murcia, desde que se recoge en los depósitos auxiliares hasta las distintas entidades de crédito de esta región. Además, se ha estudiado el recorrido que realiza el efectivo con el ciclo del efectivo.

Existen numerosas formas de resolver el TSP, entre las que podemos destacar, como método exacto el Branch and Bound, y como métodos heurísticos el del Vecino más Próximo, el de Inserción más Barata y de 2 – intercambio, aunque existen más métodos de resolución. Se han podido determinar las soluciones obtenidas por WinQSB con estos métodos para la distribución del dinero en efectivo. Si las soluciones hubieran sido correctas se habría comprobado que siempre el método exacto proporcionaría la solución óptima, a pesar de que para instancias muy grandes su tiempo de resolución se vuelve muy grande. Del mismo modo, con los métodos heurísticos se habría visto que proporcionarían una solución aceptable, aunque no siempre es la óptima. Como este problema no era demasiado grande, se ha podido utilizar el método Branch and Bound, obteniendo la supuesta solución óptima para la distribución del dinero en efectivo.

No obstante, hemos comprobado que la solución que ofrece el programa WinQSB para el método Branch and Bound no es fiable, pues la solución proporcionada para el Heurístico 2 – intercambio recorre una longitud menor. Tras realizar una revisión y comprobar, además, con otras matrices de distancias sin decimales, el resultado continuaba siendo el mismo. Por lo tanto, el algoritmo TSP mediante Branch and Bound de WinQSB está mal implementado y no puede ser considerado como la solución óptima. Creemos que se trata de un lamentable pero importante descubrimiento que debería ser conocido por todos los usuarios del software WinQSB, ya que de lo contrario se correría el riesgo de aceptar como óptima una solución que podría no serlo, con las negativas consecuencias que ello tendría.

Finalmente, consideramos que se ha conseguido así el objetivo de este trabajo, que era realizar una aplicación práctica del problema del viajante de comercio para observar su utilidad

en problemas cotidianos, que en este caso ha sido la distribución del dinero en efectivo desde los centros operativos donde se almacena el dinero, hasta las distintas entidades de crédito. Por lo tanto, se ha visto un problema que tiene lugar a diario y que tiene la consideración de problema del viajante de comercio.

## ANEXO

Se ha representado cada ciudad con su inicial y el número que le continúa. De este modo, la distancia entre, por ejemplo, Lorca y Murcia 1, sería LM1, y representaría la arista entre estas dos ciudades. La función objetivo a minimizar será:

