

Proyecto Fin de Máster Ingeniería Industrial

El problema del viajante aplicado a la red logística
de Amazon en Sevilla.

Autor: Javier Rojas Eugui

Tutor: Manuel Ordóñez Sánchez

Dpto. Matemática Aplicada II
Escuela Técnica Superior de Ingeniería
Universidad de Sevilla

Sevilla, 2020



Proyecto Fin de Máster
Ingeniería Industrial

El problema del viajante aplicado a la red logística de Amazon en Sevilla.

Autor:

Javier Rojas Eugui

Tutor:

Manuel Ordóñez Sánchez

Profesor Titular de Universidad

Dpto. de Matemática Aplicada II

Escuela Técnica Superior de Ingeniería

Universidad de Sevilla

Sevilla, 2020

Proyecto Fin de Máster: El problema del viajante aplicado a la red logística de Amazon en
Sevilla.

Autor: Javier Rojas Eugui

Tutor: Manuel Ordóñez Sánchez

El tribunal nombrado para juzgar el Proyecto arriba indicado, compuesto por los siguientes miembros:

Presidente:

Vocales:

Secretario:

Acuerdan otorgarle la calificación de:

Sevilla, 2020

El Secretario del Tribunal

*A mis abuelos, que ya descansan
en la casa del Padre.
AMDG.*

Resumen

El objeto de este trabajo es la aplicación del problema del viajante a la red logística de la zona oeste de Sevilla, mediante el empleo de tres algoritmos: Ramificación y Poda, Programación Dinámica y el algoritmo de Christofides. A través de éstos se optimizará el caso en el que un vehículo proveniente del centro logístico de Amazon en Dos Hermanas tenga que servir a todos los pueblos de la zona oeste de la capital hispalense, una única vez. Respondiendo, así, a la necesidad de reducción de costes de esta multinacional del comercio electrónico en una zona de Sevilla congestionada por el tráfico.

Abstract

The purpose of this Master Final Project is the application of the traveling salesman problem to the logistics network in the west of Seville, using three algorithms: Branch and Bound, Dynamic Programming, and the Christofides' algorithm. These will optimize the case in which a vehicle coming from the Amazon fulfillment center in Dos Hermanas has to serve all the towns in the western area of the Sevillian capital. In this way is meeting the need for cost reduction of this multinational e-commerce in an area of Seville congested by traffic.

Índice

| | |
|--|-------------|
| Resumen | IX |
| Abstract | XI |
| Índice | XIII |
| Índice de Figuras | XVII |
| Índice de Tablas | XIX |
| Notación | XXI |
| 1 Amazon, el gran “minorista” global | 1 |
| 1.1 <i>Historia</i> | 2 |
| 1.1.1 Jeff Bezos, la mente detrás de Amazon | 2 |
| 1.1.2 Amazon, su historia | 3 |
| 1.1.3 Actualidad, datos y estadísticas | 4 |
| 1.2 <i>Organización y jerarquía</i> | 10 |
| 1.2.1 Cultura de la organización | 11 |
| 1.3 <i>Modelo de negocio</i> | 12 |
| 1.3.1 Propuesta de valor | 12 |
| 1.3.2 Fuente de ingresos | 12 |
| 1.3.3 Estructura de los costes | 13 |
| 1.3.4 Actividades, socios y recursos | 14 |
| 1.3.5 Canales, segmentación y relación con los clientes. | 14 |
| 1.4 <i>Logística y Operaciones de Amazon</i> | 16 |
| 1.4.1 Objetivos - The Bezos Virtuous Cycle | 16 |
| 1.4.2 Funcionamiento | 17 |
| 1.4.3 Tipos de instalaciones | 18 |
| 1.5 <i>Amazon en Sevilla y la problemática</i> | 20 |

| | | |
|----------|--|-----------|
| 2 | -El problema del viajante, estado del arte | 27 |
| 2.1 | <i>Aplicaciones reales</i> | 29 |
| 2.1.1 | Ruta de vehículos | 29 |
| 2.1.2 | Optimización de circuitos impresos de PCB | 29 |
| 2.1.3 | Minimizar gasto del empapelado de una pared | 29 |
| 2.1.4 | Secuenciación de trabajos | 30 |
| 2.1.5 | Reducción de la dispersión de pedidos en la industria del vidrio | 30 |
| 2.1.6 | Generación de listas de reproducción según similitudes | 33 |
| 2.2 | <i>Variantes del TSP</i> | 34 |
| 2.2.1 | Minimizar el camino hamiltoniano | 34 |
| 2.2.2 | TSP con ciudades repetidas | 35 |
| 2.2.3 | El cuello de botella TSP | 36 |
| 2.2.4 | TSP dependiente del tiempo | 36 |
| 2.2.5 | TSP basados en el beneficio | 38 |
| 2.3 | <i>Problemas genéricos relacionados con el TSP</i> | 41 |
| 2.3.3 | El problema de la asignación cuadrática | 43 |
| 2.3.4 | El problema del camino más largo | 44 |
| 2.3.5 | Árbol de cobertura mínimo | 45 |
| 2.3.6 | El problema del cartero chino | 46 |
| 2.4 | <i>Algoritmos brutos y complejidad del TSP</i> | 49 |
| 2.4.1 | Algoritmo de fuerza bruta | 49 |
| 2.4.2 | Algoritmo del camino mínimo | 50 |
| 2.4.3 | El TSP pequeño | 51 |
| 2.4.4 | Orden de complejidad de un algoritmo | 56 |
| 3 | Algoritmos escogidos | 59 |
| 3.1 | <i>Algoritmo de Branch and Bound</i> | 60 |
| 3.1.1 | Relajación y cálculo de la cota mínima o <i>bound</i> | 60 |
| 3.1.2 | Proceso de ramificación | 62 |
| 3.1.3 | Cálculo de la cota mínima para ramas de niveles superiores | 63 |
| 3.1.4 | Regla de selección de subproblemas | 64 |
| 3.1.5 | Algoritmo | 64 |
| 3.2 | <i>Algoritmo de Programación Dinámica</i> | 66 |

| | | |
|----------|--|-----------|
| 3.2.1 | Complejidad | 67 |
| 3.3 | <i>Algoritmo Heurístico de Christofides</i> | 69 |
| 4 | Adaptación al problema de Amazon en Sevilla | 73 |
| 4.1 | <i>Nodos</i> | 74 |
| 4.2 | <i>Arcos</i> | 75 |
| 4.3 | <i>Matriz de costes</i> | 75 |
| 5 | Resultado y conclusiones | 77 |
| 5.1 | <i>Resultados del algoritmo de Branch and Bound</i> | 78 |
| 5.1.1 | Caso de 23 nodos | 78 |
| 5.1.2 | Caso de 12 nodos | 79 |
| 5.1.3 | Caso de 16 nodos | 79 |
| 5.2 | <i>Resultados del algoritmo de Programación Dinámica</i> | 80 |
| 5.2.1 | Caso de 23 nodos | 80 |
| 5.2.2 | Caso de 12 nodos | 80 |
| 5.2.3 | Caso de 16 nodos | 80 |
| 5.3 | <i>Resultados del algoritmo de Christofides</i> | 81 |
| 5.3.1 | Caso de 23 nodos | 81 |
| 5.3.2 | Caso de 12 nodos | 82 |
| 5.3.3 | Caso de 16 nodos | 82 |
| 5.4 | <i>Comparativa</i> | 83 |
| 5.5 | <i>Conclusión</i> | 84 |
| 6 | Bibliografía | 86 |
| | Anexo I: Matriz de costes | 92 |

ÍNDICE DE FIGURAS

| | |
|--|----|
| Figura 1: Jeff Bezos, fundador de Amazon..... | 2 |
| Figura 2: Organigrama de Amazon. [8] | 10 |
| Figura 3: Distribución de ingresos de Amazon en 2018. Fuente https://www.bbc.com/mundo/noticias-48909676 | 12 |
| Figura 4: El círculo virtuoso de Bezos. Fuente: Amazon.com | 16 |
| Figura 5: Círculo virtuoso de Bezos desde el punto de vista logístico. Fuente [15]. | 17 |
| Figura 6: Ejemplo de centro de recepción [17]..... | 20 |
| Figura 7: Recreación del centro logístico de Amazon en Dos Hermanas (Sevilla). | 21 |
| Figura 8: Circunvalación exterior de Sevilla. Tramos acabados e inacabados. Fuente: Wikipedia.commons. | 22 |
| Figura 9: Vías de congestión metropolitana [21]..... | 24 |
| Figura 10: Ejemplo típico de un mosaico de corte para una sola lámina en la industria del vidrio. [26]..... | 31 |
| Figura 11: ‘Traveler Sound Player’, regulador de canciones..... | 33 |
| Figura 12: “The Icosian Game”. Fuente: Royal Irish Academy Library en Twitter.com | 34 |
| Figura 13. Pseudocódigo del algoritmo de Floyd-Warshall. Fuente: https://resources.mpi-inf.mpg.de/ | 35 |
| Figura 14: Grafo en el cual el viajante tendrá que visitar el nodo B al menos dos veces. Fuente: http://web.mit.edu/ | 36 |
| Figura 15: <i>Subtours</i> para un problema de 5 ciudades. Fuente: chegg.com..... | 42 |
| Figura 16: Grafo completo. Fuente Wikipedia Commons..... | 45 |
| Figura 17: Ejemplo de mínimo árbol de cobertura. [38]..... | 46 |
| Figura 18: Problema de los siete puentes de Königsberg [40]..... | 47 |

| | |
|--|----|
| Figura 19: Pseudo código del algoritmo de la fuerza bruta..... | 50 |
| Figura 20: Algoritmo que resuelve El problema del camino mínimo. | 51 |
| Figura 21: Diagrama de casos para el Teorema 1. Fuente: Elaboración propia. | 54 |
| Figura 22: Ramificación en profundidad para un problema de 5 nodos. Fuente [42]..... | 62 |
| Figura 23: Pseudo algoritmo de Branch and Bound..... | 65 |
| Figura 24: Pseudo código del algoritmo de Held y Karp para el TSP. Fuente [28]. | 66 |
| Figura 25: Pseudo algoritmo de árbol de menor coste. | 69 |
| Figura 26. Un 1-emparejamiento perfecto para un grafo de 6 nodos. [45]..... | 70 |
| Figura 27: Dado un grafo completo G de 10 nodos se aplica el algoritmo de Christofides para hallar una solución al TSP del grafo G . Fuente [46] | 72 |
| Figura 28: Mapa de la zona oeste de Sevilla. | 74 |
| Figura 29: Resolución del problema de los 23 nodos con el algoritmo de Branch and Bound. | 78 |
| Figura 30: Resolución del problema de los 23 nodos con el algoritmo de Christofides. | 81 |

ÍNDICE DE TABLAS

| | |
|---|----|
| Tabla 1: Identificación de las vías de acceso a Sevilla [21]. | 23 |
| Tabla 2: Principales órdenes de complejidad. Fuente: www.cs.us.es/~jalonso | 57 |
| Tabla 3: Resultados para diferentes instancias del problema. | 83 |
| Tabla 4: Relación del Problema de Christofides con la solución óptima del problema. | 83 |

Notación

| | |
|------------------|---------------------------------|
| $A \cup B$ | La unión de A y B |
| $A \cap B$ | la intersección de A y B |
| \subseteq | Es subconjunto de |
| \in | Pertenece a |
| \notin | No está contenido en |
| $\text{Min}\{\}$ | Mínimo de un conjunto |
| : | Tal que |
| | Tal que |
| \leq | Menor o igual |
| \geq | Mayor o igual |
| \setminus | Menos (diferencia de conjuntos) |
| $ $ | Valor absoluto |
| \exists | Existe |
| \nexists | No existe |

1 AMAZON, EL GRAN “MINORISTA” GLOBAL

“En el mundo antiguo, dedicabas un 30% de tu tiempo en crear un gran servicio y un 70% en difundirlo. En el nuevo tiempo, eso se invierte.”

- Jeff Bezos -

Amazon es una empresa estadounidense dedicada principalmente al comercio electrónico con sede en Seattle, Washington, fundada en 1994 por Jeff Bezos. Actualmente, es la empresa con mayor valor de marca a nivel mundial según el informe global de BrandZ [1] con un valor de 315.505 millones de dólares y con unos beneficios netos en 2019 de 3.268 millones de dólares. Comenzó siendo un vendedor de libros online y ha pasado a vender todo tipo de artículos: “De la A al Z”, lema de la compañía. También, ha llegado a desarrollar su propia gama de productos con marca Amazon entre los que destacan sus libros electrónicos Kindle (propiedad de Amazon), la plataforma de publicación de libros, películas y música CreateSpace; servicios informáticos para todo tipo de empresas, Amazon Web Services, y dispositivos electrónicos que caminan hacia el hogar inteligente, Alexa.

Esta variedad de productos, servicios y empresas que conforman Amazon, evidencian la importancia que debe tener la gestión de la logística y de los productos en la compañía. Por una parte, la logística es el grueso de la empresa, es la base sobre la que se sustenta desde los inicios la compañía y de donde procede el mayor volumen de negocio. Por otra parte, la cartera de productos propios se integra en esta logística y le da una identidad a la propia marca, es decir, pasa de ser un intermediario, a ser una empresa con sus propios productos y servicios que afianzan la importancia de la compañía en la mentalidad de sus clientes.

1.1 Historia

1.1.1 Jeff Bezos, la mente detrás de Amazon

Se podría decir que la historia de Amazon comienza con la propia historia de Jeff Bezos y de las influencias y cualidades que éste ha tenido a lo largo de su vida. Una de las influencias más notorias fue la de su abuelo materno, Lawrence Preston Gise. Trabajó en el DARPA (*Defense Advanced Research Projects Agency*), de donde surgieron las bases ARPANET, bases a su vez de Internet; lo que proporcionó familiaridad a Bezos con la gran herramienta del siglo XXI [2]. Por otra parte, se graduó en Ciencias de la Computación e Ingeniería Eléctrica en la universidad de Princeton en 1986. Allí fue miembro de la “Phi Beta Kappa” y fue elegido para “Tau Beta Pi”, la sociedad de honor en ingeniería más antigua de Estados Unidos. También en Princeton fue presidente de la asociación de alumnos “*Students for the Exploration and Development of Space*” (SEDS), dedicada a la promoción de la exploración espacial y el desarrollo de proyectos de Ingeniería. Lo que, en resumidas cuentas, hace ver que fue un alumno destacado y con gran interés y dedicación por el mundo moderno que se conoce a día de hoy y que, por aquel entonces, se estaba gestando.

Al salir de Princeton trabajó en diferentes compañías: Fitel, empresa emergente de telecomunicaciones; Bankers Trust, una importante banca estadounidense y, finalmente, en D. E. Shaw & Co., gestora de inversiones especialista en desarrollar modelos matemáticos para descubrir anomalías en el mercado, en donde Bezos ascendería rápidamente en muy poco tiempo [3]



Figura 1: Jeff Bezos, fundador de Amazon.

1.1.2 Amazon, su historia

En D. E. Shaw & Co Bezos con tan solo cuatro años en la empresa, ascendió al cargo de vicepresidente y fue el encargado de buscar oportunidades de negocio para la compañía. Fue en esta posición en la que él comenzó a gestar la idea de Amazon; vio que el mundo de Internet estaba abriendo un nuevo territorio donde ampliar la empresa, por lo que pensó en una lista de 20 productos que se podían vender en Internet, ya que el acceso a éste estaba creciendo en un 2300% anualmente, vio la ocasión y decidió dejar D. E. Shaw & Co, para fundar su propia empresa. De esa lista, Jeff decidió vender libros, porque eran universales, porque eran baratos y porque había millones de títulos que se podían vender. Aunque ya había negocios parecidos, se diferenció de la competencia con la promesa de vender cualquier libro a cualquier lugar. Fundó su negocio en el garaje de su casa con el nombre de Cadabra, aunque a finales de 1994 cambió el nombre por Amazon, oficialmente abrió sus puertas el dieciséis de julio de 1995. En menos de 30 días ya generaba unos ingresos de más de 20.000 dólares por semana y en pocos meses ya vendía en cada estado de Estados Unidos y en más de 45 países. Al final del año había contratado cerca de 250 empleados. [4]

En 1995, Bezos recibe una inversión de 8 millones de dólares por parte de Kleiner Perkins, cuyos retornos eran superiores al 55000% para el año siguiente. Un año después Amazon ya tenía suscritos 180.000 clientes y para octubre de 1997, ya contaba con alrededor de 1.000.000. Sus ingresos crecieron de 15.7 millones de dólares en 1996 a 610 millones en 1998. [2] [5]

En 1997, Amazon se hace pública cotizando en bolsa con un precio de 18\$ por acción. Aunque el crecimiento de los beneficios fue inusualmente lento, no consiguió beneficio hasta 2001 con un margen de cinco millones de dólares y con más de mil millones de ingresos.

En 1998 la empresa expandió su negocio y comenzó a vender música y vídeos, al mismo tiempo que adquiría otras empresas como PlanetAll, Jungle y Bookpages.co.uk (más tarde se convertiría en Amazon UK). Para 1999, ya vendía juguetes, videojuegos, software y productos electrónicos entre otros.

Además de expandir su negocio aumentando la variedad de productos que vendía, Bezos decidió adquirir empresas diferentes y aumentar su cartera de servicios. En 2006 nace Amazon Web Services (AWS), una plataforma computacional que ofrecía servicios de almacenamiento en línea, así como otros servicios, que generalmente no iban dirigidos directamente a los usuarios finales, sino que ofrecía a sus clientes funcionalidades que pudieran usar en el desarrollo de sus aplicaciones para así mejorar la calidad de servicio de los consumidores. Los clientes de este servicio eran principalmente pequeñas empresas, ya que les permitía no tener que ser propietarias de sus propios equipos informáticos gracias

a la tecnología desarrollada por Amazon, S3 y EC2. Sin embargo, más adelante, también llegaron a unirse grandes compañías como Netflix, Pinterest, LinkedIn, Facebook o Twitch.

Por otra parte, expandió su negocio a otros puntos y adquirió mayor variedad de empresas tales como, IMDB, Zapoos, Whole Foods Market, entre otros. Es decir, la idea de negocio de Amazon fue cada vez alejándose de ser un simple vendedor de productos y se convirtió en una empresa multifuncional en constante renovación para aumentar sus beneficios.

En 2007, Amazon sacó al mercado su propio libro electrónico bajo la marca Kindle por un precio de 399\$, se agotaron las existencias en menos de cinco horas y media. Fue un éxito en ventas. Más tarde, en 2011 desarrolló una Tablet de bajo coste en la misma gama de productos, llamada Kindle Fire y en 2012 constituían el 50% de las *tablets* que usaban el sistema operativo de Google, Android.

En 2009 Amazon crea su primera editorial: Amazon Publishing, entendida como una plataforma de *self-publishing*, es decir, la publicación de un libro sin pasar por un intermediario. Esto les permite a los escritores ser los responsables directos de todos los procesos involucrados en la edición y venta de un libro. Amazon, les ofrece canal directo con los recursos que les permite este desarrollo. A día de hoy Amazon Publishing, recoge 15 sellos editoriales, hecho que obliga a reconocer a las compañías editoriales a Amazon, más como un competidor que como un vendedor de libros.

1.1.3 Actualidad, datos y estadísticas

1.1.3.1 Ingresos

Los ingresos de Amazon provienen principalmente de la venta de productos a través de su página online tanto en Estados Unidos como en el ámbito internacional. Otra fuente de ingreso son los servicios webs que presta a empresas y desarrolladores, así como, otros servicios de menor índole. En el apartado 1.3.2 se detallarán las fuentes de ingreso de la compañía.

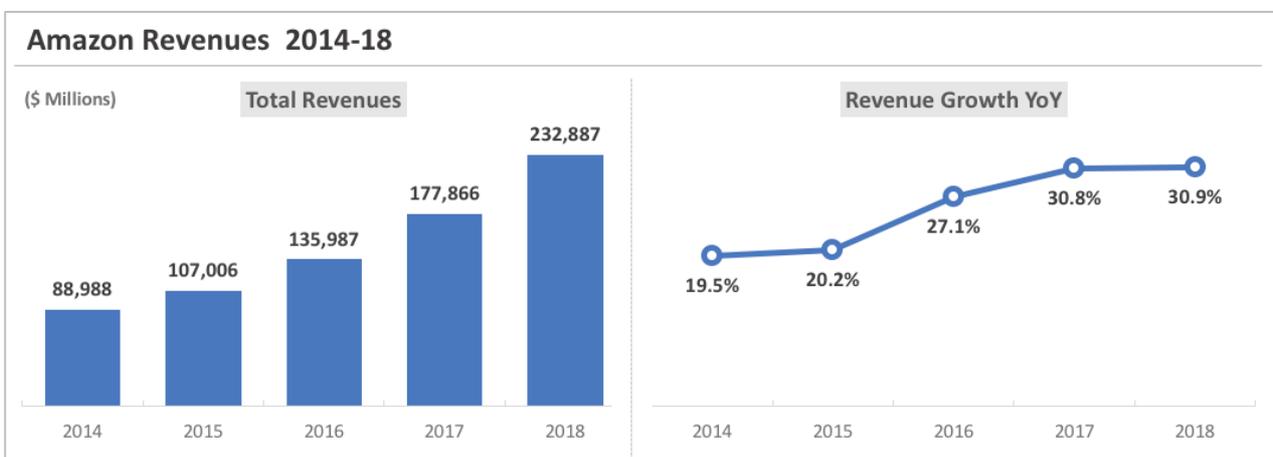


Gráfico 1. Ingresos de Amazon Inc. desde 2014 hasta 2018. Fuente: <https://mgmresearch.com/>

En los últimos años este ingreso ha ido creciendo hasta llegar a los 280.500 millones de dólares en 2019, un 20% más que en 2018 donde se percibió un ingreso de 232.900 millones. En el Gráfico 2 se puede ver un resumen de la evolución del ingreso y su crecimiento en los últimos cinco años. Se comprueba que desde 2014 en adelante los ingresos fueron en un aumento constante y con una aceleración positiva hasta el año 2018, ya que el crecimiento anual con respecto al año anterior también ha ido en aumento. Sin embargo, este no es el caso para el año anterior, en el que el crecimiento sufrió una bajada y pasó del 30.9% de 2018 a un 20% en 2019.