$$\begin{aligned} \text{Min } Z = & 110DA1 + 109 DA2 + 110DA3 + 35,8DAL + 52,3DC1 + 55DC2 + 23,9DCE + 68,7DL + 16,3DMO + \\ & 5,9DM1 + 1,8DM2 + 1,7DM3 + 1,9DM4 + 1,6DM5 + 46,1DSJ + 12,5DSA + 100A1D + A1A2 + 0,063A1A3 \\ & + 67,8A1AL + 71,2A2C1 + 72,5A1C2 + 102A1CE + 38,2A1L + 100A1MO + 103A1M1 + 101A1M2 + \\ & 111A1M3 + 112A1M4 + 102A1M5 + 100A1SJ + 110A1SA + 109A2D + 0,6A2A1 + 0,65A2A3 + 72,9A2AL \\ & + 76,3A2C1 + 77,6A2C2 + 111A2CE + 38,3A2L + 109A2MO + 112A2M1 + 110A2M2 + 117A2M3 + \\ & 117A2M4 + 107A2M5 + 105A2SJ + 115A2SA + 100A3D + 0,75A3A1 + 0,95A3A2 + 67,7A3AL + 71,2A3C1 \\ & + 72,5A3C2 + 102A3CE + 38,1A3L + 100A3MO + 103A3M1 + 101A3M2 + 111A3M3 + 112A3M4 + \\ & 102A3M5 + 100A3SJ + 110A3SA + 35,3ALD + 72,3ALA1 + 72,1ALA2 + 72,4ALA3 + 59,2ALC1 + \\ & 61,9ALC2 + 37,4ALCE + 35,2ALL + 35,3ALMO + 41,4ALM1 + 36,1ALM2 + 38,1ALM3 + 38,3ALM4 + \\ & 36,8ALM5 + 68,3ALSJ + 45,4ALSA + 55,6C1D + 76,3C1A1 + 76,1C1A2 + 76,3C1A3 + 60,8C1AL + \\ & 1,3C1C2 + 74,4C1CE + 83,9C1L + 66,8C1MO + 54,3C1M1 + 50,5C1M2 + 52,2C1M3 + 52,4C1M4 + \\ & 54,7C1M5 + 30,6C1SJ + 68,6C1SA + 54,4C2D + 77C2A1 + 76,8C2A2 + 77C2A3 + 62,2C2AL + 1,4C2C1 \\ & + 75,8C2CE + 90,9C2L + 68,2C2MO + 55,7C2M1 + 51,9C2M2 + 53,5C2M3 + 53,8C2M4 + 56C2M5 + \\ & 29,5C2SJ + 70C2SA + 23,5CED + 108CEA1 + 107CEA2 + 108CEA3 + 37,4CEAL + 72,6CEC1 + 75,3CEC2 \\ & + 70,3CEL + 8,7CEMO + 24,3CEM1 + 24,4CEM2 + 26,3CEM3 + 26,6CEM4 + 21,3CEM5 + 66,4CESJ + \\ & 27,9CESA + 67,9LD + 37,8LA1 + 37,6LA2 + 37,9LA3 + 35,7LAL + 87,3LC1 + 90LC2 + 70,1LCE + 68LMO \\ & + 71,2LM1 + 68,8LM2 + 70,8LM3 + 71LM4 + 69,5LM5 + 101LSJ + 78,1LSA + 16,4MOD + 113MOA1 + \\ & 109MOA2 + 114MOA3 + 43,2MOAL + 65,4MOC1 + 68,2MOC2 + 8,1MOCE + 67,9MOL + 18,4MOM1 + \\ & 17,2MOM2 + 19,2MOM3 + 19,4MOM4 + 14,1MOM5 + 59,3MOSJ + 20,7MOSA + 5,3M1D + 120M1A1 + \\ & 120M1A2 + 120M1A3 + 43,3M1AL + 56,4M1C1 + 59,1M1C2 + 24,9M1CE + 74,2M1L + 17,3M1MO + \\ & 7,4M1M2 + 5,9M1M3 + 6,2M1M4 + 4,5M1M5 + 40,9M1SJ + 8,6M1SA + 3,4M2D + 113M2A1 + 113M2A2 \\ & + 114M2A3 + 36,6M2AL + 49,4M2C1 + 52,2M2C2 + 24,8M2CE + 69,9M2L + 17,2M2MO + 7,4M2M1 + \\ & 1,4M2M3 + 1,6M2M4 + 4,4M2M5 + 43,3M2SJ + 19 M2SA + 3,8M3D + 107M3A1 + 115M3A2 + 115M3A3 \\ & + 37M3AL + 51,5M3C1 + 54,2M3C2 + 25,2M3CE + 69,9M3L + 17,6M3MO + 7,5M3M1 + 1,5M3M2 + \\ & 0,3M3M4 + 4,1M3M5 + 45,3M3SJ + 19,4M3SA + 3,5M4D + 115M4A1 + 107M4A2 + 115M4A3 + \\ & 36,7M4AL + 52,1M4C1 + 53,9M4C2 + 24,9M4CE + 69,6M4L + 17,2M4MO + 7,2M4M1 + 1,2M4M2 + \\ & 1,3M4M3 + 3,8M4M5 + 45M4SJ + 19M4SA + 1,6M5D + 108M5A1 + 111M5A2 + 111M5A3 + 37,4M5AL \\ & + 53,9M5C1 + 56,6M5C2 + 21,3M5CE + 70,3M5L + 13,7M5MO + 5,1M5M1 + 4,1M5M2 + 2,2M5M3 + \\ & 2,4M5M4 + 47,7M5SJ + 11,7M5SA + 45,3SJD + 104SJA1 + 104SJA2 + 104SJA3 + 68,5SJAL + 30,7SJC1 \\ & + 29,8SJC2 + 66,7SJCE + 101SJL + 59SJMO + 40,5SJM1 + 42,7SJM2 + 44,4SJM3 + 44,6SJM4 + 46,9SJM5 \\ & + 40,3SJSA + 12,4SAD + 119SAA1 + 119SAA2 + 119SAA3 + 45,2SAAL + 67,5SAC1 + 70,2SAC2 + \end{aligned}$$