Desglosando estos ingresos, se pueden diferenciar dos tipos principales: aquellos que provienen de los productos de Amazon y aquellos que vienen de los Servicios de Amazon. Por Productos de Amazon se entiende todos los ingresos que provengan de la venta directa, de las tarifas de envío y del contenido multimedia consumido. Por Servicios de Amazon se entiende las tarifas impuestas a los vendedores, que incluye comisiones y cualquier gasto de cumplimiento o de envío, las ventas de Amazon Web Services, las tarifas de suscripción de Amazon Prime, la publicidad en las diferentes páginas webs y otros servicios ofrecidos de menor envergadura. Este desglose se puede ver gráficamente en el Gráfico 2, donde se aprecia que el crecimiento del porcentaje de ingresos de Productos de Amazon en comparación en el de Servicios está viéndose reducido constantemente cada año. En 2014 el 78'8% de los ingresos eran de Productos y el 21'2% de Servicios, en cambio, en 2018, el 60'9% era de productos y el 39'1% de Servicios. Esta relación irá en aumento, ya que como se puede apreciar, en el año 2018 el crecimiento de ingresos en los Servicios fue mayor que en los Productos. [6]

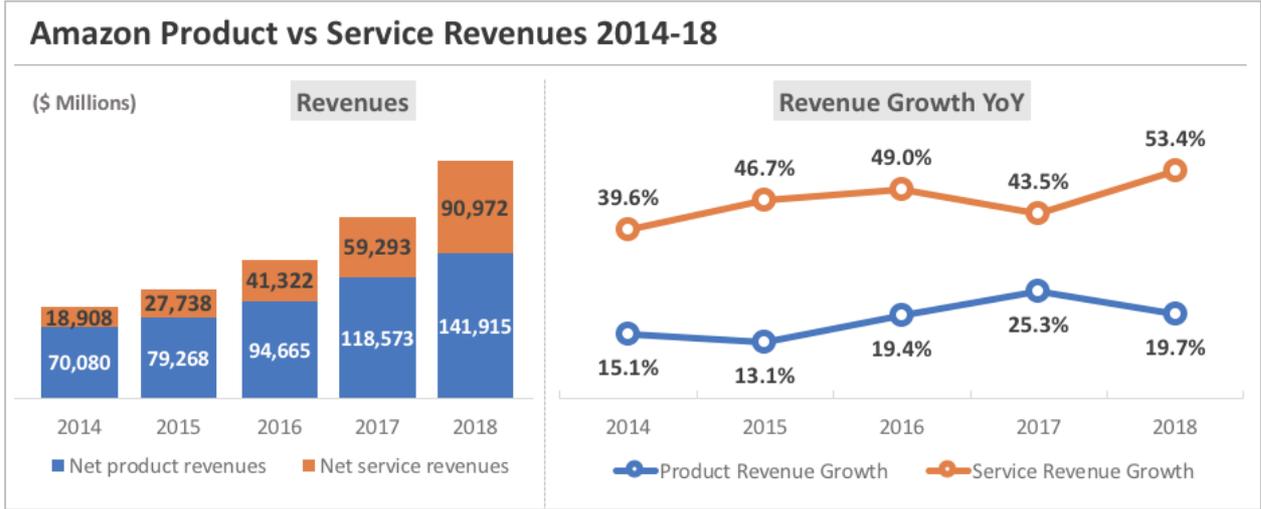


Gráfico 2. Desglose de ingreso de Amazon Inc. desde 2014 hasta 2018. Fuente: <https://mgmresearch.com/>

Por último, se muestra el Gráfico 3 con los ingresos anuales de Amazon en los últimos 15 años. Se puede ver que el crecimiento ha sido constante y es de esperar continúe siéndolo en los próximos años.

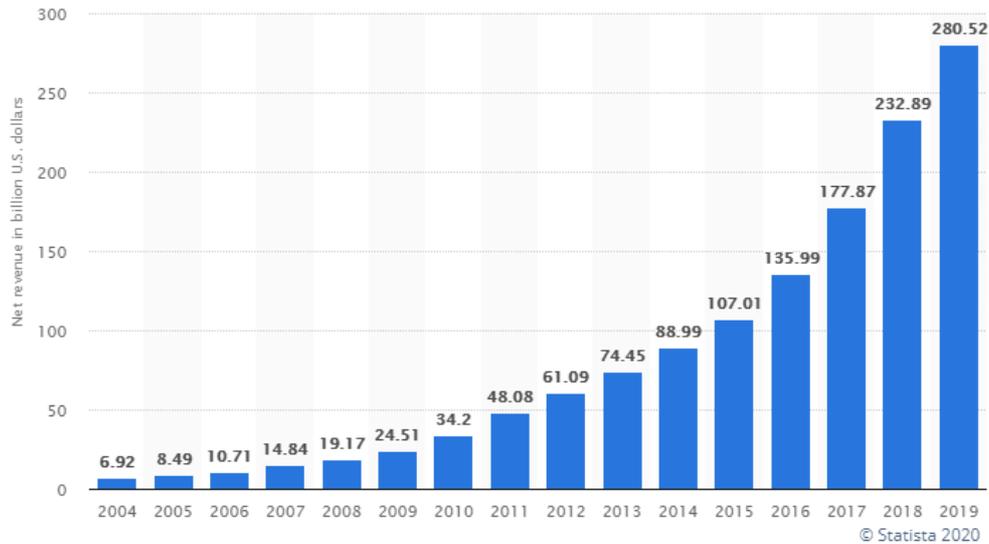


Gráfico 3. Ingresos anuales de Amazon.com. Fuente statista.com

1.1.3.2 Costes

El dinero que Amazon recibe es destinado a sufragar los costes que implican su actividad para obtener ingresos. Aquellos han ido creciendo a medida que lo han hecho los ingresos y se pueden clasificar de la siguiente manera:

- Coste de ventas (más importante de la empresa)
- Coste de cumplimiento, también llamado coste operativo.
- Coste de tecnología y contenido.
- Coste general y administrativo.
- Otros costes operativos de gastos netos.

Todos ellos serán mejor explicados en el apartado 1.3.3, donde se explicará la estructura de costes de la empresa.

En el Gráfico 4 se puede observar una comparativa de los diferentes gastos clasificados según lo enumerado anteriormente, tanto para el año 2018 como para el 2019. Se observa que el gasto de ventas supone la mayor parte del gasto global, siendo un 63,12% del total en 2018 y 62.24% del total en 2019. De mismo modo, también se comprueba que el gasto anual total ha crecido un 20.64% con respecto a 2018. En este sentido, es importante recalcar que, siendo los ingresos anuales de los Servicios de Amazon de 90.972 millones en 2018, solo incurren en unos gastos de 28.837 millones, es decir, un 31'7%. Esto hace que el mayor margen de beneficio provenga de esta fuente de ingresos.

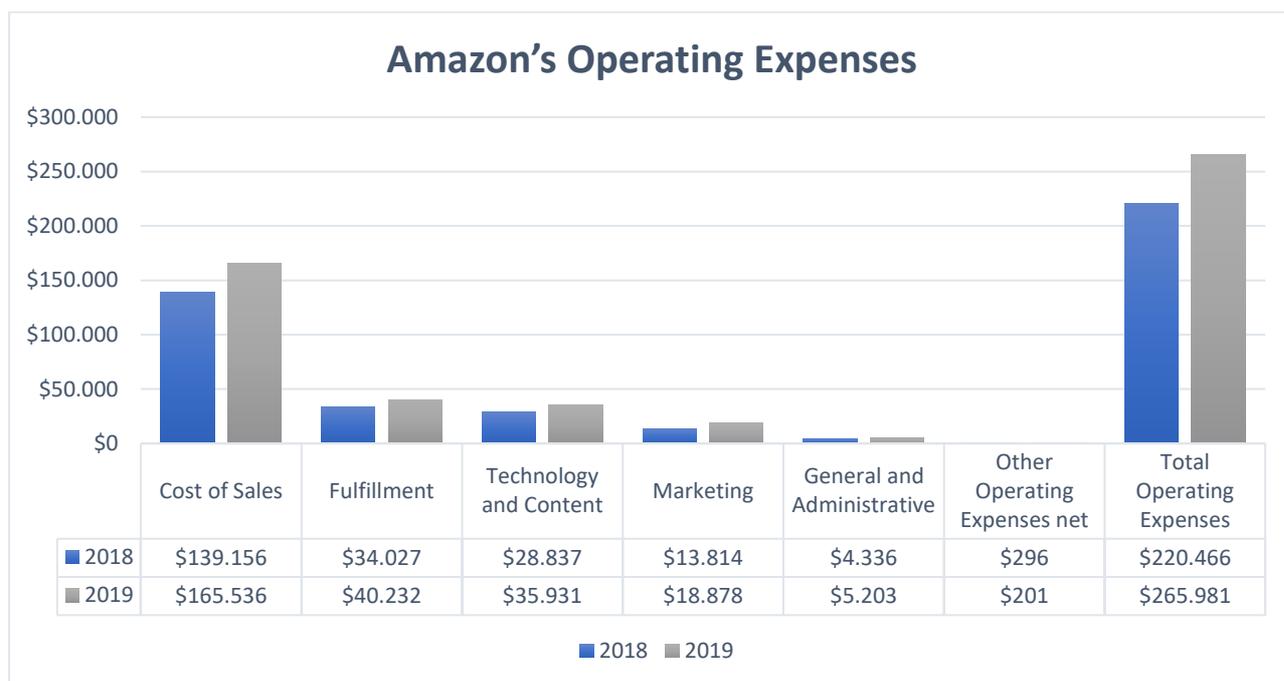


Gráfico 4 Gastos de Amazon.com, Inc. en 2018 y 2019. Fuente: statstic.com

1.1.3.3 Beneficio

El primer año en el que Amazon ganó sus primeros beneficios fue en 2001, con una cantidad de 5.000.000 \$ y desde ese momento han ido creciendo hasta los 11.600 millones de dólares en 2019; es decir, un beneficio que supone un 2320% más que en el 2001.

Esta subida de los beneficios ha ido directamente relacionada con el gran aumento de los ingresos que ha tenido la compañía desde que se fundó. El crecimiento no siempre ha sido constante; ha habido años o momentos del año en los que ha sufrido pérdidas con respecto al trimestre o al año anterior. Como por ejemplo en 2005, año en el que los ingresos aumentaron considerablemente (un 25%), pero que por culpa de los gastos en temas legales el beneficio esperado se redujo un 3.4% (30 millones de dólares en comparación con el año anterior). Lo que supuso un desplome de sus acciones del 11,72%. [7]

La razón detrás de su escaso beneficio reside en la estrategia de negocio que sigue la empresa. En 2017, invirtió gran parte de los ingresos en servicios como Amazon Prime, Amazon Web Services y Alexa, consiguiendo con ello mayor cuota de mercado a costa de su beneficio. [8] Esto desde el punto de vista financiero, para muchos, no es atractivo, sin embargo, su cuota de mercado llegó en 2019 hasta el 49% en venta online en Estados Unidos, siendo esta cuota mayor que la de sus tres grandes competidores juntos: eBay (6'6%), Apple (3'9%) y Walmart (3'7%) [9]. Se muestra así en el Gráfico 5.

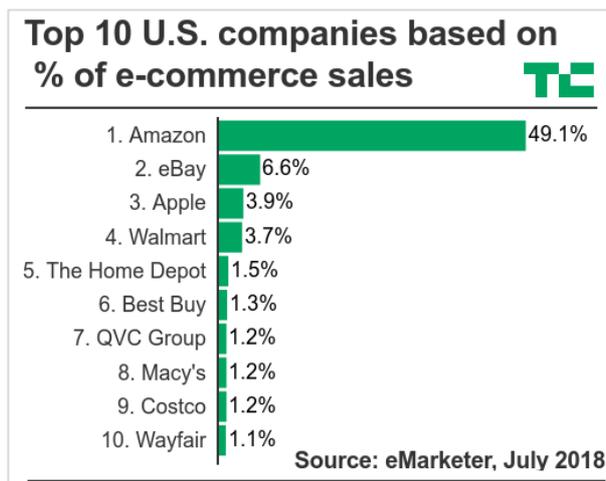


Gráfico 5: Diez compañías con mayor cuota de mercado en venta online en 2018, EE. UU. Fuente: eMarketer [9]

Actualmente, los beneficios de Amazon tienen una tendencia positiva. De hecho, el trimestre con más beneficio se ha registrado en el tercero de 2020, cerrado septiembre. Este hecho se comentará más adelante en el apartado 1.1.3.4. El mayor crecimiento se dio en el año 2018 en el que finalizó con un beneficio de 10.073 millones de dólares, lo que supuso un 332% más que en el año anterior, con un beneficio de 3.033 millones de dólares.

Entre las razones de este aumento se encuentra el crecimiento de la venta de productos vendidos directamente por Amazon, ya que son los que más beneficio le reportan. En el último año se ha vendido un 32'8% más de productos directos que en el año anterior. Por otra parte, también ha aumentado los beneficios obtenidos a través las tarifas que Amazon aplica a los vendedores que usan su Marketplace, ya que las ventas totales de estos han experimentado a nivel global un incremento positivo de sus ventas del 54'7%. En estas tarifas se incluye no solo las comisiones por el uso del Marketplace, sino también las relacionadas con el servicio de Fullfiment by Amazon (FBA), el almacenamiento de productos y los servicios logísticos, entre otros.

1.1.3.4 Amazon en la pandemia del Covid-19, tercer trimestre de 2020.

Debido a la última gran pandemia que asola al Mundo, la mayor empresa de comercio electrónico a sufrido un enorme incremento en sus ventas. La pandemia comenzó a afectar a los principales países donde Amazon tiene más fuerza en torno a marzo o abril de 2020, dependiendo del país. Se cerraron bares, negocios y minoristas, se confinó a la población durante meses, lo que propició un aumento de las compras a través de Internet. Justo después, en el cierre de cuentas del segundo trimestre de Amazon, se vieron reflejados los resultados de estas compras y de otros servicios. El beneficio neto trimestral alcanzó un máximo histórico de 4.426 millones de dólares y los ingresos totales llegaron a los 75.059 millones, un 40'2% más que el año anterior.

Además, no solo han crecido sus ventas a través de su tienda online y del Marketplace, sino que también lo ha hecho los ingresos a través de Amazon Web Services. Debido al incremento del teletrabajo, muchas empresas se han visto obligadas a contratar servicios informáticos para dar soporte a la infraestructura tecnológica de sus empresas para, así hacer posible el trabajo a distancia. Esto se ha traducido en un incremento de los ingresos a través de los Servicios de Amazon del 40'4% en comparación con el año anterior, es decir, 32.643 millones de dólares. [10]

Para el siguiente trimestre y ya terminado septiembre los ingresos de Amazon se establecen en un acumulado de 260.509 millones de dólares en estos nueve últimos meses, lo que supone un crecimiento del 34,9% con respecto al año anterior, en el que se obtuvo para la misma fecha un ingreso acumulado hasta el tercer trimestre de 193.086 millones. Sin embargo, el total de los ingresos del tercer trimestre han sufrido un crecimiento menor que los del segundo: un 37'4% frente a un 40'2% entre abril y junio.

En cuanto al desglose mencionado en el apartado 1.1.3.1, ingresos por Productos de Amazon e ingresos por Servicios de Amazon, para este último trimestre, los ingresos han sido de 96.150 millones de los cuales el 54.8% (52.770 millones de dólares) provienen de los Productos de Amazon y el 45.2% (43.370 millones) provienen de Servicios de Amazon. Si se entra más en detalle, de estos últimos ingresos 11.600 millones pertenecen a AWS, 5.400 millones de otros ingresos como la publicidad de sus páginas y 6.570 millones de las suscripciones. [11]

Por otro lado, los beneficios también aumentaron considerablemente se ha pasado de un beneficio acumulado de 8.320 millones de dólares en 2019, acabado el tercer trimestre, a uno de 14.109 en 2020, es decir, un 69'58% más. Y solo en el tercer trimestre se ha llegado a los 6.331 millones, un 210'73% más que en el año anterior para las mismas fechas.

Si se desglosan estas cifras, a nivel internacional también aumentan los beneficios en el tercer trimestre de 2020, pasan de unas pérdidas de 386 millones de dólares en el tercer trimestre del año anterior a un beneficio de 407 millones, sobre todo, provenientes de Alemania y Reino Unido. Al mismo tiempo, se sigue expandiendo y aumenta el número de Marketplaces fuera de Estados Unidos, siendo la última apertura del negocio en Suecia (amazon.se).

1.2 Organización y jerarquía

El diseño organizacional de Amazon se caracteriza por ser de estructura mecanicista, es decir, muy riguroso y altamente controlado, debido principalmente, al gran tamaño que tiene y al elevado número de empleados del que dispone. En ella las decisiones están en su mayoría centralizadas y ligadas a una jerarquía con una cadena de mando claramente definida. También destaca por su división en departamentos altamente especializados y, sobre todo, por la casi completa estandarización de todos los puestos de trabajo. Esto último supone que las funciones de un empleado en un determinado puesto estén muy definidas, es decir, compuestas por un conjunto de reglas y procedimientos del que es complicado salir. Del mismo modo, los flujos de información y de toma de decisiones también están estandarizados.

Este diseño jerárquico consta de un equipo directivo formado por dos directores ejecutivos, tres vicepresidentes senior y un controlador mundial, que reportan directamente al CEO, Jeff Bezos. A su vez, se encuentra una segmentación funcional dividida en siete áreas: Plataforma, Administración del producto masivo, Tecnología de Información, Administración y Finanzas, Recursos Humanos y, finalmente, Operaciones Legales. También se podría añadir el último segmento Amazon Studios & Prime Video. [8]



Figura 2: Organigrama de Amazon. [8]

En un nivel más general, Amazon tiene un órgano fundamental que lo gestiona, llamado Consejo de Administración de Amazon, compuesto por diez expertos a nivel mundial en gran variedad de industrias y comercios. Entorno a este consejo se encuentran tres órganos o comités que asisten al Consejo: Comité de Auditoría, Comité de Nominación y Gobierno Corporativo y Comité de Desarrollo y Compensación de Liderazgo.

A pesar de tener una estructura organizacional mecanicista, es decir, jerárquica y centralizada,

Amazon se caracteriza también por ser una empresa flexible, que se adapta a los cambios globales e incluso que los promueve debido a su potencial innovador. Dicha flexibilidad se sostiene principalmente en la figura de Bezos y a su capacidad de visión de negocio. Esta sinergia entre Jeff Bezos y la estructura organizacional centrada en su persona permite a Amazon ser una empresa adaptativa. [8]

1.2.1 Cultura de la organización

La organización de Amazon se ve impregnada en una cultura fundamentada en tres pilares principales:

1. Alto grado de centralización del cliente.
2. Reducción al máximo de los costes de la compañía.
3. Innovación constante y mejora continua.

En cuanto al primer punto, desde el comienzo de la compañía la misión fue centrarse en el cliente, en palabras de Jeff Bezos: “Nuestra visión es ser la empresa más centrada en el cliente de la tierra: construir un lugar donde las personas puedan encontrar y descubrir todo lo que quieran comprar en línea”. Esto da a entender que el enfoque al cliente implica desde la perspectiva de la compañía un dominio global del comercio electrónico, la mayor oferta de productos en todo el mundo y un alcance que llegue a todos los puntos de la Tierra.

En cuanto al segundo punto, como en el análisis que se hizo en el apartado 1.1.3, Amazon no comenzó a percibir sus primeros beneficios hasta seis años después de haber fundado la empresa, a pesar de que sus ingresos fueran millonarios desde el comienzo. Esto deja ver el elevado coste que ha tenido siempre la compañía, por lo tanto, es claro que la reducción de éste haya sido y sea siempre un punto fundamental donde centrar sus esfuerzos.

Finalmente, con respecto a la Innovación constante, ya lo dice la cita de Jeff Bezos al comienzo de este capítulo: “En el mundo antiguo, dedicabas un 30% de tu tiempo en crear un gran servicio y un 70% en difundirlo. En el nuevo tiempo, eso se invierte.” La innovación es clave del éxito en toda compañía y es fundamental, en el caso de Amazon, para mantener el liderazgo. A más se innova, más se diferencia de la competencia y a más se diferencia, más competitivos se es.

1.3 Modelo de negocio

1.3.1 Propuesta de valor

El negocio principal de Amazon es la venta, almacenamiento y distribución de productos al por menor (B2C) a través de su tienda online y su Marketplace. También ofrece otros servicios como Amazon Prime, Amazon Web Services (B2B) y Kindle Store entre otros.

1.3.2 Fuente de ingresos

En este negocio tan variado la distribución de los ingresos es la que se puede apreciar en la Figura 3, siendo la más importante, y la que hasta la fecha lo ha sido siempre, su tienda online y Marketplace, seguida de AWS y luego de sus servicios que requieren suscripciones como Amazon Prime o Amazon Video.



Figura 3: Distribución de ingresos de Amazon en 2018. Fuente <https://www.bbc.com/mundo/noticias-48909676>

La tienda online de Amazon se refiere a todos los productos vendidos directamente por la compañía, mientras el Marketplace recoge a todos aquellos productos vendidos por terceros. Por lo tanto, no es de extrañar que la mayor parte de los ingresos de Amazon venga de la tienda online, ya que en este el margen de beneficio es mayor que en el Marketplace. Por otro lado, es importante remarcar que, aunque los beneficios por el Marketplace sean menores, la cantidad de productos vendida por terceros es mayor. En 2019 las ventas de los productos vendidos en el Marketplace ascendieron a 200.000 millones de dólares, mientras que las ventas de Amazon como vendedor directo fueron de 135.000 millones, es decir un 60% vendido por terceros y 40% por la empresa de Bezos. En el Gráfico 6 se

puede ver la evolución de la proporción de productos propios de Amazon y los productos de otros vendedores vendidos en la página web Amazo.com, desde 1999, año en qué la compañía comenzó a vender por primera vez productos de otros vendedores, hasta la actualidad. [12]

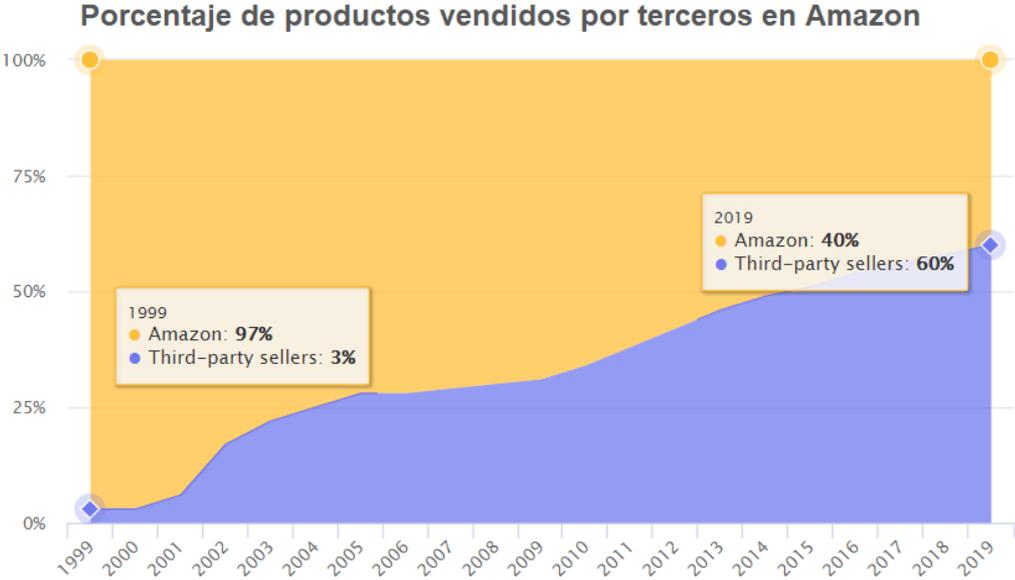


Gráfico 6. Porcentaje histórico de relación de productos vendidos por Amazon y en el Marketplace

1.3.3 Estructura de los costes

Con respecto a los costes de Amazon, la mayor parte corresponde al coste de ventas, seguido a la logística, transporte y almacenamiento de vehículos. Estos han ido creciendo a medida que lo han hecho los ingresos, tanto los costes operativos como los de investigación y desarrollo, así como el coste de cumplimiento.