$$27,3SACE + 78,1SAL + 19,7SAMO + 8SAM1 + 19,3SAM2 + 13SAM3 + 13,2SAM4 + 11,6SAM5 + 40,8SASJ$$

Esta función está sujeta a una serie de restricciones, las de llegada, de salida y de ruptura de subcircuitos.

### Las restricciones de llegada

$$A1D + A2D + A3D + ALD + C1D + C2D + CED + LD + MOD + M1D + M2D + M3D + M4D + M5D + SJD + SAD = 1$$

$$DA1 + A2A1 + A3A1 + ALA1 + C1A1 + C2A1 + CEA1 + LA1 + MOA1 + M1A1 + M2A1 + M3A1 + M4A1 + M5A1 + SJA1 + SAA1 = 1$$

$$DA2 + A1A2 + A3A2 + ALA2 + C1A2 + C2A2 + CEA2 + LA2 + MOA2 + M1A2 + M2A2 + M3A2 + M4A2 + M5A2 + SJA2 + SAA2 = 1$$

$$DA3 + A1A3 + A2A3 + ALA3 + C1A3 + C2A3 + CEA3 + LA3 + MOA3 + M1A3 + M2A3 + M3A3 + M4A3 + M5A3 + SJA3 + SAA3 = 1$$

$$DAL + A1AL + A2AL + A2AL + A3AL + C1AL + C2AL + CEAL + LAL + MOAL + M1AL + M2AL + M3AL + M4AL + M5AL + SJAL + SAAL = 1$$