El coste de ventas es el mayor que tiene Amazon, en el que se incluye: el precio de compra de todos los productos que adquiere el consumidor final, el coste del transporte tanto a la entrada como a la salida de los centros logísticos, el coste de almacenamiento y clasificación, el coste de marketing, y, finalmente los costes digitales que dan soporte a estos productos. Los costes de transporte son lo que en proporción más están creciendo, debido a los nuevos servicios que se están ofreciendo y a la disminución paulatina del tiempo de entrega.

El coste de cumplimiento se refiere a todos aquellos asociados a la contratación del personal, a los procesos de pagos y a las operaciones en sus centros logísticos norteamericanos e internacionales, en las tiendas físicas y en los “Costumer Service Centers”. En los costes de procesos de pago también se incluye los pertenecientes a AWS, aunque el resto de los costes asociados a este servicio se incluye en los de Investigación y desarrollo.

Los costes de tecnología y contenido (Investigación y desarrollo, I+D) incluyen los asociados al personal encargado de desarrollar nuevas tecnologías para la empresa, los encargados de dar soporte a la infraestructura tecnológica y también a todos los costes que implica el AWS (servidores, equipos, centros de datos, etc.).

Los costes de Marketing son aquellos asociados a la publicidad y aquellos dirigidos a mantener el personal de los departamentos de marketing y de ventas.

Finalmente, se encuentran los costes generales y administrativos y otros gastos generales. [13]

1.3.4 Actividades, socios y recursos

Las principales actividades de Amazon se pueden clasificar en tres: logística, servicios online y productos propios. Dentro de logística se encuentra la clasificación, el transporte y el almacenamiento de los productos vendidos o gestionado a través de la página web. Dentro de los servicios online destacan todas aquellas actividades dirigidas a mantenerlos y darles soporte. Por otro lado, las actividades que abarcan los productos propios son el diseño, desarrollo, producción, búsqueda de proveedores, marketing, etc., que permiten que el producto salga a la venta.

En cuanto a los socios de Amazon los más importantes son los que se encuentran aguas abajo de la venta del producto y se refiere a las partes externas a la empresa involucradas en el desempeño de su actividad. Entran dentro de este grupo los vendedores, las empresas de logística, las empresas manufactureras encargadas de producir los productos propios de Amazon como Alexa, Kindle, la gama Amazon Basics, Presto, etc., u otros proveedores. [8]

Por último, en los recursos de Amazon entra todo aquello propio a la empresa que permite la actividad principal. Por ejemplo, la infraestructura: almacenes, centros logísticos, servidores, equipos, etc. También se pueden encontrar las patentes pertenecientes a la marca. Un ejemplo de una de ellas fue el proceso de compra de “1-Click”, creado en 1997 y expirado como patenten en 2017, que permitía al consumidor final evitar todos los procesos de compra y con solo un clic pagar y hacer que el producto se enviara a su casa.

1.3.5 Canales, segmentación y relación con los clientes.

El principal canal de Amazon, y casi único, es su página web o el conjunto de páginas que forman Amazon Inc. Desde el comienzo Amazon nació como una plataforma de venta online y es así como sigue siendo a día de hoy. Prácticamente todos sus servicios (AWS, Kindle, venta de productos) se realizan a través de su página web. Solo se salva las tiendas físicas que se están comenzando a abrir: Amazon Go.

La primera se abrió en 2016 en su sede en Seattle y ya cuenta con más 500 tiendas solo para su marca Whole Foods. Este tipo de marca se encuentra dentro de las siete categorías de tiendas físicas que tiene la compañía, cuatro son para compra de comida, siendo la más importante la ya mencionada. Aunque, estos negocios están en fase embrionaria y todavía no han despegado, supone un primer acercamiento de Amazon al comercio minorista. [14]

En cuando al modo en que Amazon se relaciona con sus clientes, destaca principalmente el servicio Amazon Prime, a partir del cual el consumidor obtiene servicios tales como envío gratuito, ofertas en días como Prime Day, suscripciones gratuitas en Twitch, o acceso al catálogo de vídeos online Amazon Video. También permite la relación entre los clientes mediante las opiniones sobre los productos y la opción de plantear dudas a otros clientes o directamente al proveedor en caso de ser un vendedor externo.

Finalmente, los clientes de Amazon se pueden clasificar en empresas, consumidores finales, vendedores, desarrolladores y creadores de contenido.

- Consumidores finales: Fueron los primeros clientes de Amazon y siguen siendo su principal fuente de ingresos; estos consumidores son los que le dan a Amazon la etiqueta de minorista (B2C), aunque poco a poco se está convirtiendo en algo más que eso. A estos clientes se les sirve a través la página web y de las tiendas físicas, dónde Amazon les ofrece productos propios y de terceros.
- Vendedores: El lugar en Amazon para estos clientes es el Marketplace, mediante el que las diferentes empresas venden sus propios productos. Para este grupo también existe un servicio de fidelización Fulfillment by Amazon, donde no solo una empresa vende sus productos a través de Amazon, sino que, además, la gestión física de dichos productos la realiza Amazon mediante su infraestructura, centros logísticos y personal.
- Empresas y desarrolladores: En este punto no se incluyen las empresas que venden productos a través de Amazon, sino aquellas que contratan sus servicios de AWS. Estos son contratados por compañías y desarrolladores que buscan sus amplios servicios de computación, almacenamiento y bases de datos entre otros.
- Creadores de contenido: En este segmento de clientes se encuentran autores, músicos, pequeñas productoras de películas, desarrolladores de aplicaciones y otros tipos de publicadores que crean y quieren vender sus propias creaciones.

1.4 Logística y Operaciones de Amazon

En este apartado se hará un breve análisis de cómo Amazon lleva a cabo su actividad y cuáles son los objetivos que la determinan. También se presentarán proyectos destacados, actuales o futuros, que darán una visión más profunda sobre este punto. La finalidad de este análisis es entender cómo el problema matemático se materializa, es decir, qué infraestructura hay detrás, sobre qué pilares se sustenta y qué objetivos persigue.

1.4.1 Objetivos - The Bezos Virtuous Cycle

Los objetivos de la logística de Amazon se engloban dentro del llamado The Bezos Virtuous Cycle (El círculo virtuoso de Bezos, Figura 4), un círculo virtuoso o ‘volante de inercia’ (*flywheel*) empresarial aplicado a la forma de proceder de Amazon. Este fue desarrollado por Jeff Bezos en 2001, como explica Jeff Wilke, y desde entonces ha sido la estrategia de marketing de la empresa.

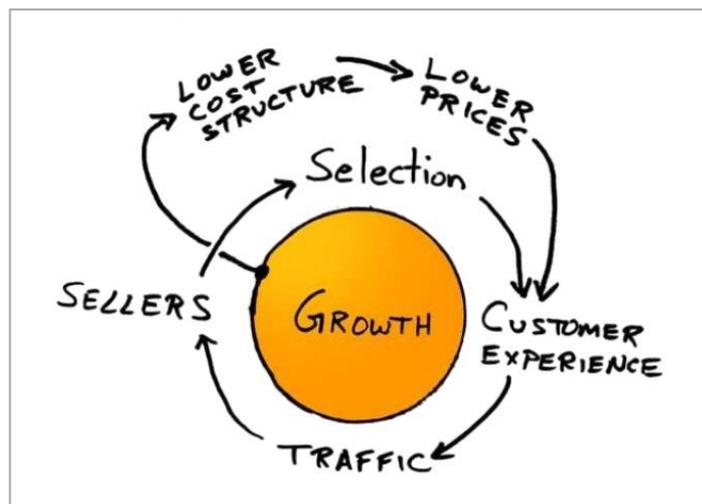


Figura 4: El círculo virtuoso de Bezos. Fuente: Amazon.com

Se puede decir que el ciclo comienza con la experiencia de compra de cliente, si este considera que ha sido una gran experiencia atraerá por el boca a boca a más consumidores, lo que provocará un incremento del tráfico de consumo. En este punto la empresa podría vender única y exclusivamente sus productos ganando así más margen de beneficio, aunque también lo podría aprovechar empleando su infraestructura para permitir a terceros comercializar sus productos. Optando por esta última oportunidad y sirviéndose del aumento de tráfico, se consigue atraer a más vendedores, que quieran vender sus productos en Amazon para darles más visibilidad, y también se consigue que los ingresos de la empresa crezcan, aunque en un primer momento sean menores que si vendiera únicamente sus productos.

Consecuentemente, a mayor número de vendedores, mayor variedad de productos se venderán en el Marketplace y, por lo tanto, mejor experiencia de compra tendrá el cliente, llegando así al comienzo

del círculo virtuoso.

Por otro lado, si aumentan los beneficios la compañía podrá optar por dos vías, o distribuye la mayor parte de la ganancia entre sus accionistas o la utiliza para aumentar y mejorar la infraestructura de la empresa. Claramente, optando por la segunda opción se crea un segundo círculo virtuoso. La mejora de su infraestructura abarca dos objetivos. El primero es el de mantener y adaptarse al crecimiento del negocio, es decir, cuando aumentan el tráfico de compras y el número de vendedores es soporte logístico debe ser mayor, lo que implicará una inversión. El segunda es el de mejora, este punto es el que alimenta y genera un segundo círculo virtuoso, ya que una apuesta en la infraestructura reduce a la larga el coste logístico o incluso permite generar nuevos servicios (por ejemplo: Amazon Prime o entrega rápida en menos de 48 horas), la reducción del precio y la mejora de servicio conducirán a un aumento de la experiencia del cliente.

Desde el punto de vista logístico el círculo virtuoso se puede desengranar de la siguiente manera (Figura 5): una mayor demanda implica un mayor caudal de productos en movimiento, lo que provocará la necesidad de aumentar la flota y una mayor utilización de esta y por consiguiente, se reducirá el coste de utilización, se reducirán los precios y crecerá la demanda. También, aumentará las solicitudes de mejores servicios, por lo que se deberá aumentar no solo la eficiencia, si no también la mejora de la calidad de la entrega. Lo que se traducirá, también, en un crecimiento de la demanda.

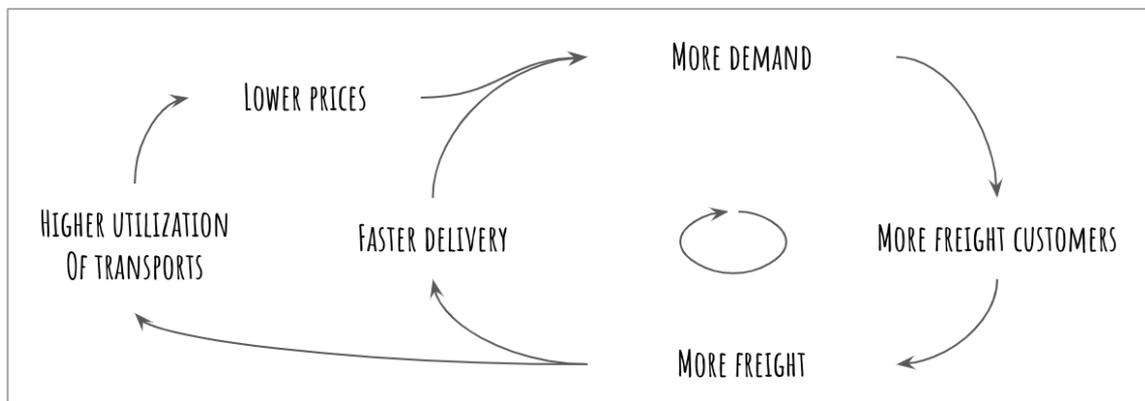


Figura 5: Círculo virtuoso de Bezos desde el punto de vista logístico. Fuente [15].

1.4.2 Funcionamiento

Para poder llevar a cabo la principal actividad de Amazon y alcanzar los objetivos planteados en el apartado anterior, es fundamental tener un sistema logístico estructurado y bien definido. Se debe tener en cuenta que, en un día con alta demanda, Amazon puede gestionar una media de 35 pedidos por segundo, de donde se deduce que la infraestructura logística debe ser robusta si se quiere llegar a cubrir las necesidades del cliente correctamente.

1.4.2.1 Pedidos, robots y “caos organizado”

En el desarrollo en Amazon de una red logística robusta juega un papel fundamental el grado de automatización de sus almacenes, planificados según la teoría del “caos organizado”. Este tipo de planificación no organiza los artículos por categoría o por temática, como lo puede estar un supermercado, sino que los productos se encuentran ubicados al azar a lo largo y ancho del almacén. “Así los recogedores de pedidos siempre tienen el producto cerca. Además, el sistema calcula a la ruta más corta y se optimiza el recorrido. Es el caos organizado, pero siempre sabemos dónde está el producto en cualquier momento”, según explica Fred Pattje, director de Operaciones de Amazon España [16]. Esto también hace que se reduzcan las probabilidades de que un robot recogedor se equivoque al tener que escoger entre dos productos similares, ya que cada artículo se encuentra rodeado de otros diferentes. Estos robots, serán los encargados de entregar los pedidos a los trabajadores, ahorrando así 1’2 km de distancia recorrida a cada empleado. En este proceso, son primero estos últimos los encargados de indicar a los robots el lugar de los nuevos productos, para que más tarde, puedan gestionarlo según el pedido de los clientes.

Una vez el empleado recibe por parte del robot el conjunto de artículos que conforman el pedido, se empaqueta y se sitúa en una cinta transportadora que lo procesa y clasifica según sus características. Dependiendo de esta clasificación, el producto se gestionará a través de una empresa logística u otra.

1.4.2.2 Organización del centro logístico

Se podría dividir en dos grandes partes el centro logístico de Amazon, la primera el punto de recepción del material y la segunda el del procesado y envío de pedidos. En la primera se realiza un control de calidad, se toman medidas de los artículos, se pesan y se les realiza fotos que son cargadas a la web. La segunda parte, se sigue el procedimiento explicado en el apartado anterior: se envían los pedidos empleando el uso de robots, cintas transportadoras y clasificadoras.

En resumen, el esquema que sigue un producto es el siguiente: Recepción de nuevos productos, procesamiento, almacenamiento, recepción de pedidos, selección de productos que componen el pedido, empaquetamiento, asignación del pedido según sus características, envío a centros de clasificación, procesado en centros de clasificación, asignación de envío en última milla y recorrido de la última milla hasta el cliente.

1.4.3 Tipos de instalaciones

La red logística de Amazon está compuesta por un estado del arte de tecnología y gran variedad de edificios de diferente tipo y tamaño, que dan soporte a todo el negocio y que hacen posible la entrega de pedidos a los clientes:

1. Centros logísticos “clasificables”: en ellos se gestionan todo lo relacionado con la logística asociada a los pedidos del cliente. Se almacenan productos, se empaquetan y se distribuyen buscando optimizar la eficiencia y eficacia, es decir, maximizando la satisfacción en la experiencia de compra del cliente. Pueden albergar hasta 1500 empleados y van asociados a marcados procesos de automatización, como el caso de los Robots, explicado en el apartado anterior. También, existen departamentos de mejora continua que buscan reducir desperdicios en la cadena de valor.
2. Centros logísticos “clasificables”: pueden ser más grande que los anteriores y están enfocados en productos grandes que no pueden ser clasificados de manera automática por robots como en el caso anterior. Va dirigido especialmente a artículos grandes de jardín, muebles, etc.
3. Centros de clasificación: reciben exclusivamente paquetes (pedidos consolidados) que deben ser enviados a centros de entrega. En estos centros, los pedidos son clasificados teniendo en cuenta el lugar y el tiempo requerido de entrega y según esto se almacenan en un camión u otro.
4. Centros especializados: procesan artículos de determinadas categorías que suelen sufrir picos de demanda. También son los centros que absorben los excesos de pedidos en campañas de gran flujo de compras como La Navidad o el Black Friday.
5. Centros de entrega: Se sitúan muy cerca del consumidor final, de manera que los pedidos puedan ser repartidos rápidamente, sirven de apoyo para recorrer la última milla
6. Centros de recepción (Figura 7). Son los que reciben grandes cantidades de productos por parte de los vendedores o proveedores. En ellos estos productos se envían en diferentes cantidades a los centros logísticos según la demanda en cada uno de ellos.

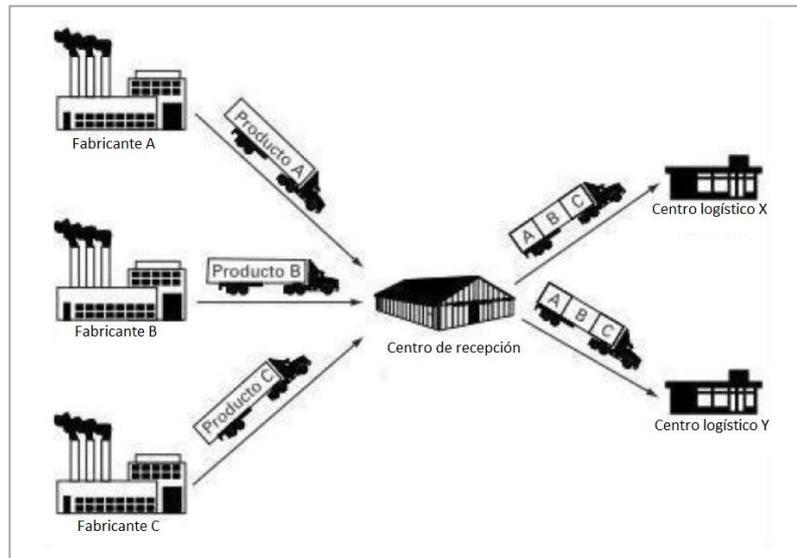


Figura 6: Ejemplo de centro de recepción [17].

1.5 Amazon en Sevilla y la problemática

En España Amazon posee seis centros logísticos: San Fernando de Henares (Madrid), El Prat (Barcelona), Martorelles (Barcelona), Castellbisbal (Barcelona) e Illescas (Toledo) y, el último que ha abierto, el de Dos Hermanas (Sevilla). Este último centro logístico se adhiere a la red logística de Amazon en Andalucía, aumentando así, la descentralización de las actividades logísticas de la compañía en toda España.

La red andaluza estaba compuesta principalmente por una estación logística en Málaga de 3.000 metros cuadrados y una nave adquirida en 2019 en la ZAL (Zona de Actividades Logísticas) del puerto sevillano, también con carácter de estación logística, con más de 8.000 metros cuadrados construidos y 30.000 de parcela. Esta última sustituyó a otra estación en Sevilla de 10.000 metros cuadrados en el polígono la Negrilla en el Centro de Transportes de Mercancías de Sevilla (CTM), abierto en 2017.

A estas dos estaciones logísticas se le suma el centro de logístico, ya mencionado de Dos Hermanas. Este provee a las estaciones logísticas y dispone de una mayor variedad de artículos, sirve también de almacén y de centro distribuidor al resto de Europa. La extensión es de 190.000 metros cuadrados distribuidos en cuatro plantas, cuya parte principal es la nave de almacenaje y proceso de mercancías.



Figura 7: Recreación del centro logístico de Amazon en Dos Hermanas (Sevilla).

El centro de Dos Hermanas cuenta con conexión directa a la SE 40 y a la A-4 y además se encuentra cerca del puerto de Sevilla. Esto le dará acceso a prácticamente toda el área metropolitana de Sevilla, aunque actualmente, no tiene acceso directo al área oeste de Sevilla, ya que el tramo de la SE-40 que une Dos Hermanas y Coria no está terminado, como se puede apreciar en la Figura 8. De este punto surge un problema importante, al estar este tramo inacabado, la única conexión más al norte con la zona oeste metropolitana de Sevilla se centra en la SE-30, concretamente en el Puente del Centenario, lo que se traduce en una zona congestionado con gran densidad de tráfico. A través, de este puente pasa casi todo el tráfico de Huelva a Cádiz, además del propio interno de la SE-30, es decir, más de 100.000 vehículos diarios [18]. Luis Lázaro, director de la compañía Merlin Properties, dueña de la parcela donde se encuentra le centro logístico de Amazon, afirma que «cualquier avance en las comunicaciones que permitan ahorrar tiempo de transporte es importantísimo», la SE-40, «es vital para mejorar la conexión y evitar el puente del Centenario, que es el mayor problema de este entorno» [19].



Figura 8: Circunvalación exterior de Sevilla. Tramos acabados e inacabados. Fuente: Wikipedia.commons.

Partiendo de este punto, y de la importancia de los costes logísticos para Amazon, se hace necesario la reducción de estos en el área metropolitana de Sevilla, especialmente en la conexión de Dos Hermanas con el oeste, es decir el denominado Aljarafe.

De este último proviene gran parte de la movilidad de vehículos, sobre todo, de la Zona Aljarafe Central y Aljarafe Sur (A-49, A-8057 y A-8058). Según un informe integral de movilidad realizado por el ayuntamiento de Sevilla en 2019 [20] de los 399.284 vehículos que circulan diariamente por Sevilla el 46'3% proviene del Aljarafe y de la zona de Montequinto y Alcalá de Guadaíra.

En otro estudio realizado por la fundación RACC sobre la congestión las vías principales de acceso a Sevilla en 2011 [21] se hace un análisis en un rango horario de 06:00 a 22:00 de la velocidad y el número de vehículos que recorren las principales vías no semaforizadas de alta capacidad (autovías y autopistas). En el análisis se identifican siete corredores principales para acceder a la ciudad: Sevilla, Ribera Norte, Aljarafe Norte, Aljarafe Sur, Norte, Este y Sur, como se muestra en la Tabla 1 (también en la Figura 9 se puede ver un plano con las zonas marcadas). En el estudio, se indica que la mayor densidad y concentración de vehículos se da a las 8 de la mañana entre el Puente del Centenario y el de los Reyes de España, 1.400 vehículos por cuarto de horas. Mientras que es en el tramo Norte donde menor densidad se registra: 500 vehículos por cuarto de hora. Además, también se indica que no se registra en el corredor de circunvalación grandes diferencias dentro del horario laboral: entre los 800 y 1.100 vehículos. Lo que muestra una gran utilización de estos tramos y de su importancia en la ciudad de Sevilla.

| Corredor | Identificador | Denominación | Titularidad |
|----------------|---------------------|---------------------------------|-----------------------|
| Sevilla | SE-30 (km 0-km 9) | Ronda Circunvalación de Sevilla | Ministerio de Fomento |
| | SE-30 (km 9-km 18) | Ronda Circunvalación de Sevilla | Ministerio de Fomento |
| | SE-30 (km 18-km 29) | Ronda Circunvalación de Sevilla | Ministerio de Fomento |
| | SE-20 | Circunvalación de Sevilla | Ayuntamiento |
| Ribera Norte | A-66 | Autovía Ruta de la Plata | Ministerio de Fomento |
| Aljarafe Norte | A-49 | Autovía del Quinto Centenario | Ministerio de Fomento |
| Aljarafe Sur | A-8057 | Autovía de Mairena | Junta de Andalucía |
| | A-8058 | Autovía de Coria | Junta de Andalucía |
| Norte | A-4 | Autovía del Sur | Ministerio de Fomento |
| Este | A-92 | Autovía de Andalucía | Junta de Andalucía |
| Sur | A-4 | Autovía del Sur | Ministerio de Fomento |
| | A-376 | Autovía Sevilla -Costa del Sol | Junta de Andalucía |

Tabla 1: Identificación de las vías de acceso a Sevilla [21].