$$DC1 + A1C1 + A2C1 + A3C1 + ALC1 + C2C1 + CEC1 + LC1 + MOC1 + M1C1 + M2C1 + M3C1 + M4C1 + M5C1 + SJC1 + SAC1 = 1$$

$$DC2 + A1C2 + A2C2 + A3C2 + ALC2 + C1C2 + CEC2 + LC2 + MOC2 + M1C2 + M2C2 + M3C2 + M4C2 + M5C2 + SJC2 + SAC2 = 1$$

$$DCE + A1CE + A2CE + A3CE + ALCE + C1CE + C2CE + LCE + MOCE + M1CE + M2CE + M3CE + M4CE + M5CE + SJCE + SACE = 1$$

$$DL + A1L + A2L + A3L + ALL + C1L + C2L + CEL + MOL + M1L + M2L + M3L + M4L + M5L + SJL + SAL = 1$$

$$DMO + A1MO + A2MO + A3MO + ALMO + C1MO + C2MO + CEMO + LMO + M1MO + M2MO + M3MO + M4MO + M5MO + SJMO + SAMO = 1$$

$$DM1 + A1M1 + A2M1 + A3M1 + ALM1 + C1M1 + C2M1 + CEM1 + LM1 + MOM1 + M2M1 + M3M1 + M4M1 + M5M1 + SJM1 + SAM1 = 1$$

$$DM2 + A1M2 + A2M2 + A3M2 + ALM2 + C1M2 + C2M2 + CEM2 + LM2 + MOM2 + M1M2 + M3M2 + M4M2 + M5M2 + SJM2 + SAM2 = 1$$

$$DM3 + A1M3 + A2M3 + A3M3 + ALM3 + C1M3 + C2M3 + CEM3 + LM3 + MOM3 + M1M3 + M2M3 + M4M3 + M5M3 + SJM3 + SAM3 = 1$$

$$DM4 + A1M4 + A2M4 + A3M4 + ALM4 + C1M4 + C2M4 + CEM4 + LM4 + MOM4 + M1M4 + M2M4 + M3M4 + M5M4 + SJM4 + SAM4 = 1$$

$$DM5 + A1M5 + A2M5 + A3M5 + ALM5 + C1M5 + C2M5 + CEM5 + LM5 + MOM5 + M1M5 + M2M5 + M3M5 + M4M5 + SJM5 + SAM5 = 1$$

$$DSJ + A1SJ + A2SJ + A3SJ + ALSJ + C1SJ + C2SJ + CESJ + LSJ + MOSJ + M1SJ + M2SJ + M3SJ + M4SJ + M5SJ + SASJ = 1$$

$$DSA + A1SA + A2SA + A3SA + ALSA + C1SA + C2SA + CESA + LSA + MOSA + M1SA + M2SA + M3SA + M4SA + M5SA + SJSA = 1$$

### Restricciones de salida

$$DA1 + DA2 + DA3 + DAL + DC1 + DC2 + DCE + DL + DMO + DM1 + DM2 + DM3 + DM4 + DM5 + DSJ + DSA = 1$$

$$A1D + A1A2 + A1A3 + A1AL + A1C1 + A1C2 + A1CE + A1L + A1MO + A1M1 + A1M2 + A1M3 + A1M4 + A1M5 + A1SJ + A1SA = 1$$

$$A2D + A2A1 + A2A3 + A2AL + A2C1 + A2C2 + A2CE + A2L + A2MO + A2M1 + A2M2 + A2M3 + A2M4 + A2M5 + A2SJ + A2SA = 1$$

$$\begin{aligned}
&A3D + A3A1 + A3A2 + A3AL + A3C1 + A3C2 + A3CE + A3L + A3MO + A3M1 + A3M2 + A3M3 + \\
&A3M4 + A3M5 + A3SJ + A3SA = 1 \\
&ALD + ALA1 + ALA2 + ALA3 + ALC1 + ALC2 + ALCE + ALL + ALMO + ALM1 + ALM2 + ALM3 + \\
&ALM4 + ALM5 + ALSJ + ALSA = 1 \\
&C1D + C1A1 + C1A2 + C1A3 + C1AL + C1C2 + C1CE + C1L + C1MO + C1M1 + C1M2 + C1M3 + C1M4 \\
&+ C1M5 + C1SJ + C1SA = 1 \\
&C2D + C2A1 + C2A2 + C2A3 + C2AL + C2C1 + C2CE + C2L + C2MO + C2M1 + C2M2 + C2M3 + C2M4 \\
&+ C2M5 + C2SJ + C2SA = 1 \\
&CED + CEA1 + CEA2 + CEA3 + CEAL + CEC1 + CEC2 + CEL + CEMO + CEM1 + CEM2 + CEM3 + \\
&CEM4 + CEM5 + CESJ + CESA = 1 \\
&LD + LA1 + LA2 + LA3 + LAL + LC1 + LC2 + LCE + LMO + LM1 + LM2 + LM3 + LM4 + LM5 + LSJ + \\
&LSA = 1 \\
&MOD + MOA1 + MOA2 + MOA3 + MOAL + MOC1 + MOC2 + MOCE + MOL + MOM1 + MOM2 + \\
&MOM3 + MOM4 + MOM5 + MOSJ + MOSA = 1 \\
&M1D + M1A1 + M1A2 + M1A3 + M1AL + M1C1 + M1C2 + M1CE + M1L + M1MO + M1M2 + M1M3 + \\
&M1M4 + M1M5 + M1SJ + M1SA = 1 \\
&M2D + M2A1 + M2A2 + M2A3 + M2AL + M2C1 + M2C2 + M2CE + M2L + M2MO + M2M1 + M2M3 + \\
&M2M4 + M2M5 + M2SJ + M2SA = 1 \\
&M3D + M3A1 + M3A2 + M3A3 + M3AL + M3C1 + M3C2 + M3CE + M3L + M3MO + M3M1 + M3M2 + \\
&M3M4 + M3M5 + M3SJ + M3SA = 1 \\
&M4D + M4A1 + M4A2 + M4A3 + M4AL + M4C1 + M4C2 + M4CE + M4L + M4MO + M4M1 + M4M2 + \\
&M4M3 + M4M5 + M4SJ + M4SA = 1 \\
&M5D + M5A1 + M5A2 + M5A3 + M5AL + M5C1 + M5C2 + M5CE + M5L + M5MO + M5M1 + M5M2 + \\
&M5M3 + M5M4 + M5SJ + M5SA = 1 \\
&SJD + SJA1 + SJA2 + SJA3 + SJAL + SJC1 + SJC2 + SJCE + SJL + SJMO + SJM1 + SJM2 + SJM3 + SJM4 \\
&+ SJM5 + SJSA = 1 \\
&SAD + SAA1 + SAA2 + SAA3 + SAAL + SAC1 + SAC2 + SACE + SAL + SAMO + SAM1 + SAM2 + \\
&SAM3 + SAM4 + SAM5 + SASJ = 1
\end{aligned}$$

### Restricciones de ruptura de subcircuitos

Estas restricciones son muchas y muy complejas y, puesto que no se precisan para encontrar la solución mediante el software WinQSB, ni son el objetivo de este trabajo, no se han incluido.

## BIBLIOGRAFÍA

Allsager M., Ali Othman Z., Ayob M., (2017), *Cheapest Insertion Constructive Heuristic based on Two Combination Seed Customer Criterion for the Capacitated Vehicle Routing Problem*, Centre of artificial intelligence, Faculty of Information Sciences and Technology, University Kebangsaan Malaysia.

Applegate, D., Bixby, R., Chvátal, V., Cook, W. (2006), *The Travelling Salesman Problem: A Computational Study*. Princeton: Princeton University Press. 593 pp

Banco de España (2005), *Los Depósitos Auxiliares del Banco de España*, Boletín Económico.

Banco de España, Eurosistema [en línea] [consulta: 7 septiembre 2020]. Disponible en: [https://www.bde.es/bde/es/areas/billemone/Publico\\_general/El\\_ciclo\\_del\\_efe/El\\_ciclo\\_del\\_efectivo.html](https://www.bde.es/bde/es/areas/billemone/Publico_general/El_ciclo_del_efe/El_ciclo_del_efectivo.html)

Biggs, N., Lloyd, K., Wilson, R. (1986), *Graph Theory, 1736-1936*. Clarendon Press. 239 pp.

Calviño Martínez, A. (2011), *Cooperación en los problemas del viajante (TSP) y de rutas de vehículos (VRP): una panorámica*. Máster Interuniversitario en Técnicas Estadísticas.

Campos Aucejo, V. (2020), *Problema de rutas*, Universidad de Valencia.

Dantzig, G., Fulkerson, R., Johnson, S. (1954) *Solution of a large-scale traveling-salesman problem*. Operations Research 2, 393-410.

Euler, L. (1736). *Solutio Problematicis ad Geometrica Situs Pertinentis*. Commentarii academiae scientiarum Petropolitanae 8, 128-140

Garey, M., y Jonson., D. (1979), *Computers and Intractability: A Guide to the Theory of NPCompleteness*. Freeman, San Francisco, CA.

*Gestión de Operaciones, Ejemplo del Algoritmo de Branch and Bound (Ramificación y Acotamiento)* [en línea] [consulta: 12 junio 2020]. Disponible en: <https://www.gestiondeoperaciones.net/programacion-entera/ejemplo-del-algoritmo-de-branch-and-bound-ramificacion-y-acotamiento/>

*Gestión de Operaciones, Ejemplo del Algoritmo de Branch and Bound* [en línea] [consulta: 14 septiembre 2020]. Disponible en: <https://www.gestiondeoperaciones.net/programacion-entera/ejemplo-del-algoritmo-de-branch-and-bound-ramificacion-y-acotamiento/>.