En las vías del aljarafe Sur (A-8057 y A-8058) se registra intensidades mayores en el punto de conexión de ambas vías y al unirse con la SE-30, legando a los 1200 vehículos por cuarto de hora entre las 07:00 y 07:30.

Por otra parte, se analiza también la intensidad de la velocidad en las vías principales. La disminución de velocidad más importante se da en el Aljarafe Norte, A-49, sobre todo por la mañana en la unión con la SE-30. En este tramo, existen también gran porcentaje de atascos en los fines de semana, cuando se producen movimientos a las playas de Cádiz y Huelva, sobre todo en períodos vacacionales.

Se indican los puntos más problemáticos en cuanto al tráfico en la zona metropolitana de Sevilla. Se enumera a continuación los que afectan al centro logístico de Amazon en Dos Hermanas en su actividad de abastecimiento a la zona oeste del área metropolitana:

- Puente del V Centenario (sentido oeste)
- Enlace A-8057 con A-8058 (en sentido entrada)
- Enlace A-49 con SE-30 (en sentido entrada)
- A-8057 entre los puntos kilométricos 2 y 5.

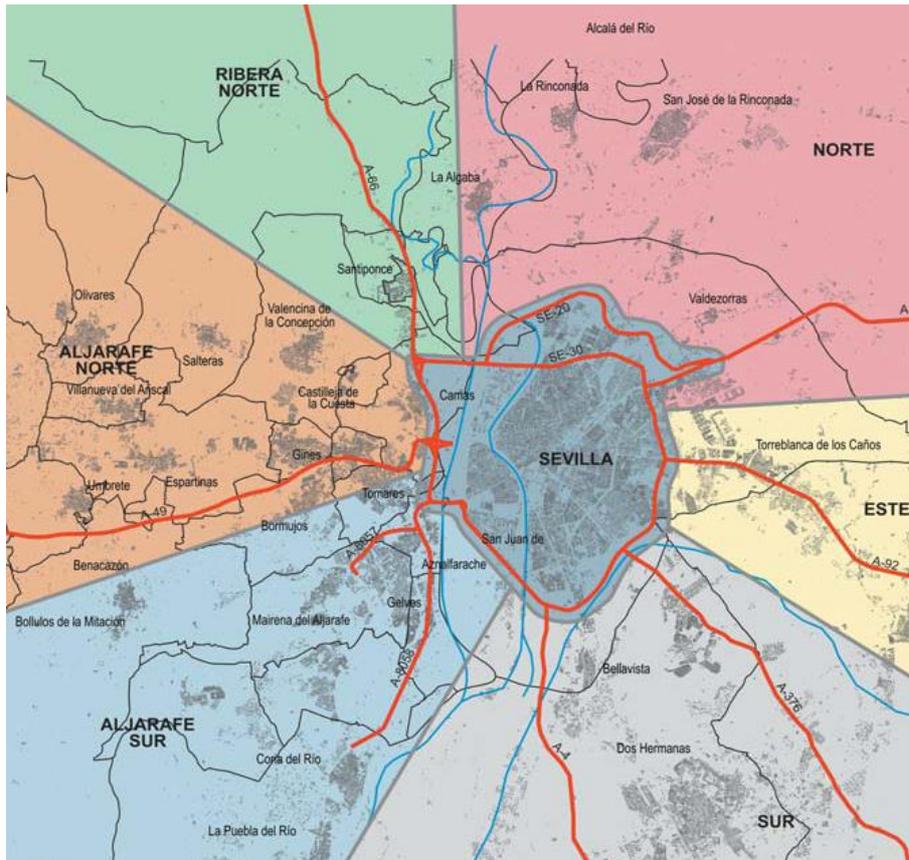


Figura 9: Vías de congestión metropolitana [21].

Finalmente, el mismo informe indica que los costes de congestión en las vías de entrada y salida a Sevilla se estiman en 267.258 euros al día, es decir, más de 66 millones de euros al año. En este coste se incluye, también, el consumo de combustible, el exceso de tiempo y las emisiones de CO₂, siendo el coste asociado a la unidad de tiempo es 9'36€/h, calculado según bases de datos de la fundación RACC. Además, también se muestran los costes medios anuales por usuario, donde en las partes que afectan a la zona oeste y suroeste se tienen: 129 €/año en la A-49, 222 €/año en la A-8058, 272€ en la A-8057 y 835 €/año en la SE-30 Oeste.

Estos datos reflejan los costes y la problemática asociada a la comunicación con la zona oeste de Sevilla. Por otra parte, se añade la importancia de la reducción de costes en Amazon, reflejado como uno de los pilares básicos para alimentar la estrategia de la empresa (Círculo virtuoso de Bezos, apartado 1.4.1), así como, punto clave para el propio crecimiento de la compañía (recuérdese que hasta 2001 Amazon no percibió beneficios en su cuenta de resultados, apartado 1.1.3.3). Y finalmente y en línea con los tiempos perdidos debido a la congestión de los corredores de Sevilla y con la importancia del servicio al cliente (mayor máxima y valor de Amazon y fundamento de la organización, apartado 1.2.1), se plantea en el presente Proyecto un modelado de la zona Oeste de Sevilla a la que se le aplicará el conocido problema del viajante, de manera que su resolución pueda servir como ayuda para la reducción de tiempos, distancias o costes. Por lo tanto, este objetivo, persigue la resolución del fruto

de la sinergia de los dos grandes problemas presentados en los anteriores apartados: el impacto de los costes en Amazon y la mala comunicación en la zona Oeste de Sevilla.

2 -EL PROBLEMA DEL VIAJANTE, ESTADO DEL ARTE

“Así como el hierro se oxida por falta de uso, también la inactividad destruye el intelecto.”

- Leonardo Da Vinci -

El problema del viajante, conocido por sus siglas en inglés como el TSP (*Travelling salesman problem*), consiste en dado un conjunto de nodos o ciudades, unidos entre sí por arcos, tratar de encontrar el camino más corto que pase por todos los nodos solo una vez y cuyo punto final e inicial sea el mismo. Un problema parecido fue propuesto por los matemáticos Sir William Rowan Hamilton y Thomas Kirkman, donde se puede apreciar en un juego creado por el primero de ellos, W.R. Hamilton, llamado “The Icosian Game” que consistía en definir un ciclo Hamilton a través de las aristas y vértices de un dodecaedro visitando única y exclusivamente una vez cada vértice. Por otra parte, el TSP, apareció por primera vez en un libro de viajes editado en Alemania, donde se mencionaba y definía el problema, pero sin llegar a recibir ningún enfoque ni trato matemático. Este tratamiento matemático del problema fue desarrollado por Karl Menger en Viena y Harvard en los años 30 del siglo pasado. Más tarde, el TSP fue promovido por Merrill Flood y Hassler Whitney gracias a su relación con RAND Corporation. [22]. Esto último, sumado a su facilidad de formulación y su gran complejidad (NP-Hard o NP-Completo) hizo que el estudio del problema del viajante se popularizara en los años venideros.

El “Travelling Salesman Problem” puede ser formulado de diferentes maneras, una de las más

conocidas e importantes es la de Miller, Tucker y Zemlin [23]: dado un número de ciudades n donde c_{ij} es la distancia entre cada una de ellas se define la variable dicotómica x_{ij} como:

| | | |
|--|--|-----|
| $x_{ij} = \begin{cases} 1 & \text{si el camino escogido va de la ciudad } i \text{ a la ciudad } j \\ 0 & \text{en otro caso} \end{cases}$ | | (1) |
|--|--|-----|

También se define la variable binaria u_i para $i = 1 \dots n$, quedando la formulación del problema de la siguiente manera:

Minimizar:

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

Sujeto a:

$$\sum_{i=1, i \neq j}^n x_{ij} = 1 \quad j = 1 \dots n \quad (3)$$

$$\sum_{j=1, j \neq i}^n x_{ij} = 1 \quad i = 1 \dots n \quad (4)$$

$$u_i - u_j + nx_{ij} < n - 1 \quad 2 \leq i \neq j \leq n \quad (5)$$

$$1 \leq u_i \leq n - 1 \quad 2 \leq i \leq n \quad (6)$$

$$u_i \in \mathbb{Z} \quad i = 2 \dots n \quad (7)$$

$$x_{ij} \in \{1, 0\} \quad i, j = 1 \dots n \quad (8)$$

$$c_{ij} > 0 \quad i, j = 1 \dots n \quad (9)$$

La función (2) es la función objetivo y minimiza el camino del viajante, las restricciones (3) y (4) obligan a que cada ciudad sea visitada una única vez y a que se salga siempre de la que se es visitada, respectivamente. Mientras que las restricciones (5) y (6) evitan que se creen *subtours* entre el conjunto de ciudades.

2.1 Aplicaciones reales

Aunque el problema definido por el TSP es ya de por sí útil, también se pueden encontrar diferentes aplicaciones del mismo que lo hacen aún más versátil. A continuación, se muestran varios planteamientos del problema cuya aplicación puede o no modificar el problema inicial, siendo siempre como referencia el TSP.

2.1.1 Ruta de vehículos

Es parecido al TSP, pero en lugar de tener un solo viajante se tiene una flota de vehículos que parten y vuelven a un mismo punto. De manera que todos los nodos sean visitados una única vez por uno de los k vehículos. De esta forma, para visitar un conjunto de n ciudades será necesario un número de k rutas. Por lo tanto, el objetivo del problema podrá ser minimizar la distancia total recorrida por el conjunto de vehículos o bien minimizar el número de vehículos empleados. Además, a este problema se le suelen añadir restricciones que impidan que un solo vehículo visite todos los nodos, ya que en ese caso nos encontraríamos frente al TSP. Una de las restricciones más comunes es la de la capacidad, donde cada vehículo tiene que servir cierta cantidad de bienes a las diferentes ciudades no pudiendo superar una capacidad máxima de artículos entregados C . También existen restricciones temporales, es decir, a cada nodo se le asocia una ventana temporal, fuera de la cual no podrá ser visitado.

2.1.2 Optimización de circuitos impresos de PCB

Se parte de una placa PCB donde ya se tiene distribuidos todos los módulos y puntos de conexión y a partir de aquí nacen dos problemas a optimizar. El primero de ellos se trata del orden de perforación, donde cada posición que debe ser perforado equivaldría a una ciudad en el problema del viajante, por lo que el objetivo es que la máquina perforadora haga el recorrido mínimo, lo que a escalas industriales supondría un importante ahorro de costes. El segundo de ellos consiste en conectar todos los puntos de la placa con sus respectivos chips (al menos dos a dos) de manera que la cantidad de cable empleado sea la mínima sin que, además, haya ninguna interferencia entre ellos. [24]

2.1.3 Minimizar gasto del empapelado de una pared

Presentado por Garfinkel [25] en 1977, este problema consiste en dado un rollo de empapelar de longitud W y una pared de alto w se deben cortar n hojas para empapelar la pared (entendiéndose que con un único rollo es suficiente), de manera que $n = \text{integer} \left(\frac{W}{w} \right)$. Suponiendo que el corte de la hoja i comienza en la posición s_i y que el corte final termina en la posición f_i , se tiene que:

$$f_i = s_i + l_i \quad i = 1 \dots n \quad (10)$$

Donde la longitud de l_i equivale a l_i veces el patrón que sigue el estampado del papel. Por otra parte, la cantidad de papel que se pierde tras cortar la hoja j después de la hoja i es:

$$c_{ij} = s_j - f_i \quad i, j = 1 \dots n \quad (11)$$

Suponiendo que el rollo finaliza en la posición f_0 , tras la última hoja el rollo debe acabar en el mismo punto de partida $s_0 = f_0$. Esto hace que se le añada una hoja más adicional al problema, por lo tanto, la dimensión sería de $(n + 1)$ ciudades y su resolución consistiría en minimizar c_{ij} , que es equivalente a la distancia en el TSP original.

2.1.4 Secuenciación de trabajos

Este problema consiste en secuenciar n trabajos en una misma máquina, donde el cambio de un trabajo a otro implica un tiempo de preparación c_{ij} . Para lo que se trata de minimizar la suma total de tiempos de preparación, de manera que se reduzcan los tiempos muertos que no aportan valor al producto final. Dada una permutación de S con un determinado orden de n trabajos, el tiempo para realizar todos los trabajos será:

$$\sum_{i=0}^n (c_{iS(i)} + p_{iS(i)}) = \sum_{i=0}^n c_{iS(i)} + \sum_{i=0}^n p_j \quad i, j = 1 \dots n \quad (12)$$

En la práctica, esta aplicación del problema suele seguir la variante asimétrica del TSP, ya que no es lo mismo pasar del trabajo A al B, que del B al A. Un caso práctico sería que para realizar el trabajo A se requieren altas temperaturas mientras que, para el B bajas. Si el tiempo de calentar es diferente al de enfriar, el resultado sería un TSP asimétrico.

2.1.5 Reducción de la dispersión de pedidos en la industria del vidrio

Otra aplicación del TSP interesante y parecida a la anterior es la propuesta por Madsen [26]. En las industrias que emplean láminas grandes para ser cortadas (como es en la industria del vidrio) y formar otras piezas más pequeñas, que serán las que compre el cliente, se dan principalmente dos tipos de problemas. El primero es la optimización del espacio de la lámina, con el fin de evitar desperdicios. En la industria del vidrio el proceso de corte tiene tres etapas, la primera es la división de la lámina en varias secciones, la segunda es cortar transversalmente la lámina en varias partes y, por último, cada parte es reducida por uno ser sus laterales, de manera que se hayan definido las tres dimensiones de la pieza, como se muestra en la Figura 10. El segundo problema, y menos conocido, es evitar que el

pedido de un cliente sea dividido en varias láminas. Es decir, si un cliente hace un pedido de varias piezas, es posible que no todas se hagan en la misma lámina y que por optimizar los desperdicios, las piezas estén dispersas en el conjunto de láminas. Este problema adquiere mucha importancia en la industria del vidrio, ya que es muy difícil almacenar las piezas. No pueden ser almacenadas en grandes pilas, sino que deben ser almacenadas individualmente, lo que supone un alto coste de tiempo y espacio. Por lo tanto, reducir el número de láminas entre las que se distribuye un pedido es fundamental.

Ambos objetivos no tienen por qué ser contradictorios. Se puede primero optimizar la colocación de las piezas de todos los pedidos en el conjunto de láminas y en un segundo momento, secuenciar las láminas para reducir la distribución de piezas de un mismo pedido. Esto es, que si el pedido k tiene piezas en las láminas i y j , éstas deberían fabricarse consecutivamente para que el tiempo que esté el pedido k fabricándose sea el mínimo posible.

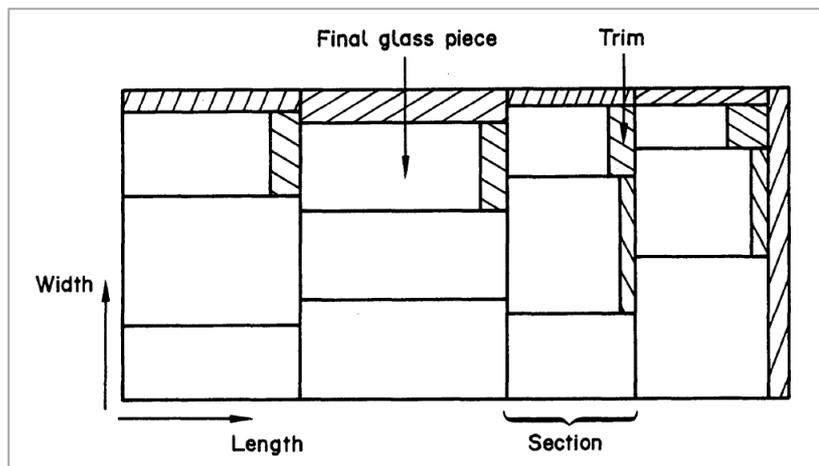


Figura 10: Ejemplo típico de un mosaico de corte para una sola lámina en la industria del vidrio. [26]

A la problemática de la secuenciación se le puede aplicar el TSP. En el problema se intenta que cada vez que la fabricación de un pedido ha comenzado, este no sea interrumpido. Esto se traduce en reducir las discontinuidades que se producen cuando la pieza de un pedido se fabrica en una lámina y en la siguiente no se hace ninguna pieza perteneciente al mismo pedido.

Se define como c_{ij} el número de pedidos en la lámina i que no se fabrican en la lámina j . Se define x_{ij} como una variable de decisión que vale 1 si la lámina j se fabrica inmediatamente después de la lámina i .

Ejemplo 2.1: Se tienen cuatro láminas entre las que se distribuyen siete pedidos:

| | | Pedidos asociados a cada lámina | | | |
|-----------|----|---------------------------------|---|---|---|
| Lámina nº | 1: | 1 | 3 | 4 | 6 |
| | 2: | 1 | 2 | 4 | 6 |
| | 3: | 1 | 2 | 5 | 7 |
| | 4: | 2 | 3 | 5 | 7 |

Según esta distribución la matriz de costes debería quedar de la siguiente manera:

$$C = \begin{bmatrix} \infty & 1 & 3 & 3 \\ 1 & \infty & 2 & 3 \\ 3 & 2 & \infty & 1 \\ 3 & 3 & 1 & \infty \end{bmatrix}$$

De manera que $c_{23} = 2$ indica que dos pedidos de la lámina 2 no continúan fabricándose cuando la lámina 3 se ha producido inmediatamente después de la dos, concretamente los trabajos son el 4 y el 6.

Para la secuencia $S = \{1,2,3,4\}$ se tiene que:

| | | | | |
|-----------|---|---|---|---|
| Lámina 1: | 1 | 3 | 4 | 6 |
| Lámina 2: | 1 | 2 | 4 | 6 |
| Lámina 3: | 1 | 2 | 5 | 7 |
| Lámina 4: | 2 | 3 | 5 | 7 |

El cuadrado significa que el pedido k ha sido interrumpido tras el cambio de lámina. Para este ejemplo han sido cuatro las veces que se han interrumpido pedidos, aunque solo uno sufre una discontinuidad, el 3.

Aunque la matriz c_{ij} recoja el número de veces que un pedido no se fábrica en la siguiente lámina, N_0 , y no el número de discontinuidades que un pedido sufre, N_D , es lo mismo minimizar ambos números:

$$N_0 = N_D + \text{número de pedidos} - 1$$

Por lo tanto, el problema de reducción de dispersión de pedidos se construirá cambiando la función objetivo del TSP (ecuación (2)) y sustituyéndola por las (13) y (14):

$$\text{Minimizar: } N_0 \tag{13}$$

$$N_0 = \sum_{i=1}^n \sum_{j=1, j \neq i}^n c_{ij} x_{ij} \quad i, j = 1, \dots, n \tag{14}$$

2.1.6 Generación de listas de reproducción según similitudes

Un enfoque interesante del TSP es el usado por Phole, Pampalk y Widmer [27], el problema consiste en ordenar una lista de reproducción según las similitudes musicales entre canciones. Es decir, en un móvil se tienen n canciones y en el momento de escuchar las canciones, cuando se acabe una, se quiere comenzar con otra que sea parecida musicalmente, de manera que el cambio no sea brusco. En la Figura 11 se puede ver como resultaría el modulador de la lista de reproducción. Girando la rueda se seleccionaría una canción, donde según la longitud del recorrido de la rueda, la diferencia con la canción actual será mayor o menor. Es decir, la rueda se mueve por la secuencia óptima o la que más se acerque al óptimo y va seleccionando canciones en función de su posición. Para que el cambio entre canción y canción no sea brusco, la ‘distancia’ que las separas debe ser lo mínimo, entendiéndose distancia por el parecido musical entre canción y canción.



Figura 11: ‘Traveler Sound Player’, regulador de canciones.

Para este problema la formulación es idéntica a la del TSP, lo novedoso es la forma en la que se construye la matriz de distancias c_{ij} . A cada canción (ciudad) n se le asocia un MFCC (Mel Frequency Cepstral Coefficients, en español: Coeficientes Cepstrales en las Frecuencias de Mel), coeficientes que representan el habla según la percepción auditiva humana. Una vez hecho esto, se representa los MFCCs en un modelo de mezcla gaussiana (GMM). Un GMM es un modelo probabilístico que asigna a cada observación un clúster, maximizando la probabilidad *a posteriori* de que el componente pertenezca a dicho clúster. Partiendo del GMM, se calcula la distancia en el gráfico entre la canción i y la canción j , y este valor se asigna a la matriz c_{ij} .

2.2 Variantes del TSP

En este apartado se verán seis variantes del TSP. Algunas se obtienen realizando pequeños cambios en el problema original, mientras que otras tienen un poco más de complejidad.

2.2.1 Minimizar el camino hamiltoniano

El problema original del camino hamiltoniano consiste en encontrar un ciclo en un conjunto de grados que incluya a todos los nodos del mismo; llamado así en referencia al inventor del juego “The Icosian Game”, explicado en la introducción del apartado 2. Este se puede considerar una variante del TSP donde la distancia c_{ij} de los arcos adyacentes a un nodo escogido es 0 y para aquellos no incluidos en el ciclo es 1, de manera que se penaliza la exclusión de nodos para encontrar un camino hamiltoniano.



Figura 12: “The Icosian Game”. Fuente: Royal Irish Academy Library en Twitter.com

Como muestran Lawler et al. [28] existen otras modificaciones a este problema. Una variante de este consistiría en quitar la restricción de acabar y termina siempre en el mismo punto. Lo cual se podría solucionar añadiendo una columna y filas $(n + 1)$ -ésima de valor 0 a la matriz de distancias c_{ij} . Si en cambio, se quisiera comenzar en un nodo determinado s y terminar en otro t , habría que sustituir las ciudades s y t por otra con ciudad u . De esta forma el número de ciudades sería igual a $(n - 1)$, quedando la nueva matriz de distancias c'_{ij} de la siguiente manera:

$$c'_{ij} = \begin{cases} c_{sj} & \text{si } i = u \\ c_{it} & \text{si } j = u \\ c_{ij} & \text{en otro caso} \end{cases} \quad i, j = 1 \dots (n - 1) \quad ()$$

De la misma forma, si se quisiera empezar en la ciudad s y acabar en cualquier otra, solo habría que poner a 0 las entradas del nodo s , es decir toda la columna s de la matriz c_{ij} . Por último, también

se podría hacer lo contrario, es decir, impedir que s sea el punto de inicio o final. Para ello, habría que crear una ciudad copia, que será llamada t , de manera que, en la matriz de distancias, c_{is} y c_{tj} fueran números lo suficientemente grandes como para que nunca fueran incluidos por la función objetivo.

2.2.2 TSP con ciudades repetidas

Otra variante simple del TSP sería incluyendo la posibilidad de poder visitar una ciudad más de una vez, en lugar de una única sola vez como plantea el problema original. Para ello, se sustituye la matriz c_{ij} por la matriz c'_{ij} , donde:

c'_{ij} = la distancia más corta del camino entre los nodos i y j (incluyendo en estos caminos otros posibles nodos).