Gomory, R. E. (1958), *Outline of an Algorithm for Integer Solutions to Linear Programs*, Bulletin of the American Mathematical Society 64, 275-278.

Iahorro, *Oficinas Murcia* [en línea] [consulta: 28 agosto 2020]. Disponible en: <https://www.iahorro.com/oficinas/murcia/>

Iberpay, *Efectivo* [en línea] [consulta: 8 septiembre 2020]. Disponible en: <https://www.iberpay.es/Secciones/03SDA/DefaultSDA.aspx>

Iberpay, *Informes Anuales* [en línea] [consulta: 1 junio 2020]. Disponible en: <https://www.iberpay.es/Secciones/01Iberpay/Paginas/InformacionCorporativa/InformacionFinanciera.aspx>

Infantes Durán, M. (2018), *El problema del viajante (TSP)*, Universidad de Sevilla.

Land, A., Doig, A. (1960), *An automatic method of solving discrete programming problems*. *Econometrica* 28.

Lawler, E.L, Lenstra, J.K., Rinnooy Kan, A.H.G. Shmoys, D.B. (1984) *The traveling salesman problem*. John Wiley & Sons.

Lin, S., Kernighan, B. (1963), *An effective heuristic algorithm for the traveling salesman problem*. *Operations Research*, 21.

Martí Cunqueiro, R. (2003), *Algoritmos heurísticos en optimización combinatoria*. Departamento de Estadística e Investigación Operativa, Universidad de Valencia.

*Matemáticas Discretas, Ciclo de Hamilton*, [en línea] [consulta: 10 julio 2020] Disponible en: <https://m4tem4ticasdiscret4s.blogspot.com/2016/11/ciclo-de-hamilton.html>

Mendoza Casanova J.J., (2017), *Travelling Salesman Problem (TSP) Diseño de Algoritmos Heurísticos y Metaheurísticos eficientes para resolver el Problema del Agente Viajero*, Facultad Regional Multidisciplinaria de Chontales Farem- Chontales, Universidad Nacional Autónoma de Nicaragua, Managua.

Menger, K. (1931). *Bericht u̇ber ein mathematisches Kolloquium*. *Monatshefte fu̇r Mathematik und Physik* 38, 17-38.

Miller, C.E., Tucker, A.W. and Zemlin, R.A. (1960) *Integer Programming Formulations and Traveling Salesman Problems*. *Journal of the Association for Computing Machinery*.

Orden INT/314/2011, de 1 de febrero, sobre empresas de seguridad privada.

Papadimitriou, C., y Steiglitz, K. (1998), *Combinatorial Optimization: Algorithms and Complexity*. General Publishing Company.

Peñalva García, S. (2015), *El problema del viajante. Métodos de resolución y un enfoque hacia la Teoría de la Computación*, Universidad de la Rioja.



Pérez de Vargas Moreno, B. (2015), *Resolución del Problema del Viajante de Comercio (TSP) y su variante con Ventanas de Tiempo (TSPTW) usando métodos heurísticos de búsqueda local*, Universidad de Valladolid.

Ramos, S. A. (2007), *Modelos y Optimización I, Heurísticas y Problemas Combinatorios*.

Reeves, C.R. (1993), *Modern Heuristic Techniques for Combinatorial Problems*. Blackwell Scientific Publications.

Sensagent Diccionario [en línea] [consulta: 21 mayo 2020]. Disponible en: <http://diccionario.sensagent.com/Camino%20hamiltoniano/es-es/>

*The operational Research Society, A brief history of the travelling salesman problem* [en línea] [consulta: 29 abril 2020]. Disponible en: <https://www.theorsociety.com/about-or/or-methods/heuristics/a-brief-history-of-the-travelling-salesman-problem/>

Veiga Losada, F. J. (2013), *El problema del viajante*, Universidad de Santiago de Compostela.