Para el cálculo de c'_{ij} se puede usar el algoritmo de Floyd-Warshall [29], que consiste en la comparación de las distancias de la matriz c_{ij} para obtener el camino más corto, aunque sea pasando por otros vértices. Este algoritmo presenta un orden de complejidad $O(n^3)$ y su pseudo código es el siguiente:

```
FloydWarshall()
for k = 1 to n do
  for i=1 to n do
    for j=1 to n do
      d(i,j) = min{d(i,j), d(i,k) + d(k,j)}
    end
  end
end
```

Figura 13. Pseudocódigo del algoritmo de Floyd-Warshall. Fuente: <https://resources.mpi-inf.mpg.de/>

Este tipo de versión del problema del viajante es fundamental para aquellos grafos en los que para visitar un nodo y salir de él es estrictamente necesario pasar por otro al menos dos veces. En la Figura 14 se observa un ejemplo, donde para llegar al nodo A es necesario pasar antes por el nodo B.

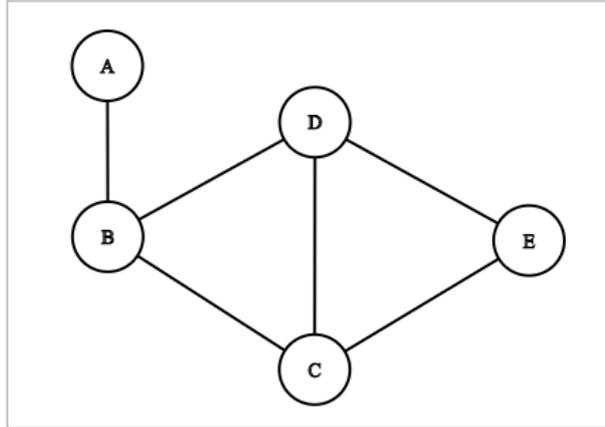


Figura 14: Grafo en el cual el viajante tendrá que visitar el nodo B al menos dos veces. Fuente: <http://web.mit.edu/>

2.2.3 El cuello de botella TSP

En este problema se intenta minimizar la mayor distancia entre nodos, la versión generalizada fue definida por Garkinfel y Gilbert [30]. Esto en la práctica es útil cuando se quiere regular el flujo de una estación de trabajo, es decir, reducir su tiempo de ciclo. En el apartado 2.1.4 se puede ver como el TSP se aplica a una estación de trabajo. Si hubiera varias estaciones concatenadas, el ritmo lo marcaría el trabajo que más durara, por tanto, si se quiere reducir la cadencia se debe minimizar el trabajo más duradero. Para ello, habría que transformar la función objetivo del TSP (ecuación (2)) y añadir una nueva restricción:

$$\min t \quad (15)$$

$$x_{ij}c_{ij} \leq t \quad i,j=1\dots n \quad (16)$$

2.2.4 TSP dependiente del tiempo

Para esta variante se tienen en cuenta la posibilidad de que la distancia o el coste de ir del nodo i al nodo j es variante con el tiempo t . Para el cual se definen dos nuevas matrices: c_{ijt} , que indica el coste de ir del nodo i al nodo j en el intervalo o periodo de tiempo t , y x_{ijt} , que es una variable binaria que vale 1 si el camino el viajante va desde la ciudad i hasta la ciudad j en el periodo t . La dimensión del conjunto de periodos es igual al número de ciudades n . Por lo que el problema final quedaría de la siguiente manera:

Minimizar:

$$\sum_{i=1}^n \sum_{j=1, j \neq i}^n \sum_{t=1}^n c_{ij} x_{ijt} \quad (17)$$

Sujeto a:

$$\sum_{t=1}^n \sum_{i=1, i \neq j}^n x_{ijt} = 1 \quad j = 1 \dots n \quad (18)$$

$$\sum_{t=1}^n \sum_{j=1, j \neq i}^n x_{ijt} = 1 \quad i = 1 \dots n \quad (19)$$

$$\sum_{t=1}^n \sum_{i=1}^n \sum_{j=1, j \neq i}^n x_{ijt} = 1 \quad t = 1 \dots n \quad (20)$$

$$\sum_{t=2}^n \sum_{j=1}^n t x_{ijt} - \sum_{t=1}^n \sum_{j=1}^n t x_{ijt} = 1 \quad i = 2 \dots n \quad (21)$$

$$x_{ijt} \in \{1, 0\} \quad i, j, t = 1 \dots n \quad (22)$$

$$c_{ijt} > 0 \quad i, j, t = 1 \dots n \quad (23)$$

La ecuación (23) obliga a que cuando se entre en el período t en la ciudad i siempre se salga de ella en el período $t + 1$, para aquellos vértices que cumplan. $2 \leq i \leq n$. Además, también fuerza al algoritmo a no introducir en la solución ningún *subtour*. Por ejemplo, si se tuviera un *subtour* donde la ciudad $i' \neq 1$ fuera la inicial y por tanto la final del *subtour*, el viajante saldría de esa ciudad en el período t' y entraría en ella en el período $t' + \delta$, donde $\delta \geq 1$, por lo que aplicando la inecuación (23) en el lado izquierdo se tendría $-\delta$. Por lo que con dicha restricción se obligaría a realizar siempre un *tour*. Esta formulación fue presentada por Fox et al. en 1980 [31].

Otra formulación del problema, dada por Picard y Queyranne [32], deriva del problema de secuenciar los trabajos de una sola máquina, donde el aparato parte de un estado inicial 0 que coincidirá con un estado final $n + 1$. Cada uno de los estados equivale a una configuración que permite la realización de cada trabajo. En este caso, la variable binaria x_{ijt} representa el cambio del estado i al estado j en el periodo t y la constante c_{ijt} el tiempo de preparación de la maquina desde el estado i al

estado j en el periodo t . El problema quedaría de la siguiente manera:

Minimizar:

$$\sum_{i=1}^n c_{0j0} x_{0j0} + \sum_{t=1}^{n-1} \sum_{i=1}^n \sum_{j=1}^n c_{ijt} x_{ijt} + \sum_{j=1}^n c_{j,n+1,n} x_{j,n+1,n} \quad (24)$$

Sujeto a:

$$\sum_{i=1}^n x_{0j0} = 1 \quad j = 1 \dots n \quad (25)$$

$$\sum_{i=1}^n x_{ijt} = \sum_{k=1}^n x_{jk,t+1} \quad t = 1 \dots n - 2 \quad (26)$$

$$j = 1 \dots n$$

$$\sum_{i=1}^n x_{ij,n-1} = x_{j,n+1,n} \quad j = 1 \dots n \quad (27)$$

$$x_{0j0} + \sum_{i=1}^n \sum_{t=1}^n x_{ijt} = 1 \quad i = 2 \dots n \quad (28)$$

$$x_{ijt} \in \{0,1\} \quad i, j, t = 1 \dots n \quad (29)$$

$$c_{ijt} > 0 \quad i, j, t = 1 \dots n \quad (30)$$

2.2.5 TSP basados en el beneficio

Existe un conjunto de variantes del TSP presentadas por Feillet et al. [33] que buscan un equilibrio entre el beneficio de visitar cada nodo y el coste que esto supone; no siendo obligatorio visitar todos y cada uno de los nodos. Es decir, por un lado, se buscará maximizar el beneficio y por otro minimizar los costes. Para todos los problemas que se encuentran en esta variante se plantean las siguientes restricciones:

$$\sum_{a \in \delta(\{v_i\})} x_a = 2y_i \quad v_i \in N \quad (31)$$

$$y_1 = 1 \quad (32)$$

$$x_a \in \{0,1\} \quad x_a \in A \quad (33)$$

$$y_i \in \{0,1\} \qquad y_i \in N \qquad (34)$$

Dónde para cada subconjunto de nodos S de N , se define $\delta(N)$ como el conjunto de arcos que están conectados por un extremo a S y por el otro a $V \setminus S$.

En cuanto a la función objetivo, esta varía según el problema que se quiere resolver, aunque en algunos casos también será necesario añadir alguna variable y restricción más. Se dan principalmente tres casos:

- (1) PTP (Profitable Tour Problem): consiste en la persecución de dos objetivos: minimizar el gasto y aumentar el beneficio. Su función objetivo consta del sumatorio del coste menos el sumatorio del beneficio.

Minimizar:

$$\sum_{a \in A} c_a x_a - \sum_{v_i \in N} p_i y_i \qquad (35)$$

Donde p_i es una constante que representa el beneficio que aporta visitar el nodo i .

- (2) OP (Orienteering Problem): en este caso, el coste objetivo se representa como una restricción y no como una función a minimizar, de tal manera que el objetivo es maximizar el beneficio sin llegar nunca a superar un coste límite c_{max} . Se puede entender esta restricción como una de capacidad, muy común en los problemas de ruta de vehículos (VRP), nombrado en el apartado 2.1.1. La formulación del OP quedaría de la siguiente manera:

Maximizar:

$$\sum_{v_i \in N} p_i y_i \qquad (36)$$

Sujeto a:

$$\sum_{a \in A} c_a x_a \leq c_{max} \qquad (37)$$

A estas restricciones se le tendría que añadir las principales de este tipo de problema, ecuaciones (31), (32), (33) y (34).

- (3) PCTS (Prize-Collecting TSP) Este problema sería justo el contrario que el anterior, en

cuanto a función objetivo se refiere ya que, en vez de maximizar el beneficio, lo establece como una función y en su lugar minimiza el coste. De esta manera el problema quedaría:

Maximizar:

$$\sum_{a \in A} c_a x_a \quad (38)$$

Sujeto a:

$$\sum_{v_i \in N} p_i y_i \geq p_{min} \quad (39)$$

Al igual que en el caso anterior, a estas restricciones se le tendría que añadir las principales de este tipo de problema, ecuaciones (31), (32), (33) y (34).

2.3 Problemas genéricos relacionados con el TSP

En este apartado se tratarán varios problemas estándares o generales de optimización combinatoria que guardan relación con el TSP.

2.3.1 El problema de la asignación

Conocido por su nombre original en inglés, *The assignment problem*, guarda mucha similitud con el TSP, sobre todo en la formulación. El problema surge con la necesidad de asignar operarios a las máquinas. En él se tiene un conjunto de personas N que debe ser asignado a un conjunto de máquinas M , con un coste c_{ij} si el operario i es asignado a la máquina j y una variable x_{ij} binaria que toma valor uno si el operario i trabaja en la máquina j y cero en caso contrario. La similitud con el problema del viajante se da cuando el conjunto N y M son del mismo tamaño, es decir, $|M| = |N|$. El objetivo es el mismo, para el caso que se está comparando, minimizar la suma total de las distancias o costes, formulándose el problema de la siguiente manera:

Minimizar:

$$\sum_{i=1}^n \sum_{j=1, j \neq i}^n c_{ij} x_{ij} \quad (40)$$

Sujeto a:

$$\sum_{i=1, i \neq j}^n x_{ij} = 1 \quad j = 1 \dots n \quad (41)$$

$$\sum_{j=1, j \neq i}^n x_{ij} = 1 \quad i = 1 \dots n \quad (42)$$

Si se compara con la formulación original del problema del viajante, dada al comienzo del apartado 2, se observa, que las restricciones que impiden la generación de *subtours*, se han eliminado, lo que supone una relajación del problema original. En la práctica es perfectamente realizable un *subtour* para el problema de la asignación. Tómese, por ejemplo, la Figura 15, donde para un problema de cinco nodos existe una solución con dos *subtours*. La interpretación de esta solución para el problema de la asignación sería la siguiente:

- En el *subtour* de la izquierda indica que el operario 1 está asignado a la máquina 3, que el operario 3 lo está a la máquina 5 y que el operario 5 está asignado a la máquina 1.

- En el subtour de la derecha se puede ver que a máquina 2 se le asigna el operario 4 y el operario 2 es asignado a la máquina 4.



Figura 15: *Subtours* para un problema de 5 ciudades. Fuente: chegg.com

Claramente, esta solución en el problema del viajante carecería de sentido físico.

2.3.2 La restricción del *subtour*

Para poder forzar al problema anterior a que no se generen *subtours* se pueden introducir diferentes restricciones, por ejemplo, la desarrollada por Dantzig et al. [34]:

$$\sum_{i \in S} \sum_{j \in S} x_{ij} \leq |S| - 1 \quad (43)$$

$$\sum_{i \in S} \sum_{j \notin S} x_{ij} \geq 1 \quad (44)$$

Donde se define S como un subconjunto no vacío de $N = \{1, \dots, n\}$ y $|S|$ el cardinal del subconjunto S . Para el ejemplo de la Figura 15, se puede observar cómo ambos *subtours* incumplen las restricciones (43) y (44), donde $S = \{1, 3, 5\}$ o $\{2, 4\}$. En el primer subconjunto, el cardinal es 3, por lo que la restricción (43) se incumple, ya que el sumatorio de todos los arcos que unen los vértices del subconjunto también es igual a 3; por lo que se incumple la inecuación; esto mismo ocurre para el segundo subconjunto. En la restricción (44), en el lado izquierdo de la inecuación, se puede observar que el sumatorio cuenta todos los arcos salientes del subconjunto S , por lo que un número mayor que 0 en el resultado del sumatorio confirma la inexistencia de *subtours* en el subconjunto evaluado. No obstante, el principal problema de estas restricciones es su elevado número, ya que será igual al cardinal del conjunto potencia de N menos dos (excluyen el subconjunto vacío de N y el subconjunto igual a N). Es decir, el número de restricciones sería igual a $2^{|N|} - 2$, donde $|N| = n$.

Una variación más compacta de estas restricciones es la propuesta por Miller, Tucker & Zemlin [23], que es la presentada en el apartado 2, ecuación (5).

$$u_i - u_j + nx_{ij} < n - 1 \quad 2 \leq i \neq j \leq n \quad (5)$$

En ella se toma arbitrariamente el vértice 1 como punto de partida, de ahí que i y j estén comprendidos entre 2 y n . Por otro lado, u_i y u_j son números reales arbitrarios que impedirán la creación de *subtours*. Si se vuelve a tomar como ejemplo el *subtour* izquierdo de la Figura 15, los valores de u_i quedarían: $u_2 = 0$ y $u_4 = 1$ por lo que para ir del vértice 4 al 2 la inecuación se incumpliría. Se comprueba entonces, que el único vértice que puede ser el inicial y final es el 1 y que para que se cumpliera la inecuación, el vértice 2 y 4 deberían estar en el primer subconjunto. Es importante recalcar que el número de restricciones disminuye a $(n - 1)^2$.

Una variante del TSP, presentada también por Miller et al. Miller y que aprovecha la ecuación anterior es aquella en la que se permiten *subtours* que parten todos del vértice 1 y para los que hay un límite máximo de p ciudades que puede abarcar. Para ello la ecuación quedaría del siguiente modo:

$$u_i - u_j + px_{ij} \leq p - 1 \quad 2 \leq i \neq j \leq n \quad (5)$$

Por otra parte, sería necesario modificar las restricciones (3) y (4) para que desde la ciudad 1 pudieran partir diferentes *subtours*. Por lo que las restricciones que las sustituirían quedarían de la siguiente manera, respectivamente.

$$\sum_{i=1, i \neq j}^n x_{ij} = 1 \quad j = 2 \dots n \quad (45)$$

$$\sum_{j=1, j \neq i}^n x_{ij} = 1 \quad i = 2 \dots n \quad (46)$$

2.3.3 El problema de la asignación cuadrática

Este problema consiste en situar n instalaciones o departamentos en n posiciones. Fue presentado por primera vez por Koopmans y Beckmann [35], y para su planteamiento se deben tener en cuenta tres matrices que recogen las características del problema:

- La matriz de flujo f_{ij} , que define el flujo entre una instalación h y otra k , en la práctica este flujo puede corresponder a unidades de materiales, personas, electrones, precio, etc.
- La matriz de distancias d_{hk} , que define la distancia que hay entre la posición h y la k .
- La matriz de costes de instalación c_{ijhk} , que recoge el coste total de asignar la instalación i

en la posición k y la instalación j en la posición h .

De esta forma las restricciones del problema y la función objetivo quedarían de la siguiente manera:

Minimizar:

$$\sum_{i=1}^n \sum_{j=1}^n \sum_{h=1}^n \sum_{k=1}^n c_{ijhk} x_{ih} x_{jk} \quad (47)$$

Sujeto a:

$$\sum_{h=1}^n x_{ih} = 1 \quad i = 1 \dots n \quad (48)$$

$$\sum_{i=1}^n x_{ih} = 1 \quad h = 1 \dots n \quad (49)$$

$$c_{ijhk} = f_{ij} d_{hk} \quad i, j, h, k = 1 \dots n \quad (50)$$

$$x_{ih} \in \{0,1\} \quad i, h = 1 \dots n \quad (51)$$

Donde x_{ih} es una variable binaria que valdrá 1 en caso de que la instalación j se instale en la posición h y 0 en caso contrario.

En la práctica este problema se puede utilizar, por ejemplo, para el diseño de aeropuertos en los que para los transbordos se querrá reducir la distancia total recorrida entre terminales por la suma total de pasajeros. En este caso los pasajeros que van de la terminal A a la B serán recogidos en la matriz de flujo, mientras que la distancia entre la posición E y la posición W, lugar donde se sitúan las terminales respectivamente, será recogida en la matriz de distancia. Un ejemplo parecido se puede utilizar para situar los departamentos de un hospital, donde los pacientes tendrán que pasar de un médico a otro según su patología. Aunque también puede aplicarse para el diseño de una fábrica, en la que se reducirá el tiempo de transporte de productos semielaborados o también para el diseño de circuitos electrónicos, en el que el precio de los conectores varía según el dispositivo que haya que conectar.

2.3.4 El problema del camino más largo

En el apartado 2.2.2 se vio como dado un grafo G que contenía un conjunto de nodos N , unidos entre sí por un conjunto de arcos A , se podía calcular la distancia mínima entre dos vértices mediante el algoritmo de Floyd y Warshall, cuyo orden de complejidad era de $O(n^3)$. En el problema que se discute en este apartado el objetivo es el opuesto, se pretende buscar el camino más largo para llegar

del nodo i al nodo j . Aunque, pueda parecer trivial o igual que complejo que el problema del camino más corto, en realidad se trata de un algoritmo NP-hard. Esto se puede comprobar reduciendo el problema a la búsqueda del camino crítico donde cada arco tiene el mismo valor (1, por ejemplo), por lo que el problema se reduce a la búsqueda de un camino hamiltoniano en el grafo G . El simple problema de decisión, es decir, indicar si existe o no un camino hamiltoniano con al menos k vértices en dicho grafo, siendo $k \leq n$, es ya de por sí un problema NP-hard. [36], lo que convierte al problema del camino más largo también con la misma complejidad.

Por otra parte, es claro, que si el G es un grafo completo (Figura 16, muestra un grafo completo en el que cada vértice está conectado a todos los demás mediante un arco), el presente problema es un TSP, dónde en vez de buscar el camino mínimo se busca el máximo. Una correcta transformación del TSP en el problema del camino más largo consistiría en cambiar la matriz de costes c_{ij} en la matriz c'_{ij} , cambiar el objetivo de la función a ‘maximizar’ y aplicar la variante del problema del TSP en la que se quiere ir del nodo s al nodo t (explicado en el apartado 2.2.1). La transformación consistirá en $c'_{ij} = M - c_{ij}$, donde M es lo suficientemente grande como para que no existan distancias negativas. [37]

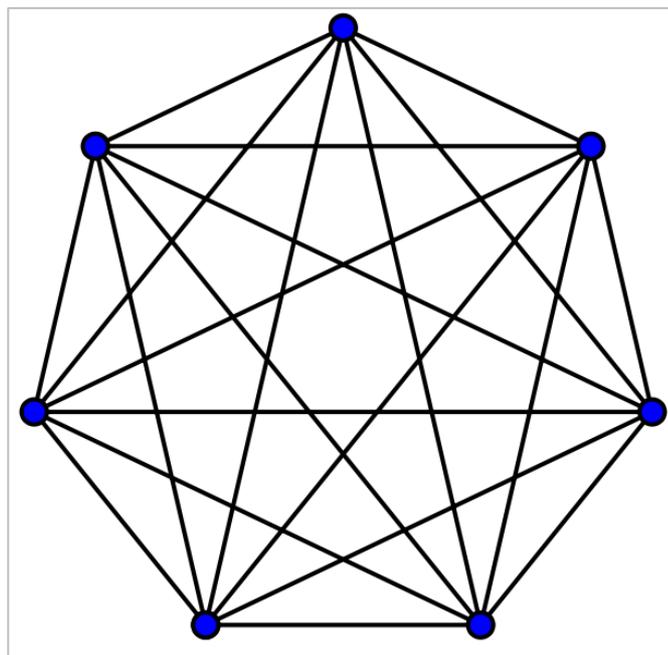


Figura 16: Grafo completo. Fuente Wikipedia Commons.

2.3.5 Árbol de cobertura mínimo

Un árbol de cobertura mínimo consiste en un subconjunto de arcos conectados y con un peso asociado a cada uno en un grafo no dirigido y conexo que conecta a todos los nodos, de manera que

no se genere ningún ciclo y que la suma de todos los pesos de los arcos que componen el subconjunto sea mínima. Es decir, encontrar un árbol de cobertura en el grafo G cuya suma de pesos sea la mínima posible.

El problema de encontrar un árbol de cobertura se puede asimilar al de encontrar un camino hamiltoniano. De hecho, todo camino hamiltoniano es un árbol de cobertura y todo árbol de cobertura es un camino hamiltoniano en el que se permite que un nodo tenga más de dos conexiones con otros nodos. De esta manera, si encontrar un árbol de cobertura es una relajación del camino hamiltoniano, también encontrar un árbol de cobertura mínimo es una relajación del problema del viajante, al que se le quita la misma restricción que para el caso anterior.

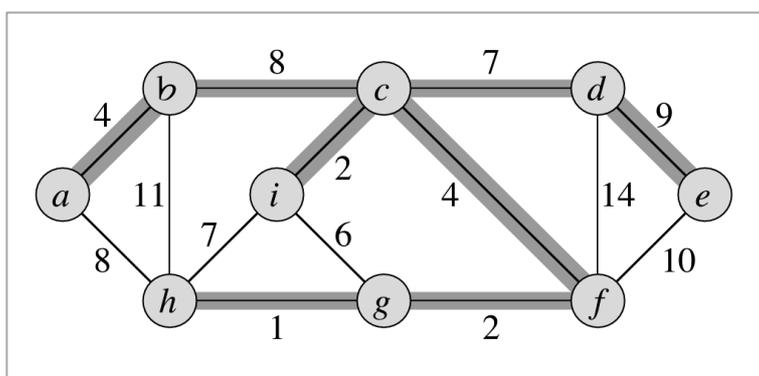


Figura 17: Ejemplo de mínimo árbol de cobertura. [38]

Para entender mejor este problema, se puede observar la Figura 17, en la que se ve la solución al problema del árbol de cobertura mínimo, con los respectivos pesos de cada arco. El peso total de la solución es de 37 y se hace un solo cambio se comprueba que la solución no es única: si se elimina el arco (b, c) y se sustituye por el arco (a, h) la solución al problema sigue valiendo lo mismo.

2.3.6 El problema del cartero chino

El matemático chino Mei-Ko Kawn [39] se hizo la pregunta de dado conjunto de calles que deben recibir un servicio postal y un cartero que debe hacer una ruta pasando por todas ellas y acabar en el mismo punto de partida, se debe buscar la ruta mínima. Es decir, dado un grafo $G(V, A)$, siendo A el conjunto de los arcos que unen los vértices, se deben recorrer todos los arcos pertenecientes a A . Este problema en la práctica se puede aplicar a más situaciones. Por ejemplo, a la recogida de nieve en las calles, a la recogida de basura o limpieza de calles, a la inspección de tuberías o cables, etc.

El problema del cartero chino (CPP, Chinese Postman Problem) y sus variantes tienen su origen en el análisis de Euler [40] a un puzzle muy popular en su época conocido como El problema de los puentes de Königsberg. Consistía en atravesar todos los siete puentes de Königsberg (Figura 18) solo y exclusivamente una vez. Euler resolvió el problema (aunque no llegó a demostrarlo del todo)

afrimando que un ciclo existe en un grafo si y solo sí cada vértice está conectado a un número de aristas par. Esto se pueden entender lógicamente pensando que, al llegar a un vértice por un arco, debe existir otro por el que salir. De esta manera, se define un ciclo que atraviesa todos los arcos de un nodo exactamente una sola vez como Ciclo euleriano. Cuando cada arco tiene un peso y el objetivo es encontrar un ciclo euleriano relajado con el camino mínimo, el problema es el CPP, aunque en éste la restricción de visitar una sola vez cada arco se relaja a visitarlo al menos una.

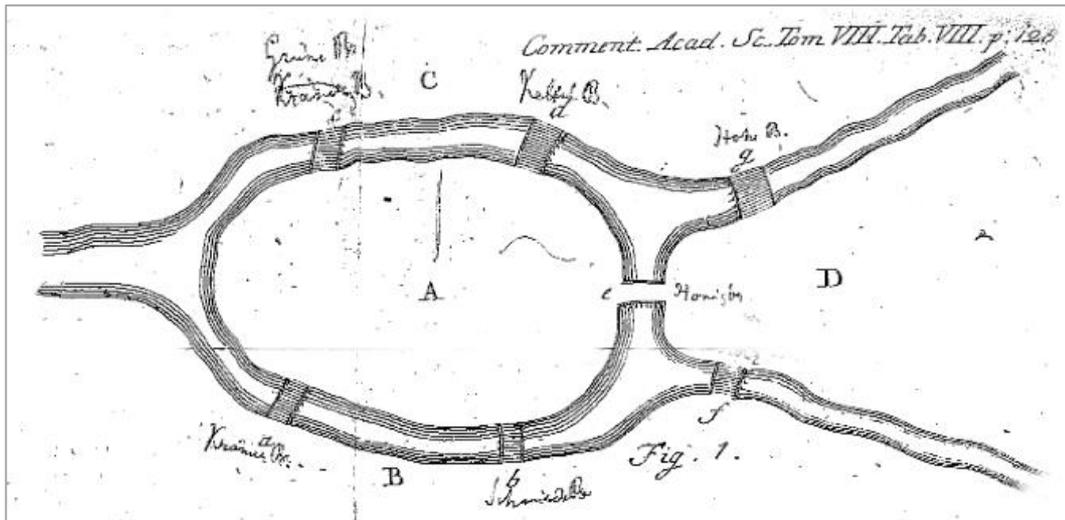


Figura 18: Problema de los siete puentes de Königsberg [40].

El problema del cartero chino se puede formular matemáticamente como, dado un grafo $G(V, A)$, donde v es un conjunto de vértices n , A es un conjunto de arcos que une dichos vértices, y cada arco (i, j) que conecta el vértice i con el vértice j tiene un coste asociado c_{ij} mayor o igual que cero, x_{ij} es el número de veces que un arco se emplea en la ruta para conectar todos los arcos A , se intenta encontrar la ruta óptima que minimice la distancia total recorrida.

Minimizar:

$$\sum_{i=1}^n \sum_{j=1, j \neq i}^n c_{ij} x_{ij} \quad (52)$$

Sujeto a:

$$\sum_{i=1}^n x_{ik} - \sum_{j=1}^n x_{kj} = 0 \quad k = 1 \dots n \quad (53)$$

$$x_{ij} + x_{ji} \geq 1 \quad i, j = 1 \dots n \quad (54)$$

$$x_{ij} \geq 0 \quad i, j = 1 \dots n \quad (55)$$

2.4 Algoritmos brutos y complejidad del TSP

Para solucionar problemas de optimización combinatoria es fundamental el uso de algoritmos. Entendiéndose estos como un conjunto de pasos que se van ejecutando uno detrás de otro y que resuelven un problema según unos datos de partida. En este apartado se verán dos algoritmos que resuelven el TSP, el primero para el TSP puro y el segundo para un caso especial de este, y otro que resuelve el problema del camino más corto, visto en el apartado 2.2.2. Por último, se hará un breve estudio de la complejidad de los algoritmos.

2.4.1 Algoritmo de fuerza bruta

El problema del TSP se puede definir como una pregunta y un conjunto de datos necesarios para responderla. Por lo que el problema podría ser representado de la siguiente manera:

DATOS: Número entero $n \geq 3$, matriz $n \times n$ $C = (c_{ij})$,
donde cada c_{ij} es un número real mayor o igual que 0.

PREGUNTA: ¿Qué permutación π de los números enteros
que van de 1 a n minimiza la suma de $\sum_{i=1}^n c_{i\pi(i)}$?

De esta forma, un algoritmo para resolver el TSP, debe recibir como datos de entrada el número n y la matriz C y deberá devolver como dato de salida un camino mínimo que una todos los puntos, de manera que se pueden definir como un a permutación cíclica.

Un algoritmo sencillo, aunque extremadamente ineficiente, es el de fuerza bruta. Consiste en calcular la longitud de todos los caminos posibles, es decir, de todas las permutaciones cíclicas del conjunto de nodos N , compararlos y obtener así el mínimo de todos ellos, siendo esta la solución final. El pseudo código de este algoritmo se muestra en la Figura 19.

* Entrada: Número entero $n \geq 3$, matriz $n \times n$ $C = (c_{ij})$, donde cada c_{ij} es un número real mayor o igual que 0.

* Salida: El camino más corto que uno todos los nodos N , que empieza en 1 y acaba en n .

comienzo

for todas las permutaciones cíclicas π de $\{1, \dots, n\}$ **do**

 coste = $\sum_{i=1}^n c_{i\pi(i)}$;

if coste < min **then**

 min = coste

 mejor_camino = π

finaliza if

finaliza for

salida mejor_camino

finaliza algoritmo

Figura 19: Pseudo código del algoritmo de la fuerza bruta.

2.4.2 Algoritmo del camino mínimo

Otro algoritmo de programación dinámica es el presentado en la Figura 20, que resolverá el problema de la ruta mínima, definido esquemáticamente de la siguiente manera:

DATOS: Número entero $n \geq 3$, matriz $n \times n$ $C = (c_{ij})$, donde cada c_{ij} es un número real mayor o igual que 0.

PREGUNTA: ¿Cuál es el camino más corto para ir de la ciudad 1 a la n ?

La principal restricción que se elimina es la necesidad de visitar todos y cada uno de los nodos. Por lo que para su resolución se definirán otras variables que permiten construir una solución de manera más eficiente. Se define $\text{coste}[j, i]$ como la distancia recorrida desde la ciudad 1 hasta la ciudad i , pasando por todos los nodos menores o iguales que j . Donde $i \neq 1$ y $j \geq 1$. Por otra parte, se hace uso en el algoritmo de una variante de la ecuación (57), en la que se busca el camino más corto desde la ciudad 1 hasta la ciudad i con un número de nodos menor o igual que j , como se muestra en la ecuación

$$\text{coste}[j, i] = \min \left\{ \text{coste}[j - 1, i], \min_{k \neq i, 1} \{ \text{coste}[j - 1, k] + c_{ki} \} \right\} \quad (56)$$

Para buscar la distancia mínima entre 1 e i se compara todos los caminos con un número de nodos menor o igual que $j - 1$ y se obtiene un camino cuya penúltima ciudad en visitar es k . Si el camino resultante es mejor que el actual, se actualiza el mejor camino actual con la solución recién obtenida, hecho esto se aumenta el número j y se vuelve a realizar el mismo procedimiento hasta que $j = n$.

```

* Entrada: Número entero  $n \geq 3$ , matriz  $n \times n$   $C = (c_{ij})$ , donde cada  $c_{ij}$  es un número
real mayor o igual que 0.
* Salida: El camino más corto que empieza en 1 y acaba en  $n$ .
begin
  for  $i = 2, 3, \dots, n$  do
     $\text{coste}[1, i] = c_{1i}$ ;
     $\text{mejor\_camino}[1, i] = (1, i)$ ;
  end
  for  $i = 2, 3, \dots, n$  do
    for  $j = 2, 3, \dots, n$  do
       $\text{coste}[j, i] = \min_{k \neq 1, i} \{ \text{coste}[j - 1, k] + c_{ki} \}$ ;
      se estable  $k$  como la ciudad para la que el coste es mínimo;
       $\text{mejor\_camino}[j, i] = (\text{mejor\_camino}[j - 1, k], i)$ 
    end
    if  $\text{coste}[j - 1, i] < \text{coste}[j, i]$  then
       $\text{coste}[j, i] = \text{coste}[j - 1, i]$ 
       $\text{mejor\_camino}[j, i] = \text{mejor\_camino}[j - 1, i]$ ;
    end
  end
  salida  $\text{mejor\_camino}[n - 1, n]$ 
end

```

Figura 20: Algoritmo que resuelve El problema del camino mínimo.

2.4.3 El TSP pequeño

Este método fue presentado por Gabovich [41] y consiste en resolver el TSP cuando la matriz de costes C de dimensión $n \times n$ es considerada pequeña. Se define matriz pequeña $n \times n$ si existen los

vectores de dimensión n , a y b y con valores distintos, tal que $c_{ij} = \min\{a_i, b_i\}$. De ahora en adelante se supondrá que los componentes del vector a cumplen la siguiente característica: $a_1 \leq a_2 \leq \dots \leq a_n$, de esta forma se define d_i como el i -ésimo número más pequeño del conjunto de todos valores $2n$ de a y b , entonces $D = \{d_1, d_2, \dots, d_n\}$. Por otro lado, se define $d = \sum_{i=1}^n d_i$. Teniendo en cuenta estas variables y condiciones, se puede resolver fácilmente el problema del viajante para una matriz pequeña C con un abanico de soluciones muy reducido. Los siguientes teoremas y demostraciones fueron resumidos y explicados por Lawler et al. [28].

Teorema 1 *Sea C una matriz pequeña con valores distintos. La longitud de la ruta mínima para C es d si y solo si, una de las siguientes condiciones se cumple.*

(Condición 1) Para cualquier ciudad i , tanto a_i como b_i , se encuentran en D .

(Condición 2) $D = \{a_1, a_2, \dots, a_n\}$.

(Condición 3) $D = \{b_1, b_2, \dots, b_n\}$.

Prueba Si la condición 1 se cumple, se plantea el siguiente procedimiento:

1. Se divide el conjunto de ciudades en cuatro conjuntos D_0, D_a, D_b y D_2 . En D_0 se incluyen aquellas ciudades o nodos para los que ni a_i , ni b_i se encuentran en D . En D_a se incluyen aquellas en las que a_i se encuentra en D . En D_b se incluyen aquellas en las que b_i se encuentra en D . Finalmente, en D_2 se incluyen aquellas en las que a_i y b_i se encuentran en D . Se puede observar fácilmente que $|D_0| = |D_2|$.
2. Se comienza a construir una ruta seleccionando una ciudad que se encuentre en D_2 .
3. Sin tener en cuenta el orden, se secuencian todas las ciudades de D_a .
4. Se selecciona cualquier ciudad de D_0 .
5. Sin tener en cuenta el orden, se secuencian todas las ciudades de D_b .
6. Se selecciona cualquier ciudad de D_2 .
7. Se selecciona cualquier ciudad de D_0 .
8. Se vuelve al punto 6 hasta que se hayan secuenciado todas las ciudades.

Se muestra a continuación un ejemplo de algoritmo para una matriz de costes de seis nodos con valores repetidos en los conjuntos a y b . Como se verá más adelante, esto no afecta a la resolución del problema.

Ejemplo 3.1 Se tiene $a = \{1,2,4,4,7,9\}$ y $b = \{3,3,6,7,1,5\}$, de manera que siendo la matriz C , $c_{ij} = \min\{a_i, b_j\}$ se tiene que:

$$C = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 2 & 2 & 2 & 2 & 1 & 2 \\ 3 & 3 & 4 & 4 & 1 & 4 \\ 3 & 3 & 4 & 4 & 1 & 4 \\ 3 & 3 & 6 & 7 & 1 & 5 \\ 3 & 3 & 6 & 7 & 1 & 5 \end{pmatrix}$$

Por otro lado, también se tiene que $D = \{a_1, b_5, a_2, b_1, b_2, a_3\} = \{1,1,2,3,3,4\}$, por lo tanto, los cuatros conjuntos en los que se divide el conjunto de ciudades N queda: $D_0 = \{4,6\}$, $D_2 = \{1,2\}$, $D_a = \{3\}$ y $D_b = \{5\}$. Nótese que $|D_0| = |D_2| = 2$.

Finalmente, comenzando por el punto 2 se obtiene la secuencia $\pi = (1,3,4,5,2,6)$, se verifica entonces que $c(\pi) = c_{13} + c_{34} + c_{45} + c_{52} + c_{26} + c_{61} = 1 + 4 + 1 + 3 + 2 + 3 = d = 14$.

Nótese, que en este ejemplo se ha hecho con valores de a y de b no distintos. Más adelante (en el Teorema 2) se verá que esto no influye en la obtención del valor óptimo d .

Para el caso en el que no se cumpla la condición 1, pero sí la 2 y la 3. El problema se convierte en un TSP constante, con coste para cualquier ruta igual a d .

Por otro lado, si ninguna de las tres condiciones anteriores se cumple es porque el conjunto D se compone por los siguientes elementos: $D = \{a_1, \dots, a_k, b_{k+1}, \dots, b_n\}$ de manera que $1 \leq k \leq n - 1$. En tal caso, si se afirmara que existe un camino con coste d y en el que todos los valores de a_i y b_j fueran distintos, sería imposible llevarlo a cabo, ya que, a cierto punto de la secuencia, un arco a_i sería inmediatamente seguido por b_i .

Se dice que un conjunto de costes $D = \{d_1, d_2, \dots, d_n\}$ domina a otro $D' = \{d'_1, d'_2, \dots, d'_n\}$ si existe cualquier permutación π , tal que $d_{\pi(i)} \geq d'_i$ para $i = 1, \dots, n$. Por otra parte, un conjunto de costes $D = \{d_1, d_2, \dots, d_n\}$ es factible si existe una ruta σ y una permutación π , tal que $c_{i\pi(i)} = d_{\sigma(i)}$ para $i = 1, \dots, n$. De esta manera el Teorema 1 se puede formular de otra manera.

Teorema 1' Sea C una matriz pequeña con valores distintos y sea $D' = \{d'_1, d'_2, \dots, d'_n\}$ un subconjunto de $\{a_1, \dots, a_n, b_1, \dots, b_n\}$, tal que no exista otro subconjunto al que domine y que sea

factible. Existe entonces, un subconjunto que usa los costes en D si y solo si, una de las siguientes tres condiciones se cumple.

(Condición 1') Para cualquier ciudad i , tanto a_i como b_i , se encuentran en D' .

(Condición 2') $D' = \{a_1, a_2, \dots, a_n\}$.

(Condición 3') $D' = \{b_1, b_2, \dots, b_n\}$.

En caso de que ninguna de las tres condiciones del Teorema 1 se cumpliera, se aplicaría el siguiente teorema.

Teorema 1 Sea C una matriz pequeña con valores distintos. La longitud del camino mínimo según el TSP para la matriz C puede ser uno de los siguientes valores: d , $d - d_n + d_{n+1}$ o $\min\{d - d_n + d_{n+2}, d - d_{n-1} + d_{n+1}\}$. Además, si el coste óptimo no es d , entonces será mayor que $d - d_n + d_{n+1}$ si y solo si, se cumple una de las siguientes condiciones:

(Condición 4) $k = 1; d_n = b_2; d_{n+1} = a_2$.

(Condición 5) $k = n - 1; d_n = a_{n-1}; d_{n+1} = b_{n-1}$

(Condición 6) $2 \leq k \leq n - 2$; o bien $(d_n = a_k$ y $d_{n+1} = b_k)$, o bien $(d_n = b_{k+1}$ y $d_{n+1} = a_{k+1})$.

Donde k es el índice más largo i , tal que $a_i \in D$.

Prueba Partiendo del caso en el que el Teorema 1 se incumple y en el que, por lo tanto, el valor d ya no puede ser el coste óptimo de camino más corto para el TSP, el siguiente valor posible es $d - d_n + d_{n+1}$. Para ello, se define $D' = D \cup \{d_{n+1}\} - \{d_n\}$. Por el Teorema 1', d' se puede obtener, si y solo si, D' cumple con las condiciones 1', 2' y 3'.

Es posible enumerar los cinco casos fuera de los cuales se incumple dichas condiciones y que son recogidos en las condiciones del Teorema 1. Para ello, se seguirá el diagrama de la Figura 21.

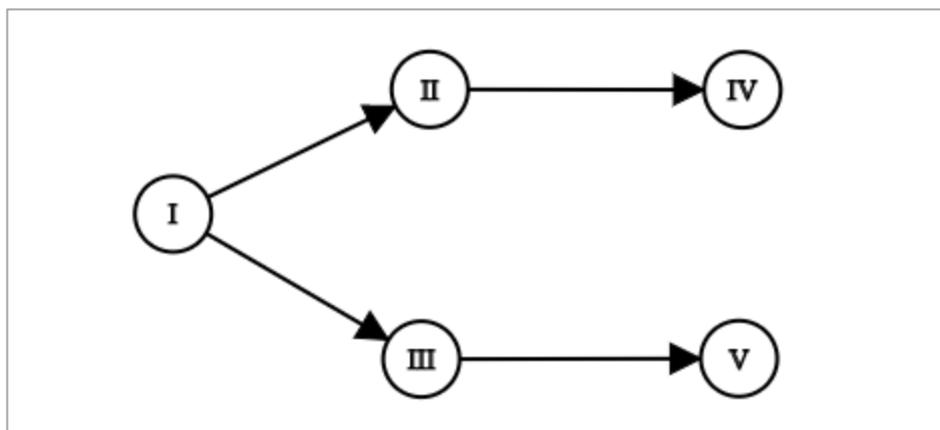


Figura 21: Diagrama de casos para el Teorema 1. Fuente: Elaboración propia.

CASO I

Siendo $D = \{a_1, \dots, a_k, b_{k+1}, \dots, b_n\}$ o $D = \{d_1, \dots, d_n\}$ y $D' = \{d_1, \dots, d_{n-1}, d_{n+1}\}$. Si $d_n = b_i$ y $d_{n+1} = b_j$, entonces la condición 1' se cumple, ya que D' contendría a a_j y a b_j , siendo siempre $1 \leq j \leq k$ y $k + 1 \leq i \leq n$. Si esto no se cumple, se pueden dar dos casos.

CASO II

Si $d_n = a_k$ y $d_{n+1} \neq b_k$, se encontraría en D' a_i y b_i , siendo $1 \leq i \leq k - 1$, por lo que la condición 1' se volvería a cumplir. Si por el contrario $d_{n+1} = b_k$ se pasaría al Caso IV.

CASO III

Si $d_n = b_{k+1}$ y $d_{n+1} \neq a_{k+1}$, se encontraría en D' a_i y b_i , siendo $k + 2 \leq i \leq n$, por lo que la condición 1' se volvería a cumplir. Si por el contrario $d_{n+1} = a_{k+1}$ se pasaría al Caso V.

CASO IV

Si el Caso II no se cumple, quiere decir que, $d_n = a_k$ y $d_{n+1} = b_k$, donde se obtiene que $D' = \{a_1, \dots, a_{k-1}, b_k, \dots, b_n\}$ para el que solo existe un caso donde se cumpliría una de las tres condiciones del Teorema 1'. Para $k = 1$, la condición 3' se cumpliría, ya que el único valor del vector a en D , a_1 se sustituiría por b_1 , lo que daría lugar a que D' se compusiera única y exclusivamente del vector b .

CASO V

Si el Caso III no se cumple, significa que $d_n = b_{k+1}$ y $d_{n+1} = a_{k+1}$, y D' resultaría $D' = \{a_1, \dots, a_{k+1}, b_{k+2}, \dots, b_n\}$, es decir, se incumpliría todas las condiciones del del Teorema 1'. Solo para el caso en el que $k = n - 1$ se cumple la condición 2', ya que en esa situación el único valor de b que pudiera estar en D sería b_n (de lo contrario se cumpliría el Teorema 1) y, por lo tanto, al sustituir b_n por a_n , D' incluiría solo y exclusivamente a los valores de a .

Finalmente, para comprobar que el resto del Teorema 1 es cierto, se supone que d' no se puede cumplir. Por lo que habría dos opciones posibles para la redefinición de D : $D' = D \cup \{d_{n+2}\} - \{d_n\}$ o $D' = D \cup \{d_{n+1}\} - \{d_{n-1}\}$. Ambas siempre cumplirán las restricciones del Teorema 1, por lo que el óptimo sería uno de estos dos valores: o $\mathbf{d} - \mathbf{d}_n + \mathbf{d}_{n+2}$ o $\mathbf{d} - \mathbf{d}_{n-1} + \mathbf{d}_{n+1}$.

Se puede añadir un tercer teorema para el caso en el que los valores contenidos en a y en b no fueran distintos.

Teorema 2 Sea C una matriz pequeña con valores no distintos para el TSP. La longitud óptima del camino mínimo para la matriz C es igual a d , si y solo si, se cumple una de las siguientes condiciones:

(Condición 7) Las condiciones 1, 2 y 3 se cumplen.

(Condición 8) $d_n = d_{n+1}$ y ninguna de las condiciones 4, 5 y 6 se cumplen.

(Condición 9) $d_n = d_{n+1} = d_{n+2}$

(Condición 10) $d_n = d_{n+1} = d_{n-1}$

Prueba Si las condiciones 1, 2 y 3 se cumplen, como dice el Teorema 1 la longitud óptima de la ruta será d . En el caso que no se cumplan ni las condiciones 1, 2 y 3, ni las 4, 5 y 6 por el Teorema 1 el valor óptimo de la longitud de la ruta será $d - d_n + d_{n+1}$, por lo tanto, si $d_n = d_{n+1}$, el valor óptimo volvería a ser d . Finalmente, en caso de incumplir las condiciones 1, 2 y 3, pero cumplir una de las condiciones 4, 5 y 6, el valor óptimo de la ruta sería $\min\{d - d_n + d_{n+2}, d - d_{n-1} + d_{n+1}\}$, donde si se cumple la condición 9 $d - d_n + d_{n+2} = d$ y si se cumple la condición 10 $d - d_{n-1} + d_{n+1} = d$, es decir, que si se cumple una de estas dos últimas condiciones, $\min\{d - d_n + d_{n+2}, d - d_{n-1} + d_{n+1}\} = d$.

Por último, puede darse el caso, en el que se tenga que decidir entre un valor de a , a_i , y otro de b , b_j , para asignarlos a d_n , tal que $a_i = b_j$. En esta situación se pueden dar tres casos que pueden dar lugar al incumplimiento del Teorema 1. Si se escoge $a_i = d_n$ de manera que $i = k$ y $1 \leq j \leq k - 1$, se incumpliría el Teorema 1, pero se cumpliría el Teorema 1, y al ser $b_j = d_{n+1}$, resultaría que $d_n = d_{n+1}$ y por el Teorema 2, el valor óptimo seguiría siendo d . Ocurriría en los mismo para el caso en el que se asigne $b_j = d_n$ y $a_i = d_{n+1}$. El tercer caso se daría cuando $i = j$, en el que el valor óptimo igual a d solo se daría si se cumplieran las condiciones 9 o 10.

2.4.4 Orden de complejidad de un algoritmo

Para medir la complejidad de un algoritmo es importante definir un conjunto de procedimientos y características que permitan comparar dos algoritmos independientes. Los dos más importantes son el espacio y el tiempo. Se entiende por espacio la memoria que ocupa el algoritmo y por tiempo la tardanza que implica su ejecución.

En cuanto al tiempo, se pueden hacer dos tipos de mediciones de complejidad: una teórica y otra real. La teórica consiste en aquella que, partiendo de unos datos de entrada, defina los límites temporales que implica la ejecución del algoritmo, es decir, defina una cota inferior y una cota superior. La real tiene en cuenta no solo los datos de entra, sino también el ordenador que ejecuta el algoritmo.

Es claro, que aquella medición que permite comparar dos algoritmos independientes entre sí es la teórica, ya que no depende del ordenador en que se ejecute.

Por otra parte, es importante definir la unidad de tiempo que se tomará para la medición. Como se evaluará solamente la medición teórica no se podrá emplear las unidades de medida de tiempo del Sistema Internacional, ya que para ello sería necesario ejecutar el algoritmo en un ordenador real. Se usará entonces, la única variable común a dos algoritmos diferentes, los datos de entrada n . Definiéndose $T(n)$ como el tiempo teórico que un algoritmo tarda en procesar n datos.

Para eludir comparar las implementaciones de un mismo algoritmo mediante la medición real, se introduce el Principio de invarianza. Que establece que siendo $T_1(n)$ el tiempo de ejecución de la implementación I_1 y $T_2(n)$ el tiempo de ejecución de la implementación I_2 , para $n \geq n_0$, siendo n_0 un número natural y c una constante, tal que $c > 0$, se evidencia que $cT_1(n) \geq T_2(n)$. Esto implica que para un mismo algoritmo lo único que diferencia una implicación de otra es una constante c .

Otro indicador de complejidad importante es la cota superior de tiempo, $O(f)$, de la función f , ya empleado en otros apartados. Tiene la propiedad de asegurar que el algoritmo nunca excederá el tiempo indicado por la cota. A continuación, se estudiarán los principales órdenes de complejidad de la cota superior. Los más comunes son los mostrados en la Tabla 2, donde se encuentran ordenados de mayor a menor complejidad, siendo el menor de todos el orden de complejidad constante y el mayor el factorial. Una comparativa de éstos se puede observar en el Gráfico 7, en el que claramente se comprueba que el más complejo es el factorial.

Este último orden es el que tiene el algoritmo de la fuerza bruta explicado en el apartado 2.4.1, ya que el bucle principal se repite $(n - 1)!$ veces, uno por cada permutación cíclica de $\{1, \dots, n\}$. Por lo que tiene un orden aproximado de $O(n!)$.

| Orden | Nombre |
|---------------|---------------|
| $O(1)$ | constante |
| $O(\log n)$ | logarítmica |
| $O(n)$ | lineal |
| $O(n \log n)$ | casi lineal |
| $O(n^2)$ | cuadrática |
| $O(n^3)$ | cúbica |
| $O(a^n)$ | exponencial |

Tabla 2: Principales órdenes de complejidad. Fuente: www.cs.us.ess/~jalonso

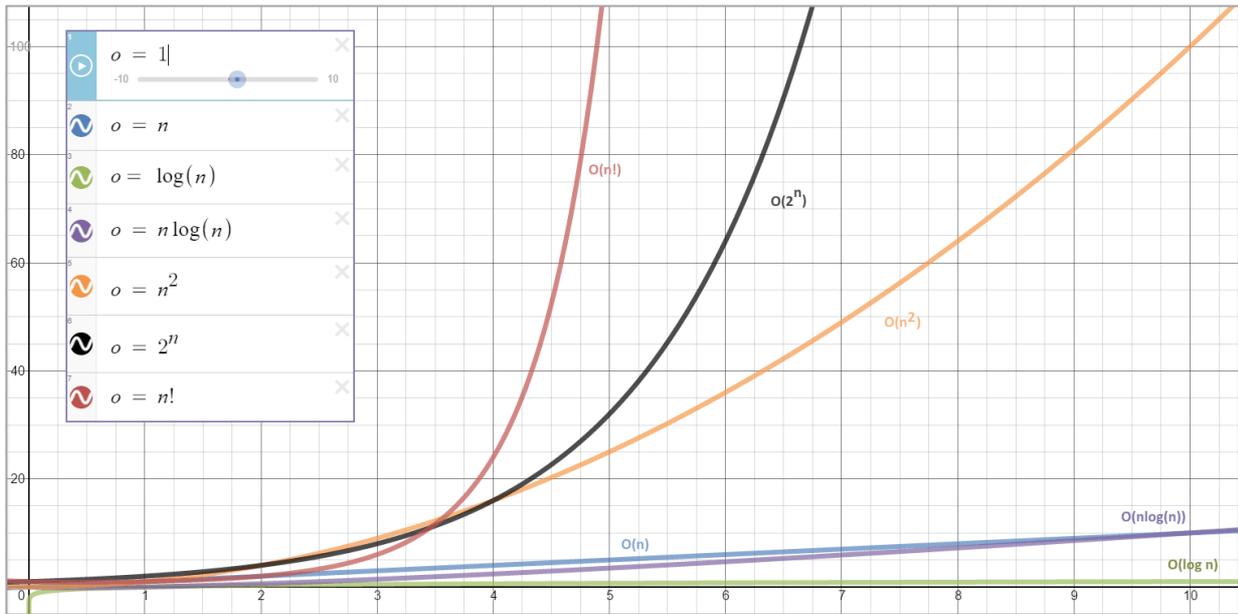


Gráfico 7: Comparativa de los órdenes de magnitud para diferentes funciones. Fuente: <https://www.campusmvp.es/>

En cuanto al algoritmo del camino mínimo, el bucle principal se itera $(n - 1)^2$ veces, y dentro de este se encuentra el de la búsqueda de mínimo que se itera un máximo de $(n - 1)$ veces, por lo que el orden de complejidad del algoritmo se puede establecer como $O(n)$.

Finalmente cabe destacar, que no para todos los algoritmos es fácil calcular su grado de complejidad. De hecho, para muchos, este depende considerablemente de los datos de entrada que reciba. De esta forma, se pueden establecer tres parámetros que caractericen la complejidad de este tipo de algoritmos: el caso mejor, el caso peor y el caso promedio.

El caso mejor es aquella casuística del algoritmo en la que el número de operaciones es el menor de los posibles, el caso peor es aquel para el que el número de operaciones es máximo y el caso promedio corresponde a la media todos los posibles casos. Para este último se debe desarrollar una distribución de probabilidad para todas las casuísticas del problema y en algunos casos puede suponer una gran dificultad el propio modelado de la distribución.

3 ALGORITMOS ESCOGIDOS

“Los grandes algoritmos son la poesía de la computación.”

- Francis Sullivan -

Aunque el TSP sea un problema difícil de resolver, con una complejidad NP-Hard y que a día de hoy siga sin haberse encontrado una manera de resolverlo en un tiempo polinomial, existen algoritmos que reducen la complejidad de este. Entre las características fundamentales del algoritmo se encuentran:

- La eficacia para encontrar una solución óptima o cercana a la óptima.
- La eficiencia para obtenerla, es decir, la complejidad.
- La sencillez del algoritmo. Aunque este punto pueda parecer trivial, cuando se mejoran, estudian y desarrollan algoritmo, la sencillez es fundamental a la hora de analizarlo. Permite, adaptarlo, descubrir errores y optimizarlo

Estos tres objetivos no suelen darse al mismo tiempo, por lo que siempre se deberá buscar un equilibrio en función de la naturaleza del problema que el algoritmo intente resolver. En este apartado se estudiará primero el conocido algoritmo de Branch and Bound, más adelante, un algoritmo de fuerza bruta conocido por Programación Dinámica y Finalmente, se tratará el algoritmo de Christofides, actualmente la heurística con tiempo polinomial que más se acerca al óptimo para el TSP.

3.1 Algoritmo de Branch and Bound

El origen del algoritmo de Branch and Bound o Ramificación y Poda para el TSP tiene su origen en Dantizing et al. [34]. El algoritmo rompe el conjunto de todas las posibles soluciones y las divide incrementalmente en pequeños subconjuntos. Para cada subconjunto se calcula un límite mínimo (*bound*) de las distancias o coste del mejor camino de las soluciones que abarque. Estos límites los va usando el algoritmo como guía para dividir las soluciones, a medida que se van dividiendo se encontrará un camino cuyo *bound* es menor que o igual que todos los *bound* del resto de soluciones. Este último camino será entonces, óptimo.

Los subconjuntos de soluciones o caminos son representados como nodos de un árbol y el proceso de división de soluciones como la ramificación de un árbol. De ahí, el nombre de ramificación y poda.

En general para cualquier algoritmo de Branch and Bound se sigue un esquema a través del cual se llega a una solución. Para un problema cualquiera P se define $v(P)$ como el valor de la solución óptima de P . Las características principales de un algoritmo de Ramificación y poda para un problema de optimización discreta P de la forma $\min\{f(x)|x \in S\}$ son:

1. Una relajación de P . Por ejemplo, un problema R de la forma $\min\{g(x)|x \in T\}$, tal que $S \subseteq T$ y para toda $x, y \in S$, $f(x) < f(y)$ implica que $g(x) < g(y)$.
2. Una norma o regla que seguir para el proceso de ramificación. Por ejemplo, una regla que consista en romper o dividir un conjunto de soluciones factibles S_i del subproblema P_i en subconjuntos más pequeños S_{i1}, \dots, S_{iq} , tal que $\bigcup_{j=1}^q S_{ij} = S_i$.
3. El cálculo de una cota mínima o *bound*. Por ejemplo, un procedimiento para encontrar $v(R_i)$ para cada relajación R_i de cada subproblema P_i .
4. Una procedimiento o regla para la selección de un subproblema. Por ejemplo, una regla para escoger el siguiente subproblema que deberá ser procesado y calculado.

3.1.1 Relajación y cálculo de la cota mínima o *bound*

Un concepto útil para la construcción de cotas mínimas es el de reducción de matrices, que se puede utilizar como una relajación del problema, R . Si una constante, h , es sustraída o restada a cada elemento de una fila de la matriz de costes c_{ij} , el coste de cualquier camino según la nueva matriz c'_{ij} será al menos menor que h con respecto a la matriz antigua. Esto es porque cada solución o camino debe contener al menos uno, y solo un elemento de dicha fila. De esta forma, aunque se hayan cambiado los costes relativos, una solución óptima en la nueva matriz también lo es en la antigua.

Este proceso de restar a cada fila una constante h , equivalente al elemento más pequeño de la fila,

se le conoce como reducción de la fila. Una matriz compuesta por elementos no negativos y con al menos un 0 en cada matriz o cada columna es llamada matriz reducida, y puede ser obtenida, por ejemplo, reduciendo filas y columnas. Si $c(\pi)$ es el coste de un camino en una matriz de costes antes de ser reducida, $c_1(\pi)$ el nuevo coste para la matriz reducida y h la suma de constantes que reducen la matriz, entonces:

$$c_1(\pi) = c(\pi) + h$$

Ya que la matriz reducida no contiene elementos no negativos, h se puede utilizar como una cota o límite mínimo para el coste de la solución π de la matriz de coste original.

Ejemplo Para un TSP con cinco ciudades se muestra el proceso de reducción de la matriz de costes y el resultado correspondiente:

Matriz original c_{ij} con reducción de filas:

$$c_{ij} = \begin{pmatrix} \infty & 20 & 30 & 10 & 11 \\ 15 & \infty & 16 & 4 & 2 \\ 3 & 5 & \infty & 2 & 4 \\ 19 & 6 & 18 & \infty & 3 \\ 16 & 4 & 7 & 6 & \infty \end{pmatrix} \begin{matrix} \rightarrow 10 \\ \rightarrow 2 \\ \rightarrow 2 \\ \rightarrow 3 \\ \rightarrow 4 \end{matrix} \quad \rightarrow \quad c'_{ij} = \begin{pmatrix} \infty & 10 & 20 & 0 & 1 \\ 13 & \infty & 14 & 2 & 0 \\ 1 & 3 & \infty & 0 & 2 \\ 16 & 3 & 15 & \infty & 0 \\ 14 & 0 & 3 & 2 & \infty \end{pmatrix}$$

Se obtiene así que la cota obtenida por la reducción de filas, h_f , es igual a 21.

Matriz reducida por filas con reducción de columnas:

$$c'_{ij} = \begin{pmatrix} \infty & 10 & 20 & 0 & 1 \\ 13 & \infty & 14 & 2 & 0 \\ 1 & 3 & \infty & 0 & 2 \\ 16 & 3 & 15 & \infty & 0 \\ 14 & 0 & 3 & 2 & \infty \end{pmatrix} \quad \rightarrow \quad c'_{ij} = \begin{pmatrix} \infty & 10 & 17 & 0 & 1 \\ 12 & \infty & 11 & 2 & 0 \\ 0 & 3 & \infty & 0 & 2 \\ 15 & 3 & 12 & \infty & 0 \\ 13 & 0 & 0 & 2 & \infty \end{pmatrix}$$

$$\begin{matrix} \downarrow & \downarrow & \downarrow & \downarrow & \downarrow \\ 1 & 0 & 3 & 0 & 0 \end{matrix}$$

Se obtiene así que la cota obtenida por la reducción de columnas, h_c , es igual a 4. Por lo tanto, la cota h de la matriz reducida se obtiene:

$$h = h_c + h_f$$

Finalmente, se puede decir que al ser $h = 25$ no existirá ninguna solución menor que 25.

Este cálculo de la matriz reducida se corresponde con la regla para el cálculo de un límite mínimo del apartado anterior (apartado 3.1), donde $v(R_i)$ corresponde al valor de h y S_i abarca todas las posibles soluciones del problema.

3.1.2 Proceso de ramificación

La división de las soluciones en subconjuntos de éstas se representa como la ramificación de un árbol, como se muestra en la Figura 22. El primer nodo contiene todas las posibles soluciones y corresponde al punto de partida del viajante, sería el nivel 0; habrá tantos niveles como ciudades tenga el problema. Desde ahí se llega al nivel 1, que corresponde con la primera ciudad que visita el viajante. Sucesivamente se llegará hasta el nivel n , que corresponderá con la última ciudad por visitar.

Un proceso de ramificación en profundización consiste en comprobar las cotas mínimas de cada subconjunto de soluciones profundizando en los niveles, es decir, primero se llega hasta el último nivel para un subconjunto de soluciones (hasta quedarse solo con una) y luego se explora el resto de los subconjuntos, desde el nivel n hasta el primero.

En la Figura 22 primero se ha visitado el subconjunto de ciudades S_B en las que A y B son las primeras ciudades en ser visitadas, luego el S_{BC} (visitando primero A, B y luego C), a continuación, el S_{BCD} y finalmente el S_{BCDE} . Es decir, primero se ha llegado a la solución para la permutación S_{BCDE} y luego se ha ido explorando el resto de las soluciones.

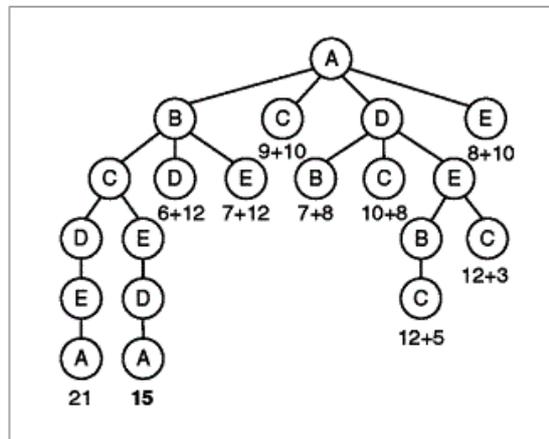


Figura 22: Ramificación en profundidad para un problema de 5 nodos. Fuente [42]

Se puede ver que en este apartado se ha definido la segunda regla del apartado (apartado 3.1), la correspondiente a la ramificación de los subconjuntos de soluciones. Por ejemplo, $S_B = S_{BC} + S_{BD} + S_{BE}$. Es decir, la división de un subconjunto de soluciones en el problema de TSP, consiste en obligar y en restringir la siguiente ciudad en ser visitada, dejando el orden del resto de ciudades por ser visitadas libre.

3.1.3 Cálculo de la cota mínima para ramas de niveles superiores

En el apartado 3.1.1 se ha visto como reducir la matriz completa c_{ij} sin haber visitado todavía ninguna ciudad, pero cuando se visita una ciudad, se debe calcular una nueva cota mínima. Es decir, visitar una ciudad implica un nuevo subproblema y por lo tanto una nueva cota. Se continúa el ejemplo anterior para mostrar este cálculo.

Ejemplo Partiendo de la matriz c'_{ij} se decide visitar primero el nodo 1 y luego el nodo 4. Para ello la matriz deberá ser modificada y a la nueva matriz le corresponderá un nuevo valor de reducción.

Cuando se visita un nodo j desde un nodo i se establecen todos los elementos de la fila i y la columna j como infinito. De esta manera la nueva matriz quedaría:

$$\begin{array}{c} \downarrow \\ \rightarrow \end{array} c^{(1)'}_{ij} = \begin{pmatrix} \infty & 10 & 17 & 0 & 1 \\ 12 & \infty & 11 & 2 & 0 \\ 0 & 3 & \infty & 0 & 2 \\ 15 & 3 & 12 & \infty & 0 \\ 13 & 0 & 0 & 2 & \infty \end{pmatrix} \rightarrow c^{(2)'}_{ij} = \begin{pmatrix} \infty & \infty & \infty & \infty & \infty \\ 12 & \infty & 11 & \infty & 0 \\ 0 & 3 & \infty & \infty & 2 \\ 15 & 3 & 12 & \infty & 0 \\ 13 & 0 & 0 & \infty & \infty \end{pmatrix}$$

De esta manera la nueva matriz $c^{(2)'}_{ij}$ puede volver a ser reducida, ya que si se sustituye un 0 por infinito el nuevo mínimo de la columna o fila varía. En este caso, no existe ningún nuevo mínimo, por lo que $h^{(2)} = 0$. Por lo tanto, el cálculo de la nueva cota sería igual a la constante de la matriz reducida anterior, más el de la nueva, más la nueva distancia recorrida. Es decir:

$$U_{14} = h^{(1)} + c^{(1)'}_{14} + h^{(2)} = 25 + 0 + 0 = 25$$

El procedimiento sería el mismo si del nodo 4 se quisiera ir al 2:

$$\begin{array}{c} \downarrow \\ \rightarrow \end{array} c^{(2)'}_{ij} = \begin{pmatrix} \infty & \infty & \infty & \infty & \infty \\ 12 & \infty & 11 & \infty & 0 \\ 0 & 3 & \infty & \infty & 2 \\ 15 & 3 & 12 & \infty & 0 \\ 13 & 0 & 0 & \infty & \infty \end{pmatrix} \rightarrow c^{(3)'}_{ij} = \begin{pmatrix} \infty & \infty & \infty & \infty & \infty \\ 12 & \infty & 11 & \infty & 0 \\ 0 & \infty & \infty & \infty & 2 \\ \infty & \infty & \infty & \infty & \infty \\ 13 & \infty & 0 & \infty & \infty \end{pmatrix}$$

Donde, de nuevo no se puede reducir más la matriz y $h^{(3)} = 0$. En este caso, se calcula la cota inferior del siguiente nivel, partiendo de la cota del nivel anterior:

$$U_{142} = U_{14} + h^{(3)} + c^{(2)'}_{42} = 25 + 0 + 3 = 28$$

Esta cota se interpreta como: si se quiere visitar primera la ciudad 1, luego la ciudad 4, luego la ciudad 2 y finalmente la 3 y la 5, estas dos últimas sin orden definido, al menos costará 28.

De esta forma se calcula cualquier cota para cualquier subproblema TSP_{iq} .

3.1.4 Regla de selección de subproblemas

Como se ya se ha comentado, el algoritmo divide el problema en subproblemas. En el TSP_{iq} un subproblema iq consiste en establecer el nodo i como el inicial y en otro subconjunto S_{iq} de nodos visitados, de manera que el orden de visita del subconjunto del resto de nodos sin visitar, $A \setminus S_{iq}$, quede por determinar. Así, para cada subconjunto de nodos S_{iq} existe un subproblema TSP_{iq} .

El primer paso, como se ha comentado en el apartado anterior, es calcular el valor del primer subconjunto en el nivel n (en la Figura 22, el primer subconjunto de nivel n es el $\{A, B, C, D, E\}$) y dicho valor se establece como la cota mínima hasta el momento. Para el siguiente paso se deberá escoger otro subconjunto de soluciones o subproblema que deberá ser evaluado y comparado con la cota mínima.

Para ello primero se bajará al nivel $(n - 1)$ y se evaluará por orden alfabético o de menor a mayor los nodos que queden por visitar en dicho nivel. Si en ese nivel ya se han visitado todos los nodos, se bajará al nivel $(n - 2)$ y así sucesivamente hasta haber evaluado todos los niveles. Nótese que para el nivel n y $(n - 1)$ solo existe una permutación de ciudades. Sin embargo, para el nivel $(n - 2)$, existirán dos, una por cada ciudad sin visitar. Por ejemplo, en la Figura 22, en el nivel $(n - 2)$ para el subconjunto $\{A, B, C\}$ existen dos ciudades por ser visitadas: D y E , y por lo tanto dos permutaciones: $\{A, B, C, D, E\}$ y $\{A, B, C, E, D\}$.

3.1.5 Algoritmo

Finalmente, teniendo en cuenta la relajación del problema mediante matrices reducidas, la obtención de la cota mínima mediante la constante sobrante de reducir la matriz y la regla de ramificación en profundización, se puede establecer un algoritmo de Branch and Bound. Este se muestra en la Figura 23.

1. **Inicializar** Se añade el TSP completo a la lista de subproblemas y se define la cota mínima $U = \infty$
2. **Selección del subproblema**
 - a. Si la lista de subproblemas está vacía \rightarrow **Fin**, el camino asociado con U es óptimo (si $U = \infty$, no existe solución).
 - b. Si la lista está llena, se escoge un subproblema TSP_i y se elimina este de la lista.
3. **Cota mínima** Se resuelve la relajación (matriz reducida) R_i del problema TSP_i , es decir, se calcula su cota mínima $v(R_i)$, siendo L_i el valor obtenido.
 - a. Si $L_i \geq U$, se vuelve al paso 2.
 - b. Si $L_i < U$ y la solución es equivalente a un nuevo camino o ruta se establece $U \leftarrow L_i$. Si la solución no equivale a un camino completo se sube de nivel y se pasa al paso 4.
4. **Reducción** Se eliminan todos los arcos que deriven de TSP_i cuya inclusión sobrepase la cota mínima.
5. **Ramificación** Se ramifica el problema TSP_i y se generan el resto de los subproblemas que derivan de este $TSP_{i1}, \dots, TSP_{iq}$, estos se incluyen en la lista. Se vuelve al paso 2.

Figura 23: Pseudo algoritmo de Branch and Bound.

3.2 Algoritmo de Programación Dinámica

Este algoritmo es de tipo fuerza bruta pero más eficiente que el algoritmo explicado en el apartado 2.4.1. Desarrollado por Held y Karp [43] este encuentra para cada ciudad i el camino más corto desde la ciudad 1 hasta la ciudad i visitando todos los nodos del subconjunto $N = \{2, \dots, n\}$. Una vez se hayan encontrado estos caminos, por comparación se encuentra fácilmente el camino óptimo, tal como se muestra en la Figura 24.

```
* Entrada: Número entero  $n \geq 3$ , matriz  $n \times n$   $C = (c_{ij})$ , donde cada  $c_{ij}$  es un número real mayor o igual que 0.
* Salida: El camino más corto que uno todos los nodos  $N$ , que empieza en 1 y acaba en  $n$ .
comienzo
  for  $i = 2, 3, \dots, n$  do
     $\text{coste}[\{i\}, i] = c_{1i}$ ;
     $\text{mejor\_camino}[\{i\}, i] = (1, i)$ ;
  end
  for  $i = 2, 3, \dots, n - 1$  do
    for each  $S \subseteq \{2, 3, \dots, n\}$  con  $|S| = j$ 
      for each  $i \in S$  do
         $\text{coste}[S, i] = \min_{k \in S - \{i\}} \{\text{coste}[S - \{i\}, k] + c_{ki}\}$ ;
        se estable  $k$  como la ciudad para la que el coste es mínimo;
         $\text{mejor\_camino}[S, i] = (\text{mejor\_camino}[S - \{i\}, k], i)$ 
      end
    end
     $\text{coste\_minimo} = \min_{k \neq 1} \{\text{coste}[S - \{1\}, k] + c_{ki}\}$ ;
    se estable  $k$  como la ciudad para la que el coste es mínimo;
     $\text{mejor\_ruta} = \text{mejor\_camino}[\{2, 3, \dots, n\}, k]$ 
  en
  output  $\text{mejor\_ruta}$ 
finaliza algoritmo
```

Figura 24: Pseudo código del algoritmo de Held y Karp para el TSP. Fuente [28].

Para calcular el camino mínimo de 1 hasta i , el algoritmo resuelve el problema construyendo paso a paso la solución final. Para $S \subseteq \{2, 3, \dots, n\}$ y $i \in S$, se define (S, i) como un camino que finaliza en el nodo i y que visita una única vez todos los nodos contenidos en S y ningún otro más. También,

se define $\text{coste}[S, i]$ como la distancia recorrida por la ruta mínima de (S, i) . Por lo tanto, $\text{coste}[S, i]$ satisface la siguiente ecuación cuando $|S| \geq 2$:

$$\text{coste}[S, i] = \min_{k \in S - \{i\}} \{\text{coste}[S - \{i\}, k] + c_{ki}\} \quad (57)$$

En el lado derecho de la ecuación (57) se calcula el mínimo de todos los caminos de $(S - \{i\}, k)$ sumado a la distancia entre k e i , donde $S - \{i\}$ es el subconjunto S excluyendo la ciudad i y k es una ciudad perteneciente a $S - \{i\}$.

En el primer bucle del algoritmo se obtienen los costes para la permutación $S = (1, i)$, siendo $|S| = 2$. En el siguiente bucle, mediante la ecuación (57) se construye paso a paso la permutación óptima con el camino mínimo de (S, i) hasta que i sea igual a n , momento en el cual se sale del bucle. Para este momento ya se habrán calculado las rutas mínimas de (S, i) siendo $|S| = n$.

Finalmente, una vez se tienen las rutas mínimas se comparan todas ellas sumando al $\text{coste}[S, i]$ la distancia entre i y 1 , c_{ki} , obteniéndose de esta forma el camino mínimo $\{(S, k), 1\}$, siendo k la última ciudad en ser visitada. Este tipo de técnica algorítmica en la que se va solucionando cada vez problemas más grandes para llegar a una solución mayor es conocida como programación dinámica.

3.2.1 Complejidad

Para el algoritmo de programación dinámica el rendimiento es menor, la mayor complejidad del algoritmo reside en la obtención del subconjunto $S \subseteq \{2, 3, \dots, n\}$, que puede ser cualquiera dentro del conjunto $\{2, 3, \dots, n\}$ siempre y cuando $|S| = j$. Esta operación se encuentra dentro de un bucle que se repite $n - 1$ veces. Por otra parte, la búsqueda del subconjunto S contiene otros bucles que se encargan de buscar el mínimo para cada elemento de S y que se repite j veces, hasta un máximo de $n - 1$ veces por lo que el número de operaciones quedaría de la siguiente manera:

$$\sum_{j=2}^{n-1} \binom{n-1}{j} j^2 \leq 2^{n-1} (n-1)^2 \quad (58)$$

El lado derecho de la inecuación (58) es el límite máximo que podría tener el algoritmo para $j = n - 1$, El sumatorio del coeficiente binomial se puede calcular a partir de la ecuación de Newton:

$$(x + y)^n = \sum_{k=0}^n \binom{n}{k} x^{n-k} y^k \quad (59)$$

Donde para $x = 1$ y $y = 1$, se obtiene que:

$$2^n = \sum_{k=0}^n \binom{n}{k}$$

Sustituyendo en la ecuación (58), se tiene que el límite máximo se puede aproximar a 2^{n-1} , de manera que el orden de complejidad de $2^{n-1}(n-1)^2$ se puede establecer como $O(2^n n^2)$. El resto de los bucles del algoritmo, tanto el primero como el último son de orden $O(n)$ por lo que queda absorbido por el bucle principal.

3.3 Algoritmo Heurístico de Christofides

Esta heurística la definió por Christofides (1976) [44]. Está basado en la relación entre los árboles de mínimo coste y el problema del viajante.

Un árbol de mínimo coste es un árbol tal que la suma de los costes de sus aristas es mínima. Si los costes vienen reflejados en una matriz de orden n , la complejidad computacional para alcanzar el óptimo es aproximadamente $O(n^2)$.

La resolución de un problema de este tipo proporciona una cota mínima del coste de la solución óptima al TSP: esto es debido a que si en la solución óptima, que será un camino hamiltoniano, se quita un arco, se obtiene un árbol. En otras palabras, si T es el árbol de cobertura mínima y τ es la permutación cuyo valor es óptimo en el TSP, resulta que $c(T) \leq c(\tau)$.

Supóngase que se quiere encontrar una ruta para visitar todas las ciudades y que solo se puede utilizar los arcos del árbol de menor coste. En este caso, si se duplica dicho árbol y se da una orientación diferente a los 2 arcos que conectan las mismas ciudades, se obtendrá un circuito que recorre todas las ciudades con un coste igual al doble del coste del árbol de cobertura mínima, es decir $2c(T)$. El problema de esta ruta, que es lo que la convierte en la ruta óptima del TSP, es que algunas ciudades se visitan más de una vez. Pues si se aplica lo que se denominan atajos (seguir la ruta propuesta hasta esto nos hace visitar una ciudad ya visitada y en ese punto unirnos a la ciudad en la que nos encontramos con la siguiente ciudad no visitada según el orden establecido por el circuito hasta todos los vértices están unidos) se obtendrá un circuito que recorre todas las ciudades una vez y que costará menos de 2 veces el coste del óptimo del árbol de cobertura mínima y, por lo tanto, menos de 2 veces el coste de la ruta óptima.

El algoritmo que se acaba de describir se denomina “Algoritmo de árbol de menor coste” (Figura 25) y tiene múltiples variantes. Ahora bien, si lo que se busca es mejorar este límite de 2 veces el coste óptimo, se debe utilizar el algoritmo de Christofides porque, como ya se mencionó anteriormente, el mejor límite de este tipo encontrado para una heurística del TSP está asociado a este algoritmo.

| |
|--|
| <p>Dado un grafo $G = (V, E)$</p> <ol style="list-style-type: none">1. Encontrar un árbol de cobertura mínima T de G.2. Generar un segundo grafo G' duplicando las aristas de T.3. Encontrar un camino euleriano de G' y un camino hamiltoniano τ contenido en él. |
|--|

Figura 25: Pseudo algoritmo de árbol de menor coste.

La idea de Christofides fue mejorar el circuito sobre el que luego se aplican los atajos. Para ello, se recurre a los grafos eulerianos, que se definen como aquellos grafos conexos cuyos vértices tienen todos grado par. En un grafo euleriano es relativamente fácil encontrar un circuito euleriano, es decir, un circuito que recorra cada arco exactamente una vez, por lo que una vez obtenido un circuito euleriano puede obtenerse un circuito hamiltoniano (en el caso anterior, la forma de encontrar un circuito euleriano consistía en duplicar el árbol de mínimo coste), es decir, el cambio se encuentra en el paso 2 del algoritmo del árbol, Figura 25. Christofides propuso recurrir al 1-emparejamiento perfecto para obtener dicho circuito euleriano. Un 1-emparejamiento perfecto es una colección de arcos tales que cada vértice es extremo de un único arco. Dado un conjunto de vértices, puede demostrarse que el 1-emparejamiento perfecto de coste mínimo puede obtenerse en un tiempo $O(n^3)$ (Figura 26).

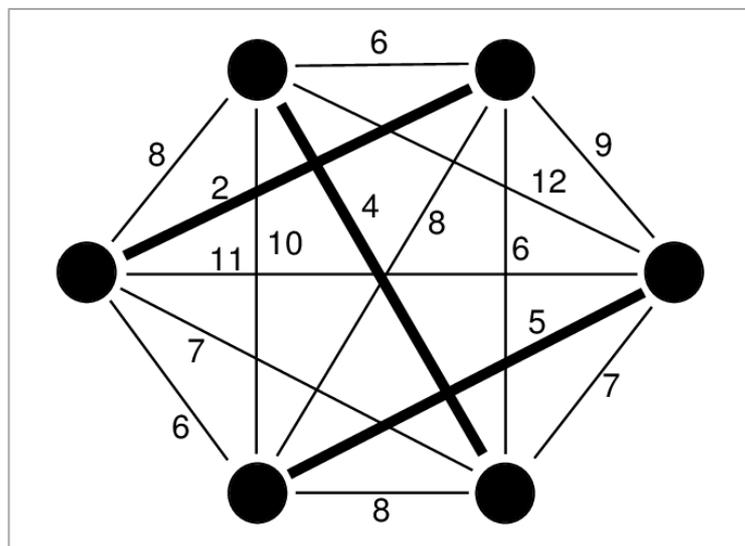


Figura 26. Un 1-emparejamiento perfecto para un grafo de 6 nodos. [45]

El algoritmo de Christofides comienza obteniendo el árbol de mínimo coste. Algunos de los vértices de este árbol tendrán grado par, por lo que no será necesario añadirle nada. Pero no ocurre lo mismo con los nodos de grado impar. Una forma simple de conseguir que todos los vértices tengan grado par es añadir el 1-emparejamiento perfecto de coste mínimo de los vértices de grado impar. Esto hará que el grado de los nodos impares se incremente en una unidad, convirtiéndolos en nodos de grado par. De esta forma, se obtiene un grafo euleriano.

Dado un grafo G :

1. Encontrar un árbol de cobertura mínima T .
2. Generar un segundo grafo G' con los vértices de grado impar en T . Entonces, G' tiene un número par de vértices.
3. G' corresponde a un grafo completo con un número par de vértices, donde se define M como el 1-emparejamiento perfecto de G' .
4. $T + M$ tienen un camino euleriano K ya que todos los nodos son de grado par [40].
5. Escoger el atajo en K para llegar a un camino hamiltoniano para el TSP.

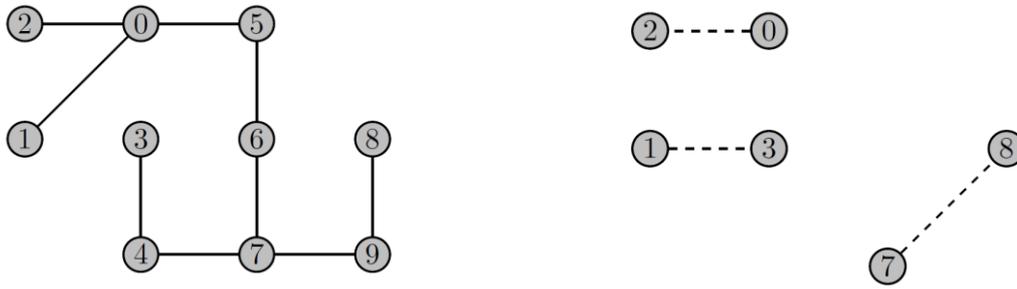
En la Figura 27 se muestra un ejemplo de algoritmo de Christofides para un grafo completo de 10 nodos. En el paso 1 se encuentra un árbol de cobertura mínimo T . La suma de los grados de todos los nodos es igual a $S(d) = 2m$, donde m es el número de aristas. Luego $S(d)$ es un número par. Si se toma $S_e(d)$ como la suma de los grados de los nodos que tienen grado par, también $S_e(d)$ es par. Por lo tanto, $S(d) - S_e(d) = 2k$, siendo $k = 1, 2, \dots$, de donde se obtiene que la suma de los nodos que tienen grado impar también es par. Se concluye entonces, que para que una suma de números impares sea par, el número de elementos sumados debe ser también par, entonces, el número de vértices de grado impar es par. En el paso 2 de la Figura 27 se comprueba que existen 6 nodos de grado impar, por lo que siempre será posible hacer un 1-emparejamiento perfecto entre ellos.

En el paso 3 se genera un nuevo grafo a partir de las aristas obtenidas en el paso 2, provenientes del 1-emparejamiento perfecto. De este modo todos los nodos tienen ahora grado par por lo que por el teorema de Euler [40], se puede asegurar que existe un camino euleriano que recorre todas las aristas, paso 4: $\{0, 2, 0, 5, 6, 7, 9, 8, 7, 4, 3, 1, 0\}$. Una vez hallado el camino euleriano, se busca a partir de este un camino hamiltoniano. En el siguiente paso se elimina el paso por los nodos que se repiten en el camino euleriano: $\{0, 7\}$, por lo que la secuencia final queda: $\{0, 2, 5, 6, 7, 9, 8, 4, 3, 1, 0\}$

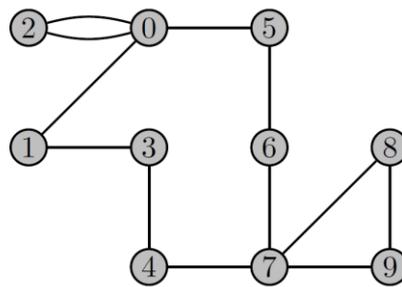
Christofides demostró que, como máximo, la longitud de la ruta obtenida será 1.5 veces la de la ruta óptima si la matriz de distancias cumple la desigualdad triangular. Es decir, si:

| | | |
|----------------------------------|-------------------------|------|
| $c(u, w) \leq c(u, v) + c(v, w)$ | $u, v, w = 1, \dots, n$ | (60) |
|----------------------------------|-------------------------|------|

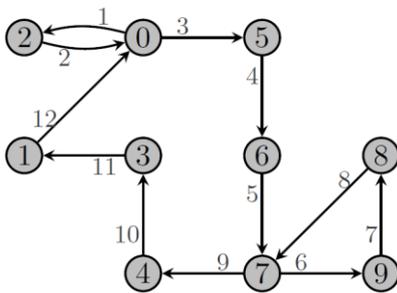
La ecuación (60) se cumplirá siempre en el espacio euclidiano. Es decir, que para ir de una ciudad a otra, el camino más corto es yendo directamente a ella sin pasar por ninguna otra más.



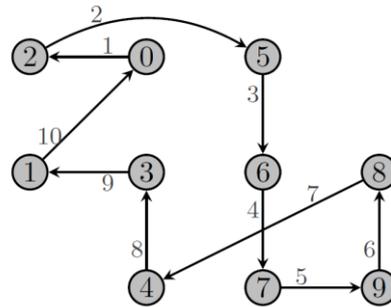
1.- Árbol de cobertura mínimo para el grafo G . 2.- El 1-emparejamiento perfecto de nodos de grado impar.



3.- Generación de un nuevo grafo $(T + M)$.



4.- Selección del camino euleriano.



5.- Selección de atajos K para crear un camino Hamiltoniano

Figura 27: Dado un grafo completo G de 10 nodos se aplica el algoritmo de Christofides para hallar una solución al TSP del grafo G . Fuente [46]

4 ADAPTACIÓN AL PROBLEMA DE AMAZON EN SEVILLA

*“La distancia más corta entre dos puntos es
una línea recta.”*

-Arquímedes -

Para poder aplicar los algoritmos presentados en el capítulo 3, es necesario desarrollar un grafo G que represente el problema que queremos resolver, se aplicará el TSP a dicho grafo y se resolverá mediante los tres algoritmos escogidos.

Como se dijo en el apartado 1.5, la zona oeste de Sevilla presenta grandes congestiones y retenciones de tráfico. Sobre todo, debido al tramo inacabado de la SE-40. En esa línea se pretende modelar el conjunto de pueblos que conforma el área oeste metropolitana de Sevilla, de manera que sirva como método de reducción de costes, tiempos o distancias, según la medida que se quiera emplear.

Para el modelado se empleará el uso del problema del viajante, entendiéndose este como aquel que para un vehículo cuyo objetivo es visitar todos los pueblos del área oeste de Sevilla, partiendo desde el centro logístico de Amazon en Dos Hermanas y regresando a este mismo punto, lo haga de la manera más eficiente posible, es decir, minimizando el coste de la ruta escogida. Nótese que el problema es un puro problema del viajante, es decir, no se consideran restricciones de tiempo ni de capacidad. Solamente la obligatoriedad de visitar un pueblo una sola vez y no más.

4.2 Arcos

Los arcos del problema son las conexiones entre los pueblos. Estas conexiones pueden ser directas o indirectas, es decir, que se puede pasar a través de un pueblo para llegar a otro sin que el algoritmo dé por visitado el pueblo intermedio. Esta técnica se emplea para los grafos incompletos y está explicada en el apartado 2.2.2. En este apartado se explicaba que en aquellos grafos en los que existan nodos no conectados con otros, se puede relajar el problema, calculando el camino mínimo entre un nodo y otro (problema resuelto por el algoritmo de Floyd-Warshall [29]). De esta forma el problema, se ha modelado como un grafo conexo, donde todos los nodos están conectados entre sí.

Claramente, las conexiones están representadas por carreteras, autopistas o autovías.

4.3 Matriz de costes

Para el presente Proyecto, se ha querido dar un enfoque distinto a la unidad de medida empleada en el problema. Normalmente, se tienen en cuenta tres unidades básicas: tiempo, distancia y dinero. En el caso de las carreteras, se puede introducir una variable más que represente el estado de la carretera o el tipo de esta. Esto quiere decir que no es lo mismo ir por una autopista que por una carretera nacional. Normalmente, las vías grandes suelen estar más descongestionadas y en mejor estado, lo que implica un ahorro de tiempo, coste e incluso distancia por la velocidad de circulación. En ese sentido, se ha querido reforzar el peso de este tipo de carreteras en el Proyecto, de manera que el empleo de éstas tenga un impacto menor en la función de costes que el empleo de las carreteras nacionales o comarcales.

Por lo tanto, la variable escogida, es una modificación de la longitud real a través de carreteras entre dos pueblos de manera que, se define:

- w_{ij} como la distancia recorrida a través de carretera comarcal o nacional entre el pueblo i y el pueblo j .
- v_{ij} como la distancia recorrida a través de autopista o autovía entre el pueblo i y el pueblo j .
- d_{ij} como la longitud real a través de carreteras o autopistas entre el pueblo i y el pueblo j .
- c_{ij} como la distancia virtual empleada en el problema del Proyecto entre el pueblo i y el pueblo j .

De esta forma la relación entre las distancias es la siguiente:

$$d_{ij} = w_{ij} + v_{ij} \quad i, j = 1, \dots, n \quad (61)$$

$$c_{ij} = 2w_{ij} + v_{ij} \quad i, j = 1, \dots, n \quad (62)$$

Es decir, como se puede ver en la ecuación (62), se ha utilizado el doble de penalización al empleo de carreteras comarcales o nacionales. La matriz de costes se adjunta en el Anexo I.

5 RESULTADO Y CONCLUSIONES

*“Mide lo que es medible, y haz medible lo que
no lo es”*

- Galileo Galilei -

Aunque, un algoritmo dé un resultado óptimo, o ni siquiera dé una solución para el problema en cuestión, siempre provee datos que sirven para entender mejor el problema que se está intentando resolver. De los algoritmos presentados en este Proyecto, dos son heurísticos y uno es de fuerza bruta, cada uno tiene sus ventajas y sus inconvenientes, pero todos aportan algo al entendimiento del problema. La complejidad de la formulación de una heurística puede ralentizar el ordenador, aunque la sencillez de un algoritmo de fuerza bruta también. Del mismo modo, la sutileza de otro algoritmo puede proveer resultados en muy poco tiempo, pero pueden darse malas (o no cercanas al óptimo) soluciones. Este es uno de los grandes equilibrios que se buscan en los desarrollos de algoritmos, entre la simpleza y la complejidad, la precisión en detrimento de la velocidad, o la rapidez a costa de la bondad de la solución.

Con los tres algoritmos presentados, este equilibrio o estas diferencias entre unos y otros se puede observar perfectamente. En este apartado se analizarán las soluciones obtenidas por el algoritmo en dos grafos diferentes: uno con 23 nodos y otro con 11. Se hará una breve comparación de los mismos, enumerando las ventajas e inconvenientes de cada uno.

Se verán principalmente tres datos de entrada: La secuencia, la distancia virtual recorrida y el tiempo real empleado, que servirá a modo de comparativa.

5.1 Resultados del algoritmo de Branch and Bound

Este algoritmo se caracteriza por su equilibrado balance entre la bondad de la solución y el tiempo empleado en alcanzarla.

5.1.1 Caso de 23 nodos

Para el caso de los 22 pueblos y el centro logístico de Amazon en Dos Hermanas el algoritmo llega a la siguiente solución:

['Amazon SVQ1' , 'Gelves' , 'San Juan de Aznalfarache' , 'Mairena del Aljarafe' , 'Palomares del Río' , 'Almensilla' , 'Bollullos de la Mitación' , 'Umbrete' , 'Benacazón' , 'Sanlúcar la Mayor' , 'Olivares' , 'Villanueva del Ariscal' , 'Espartinas' , 'Salteras' , 'Gines' , 'Bormujos' , 'Castilleja de la Cuesta' , 'Castilleja de Guzmán' , 'Valencina de la Concepción' , 'Santiponce' , 'Camas' , 'Tomares' , 'Amazon SVQ1']

Esta secuencia provee un coste de 212'7 km virtuales, con un tiempo de ejecución de 938.8700778 segundos.

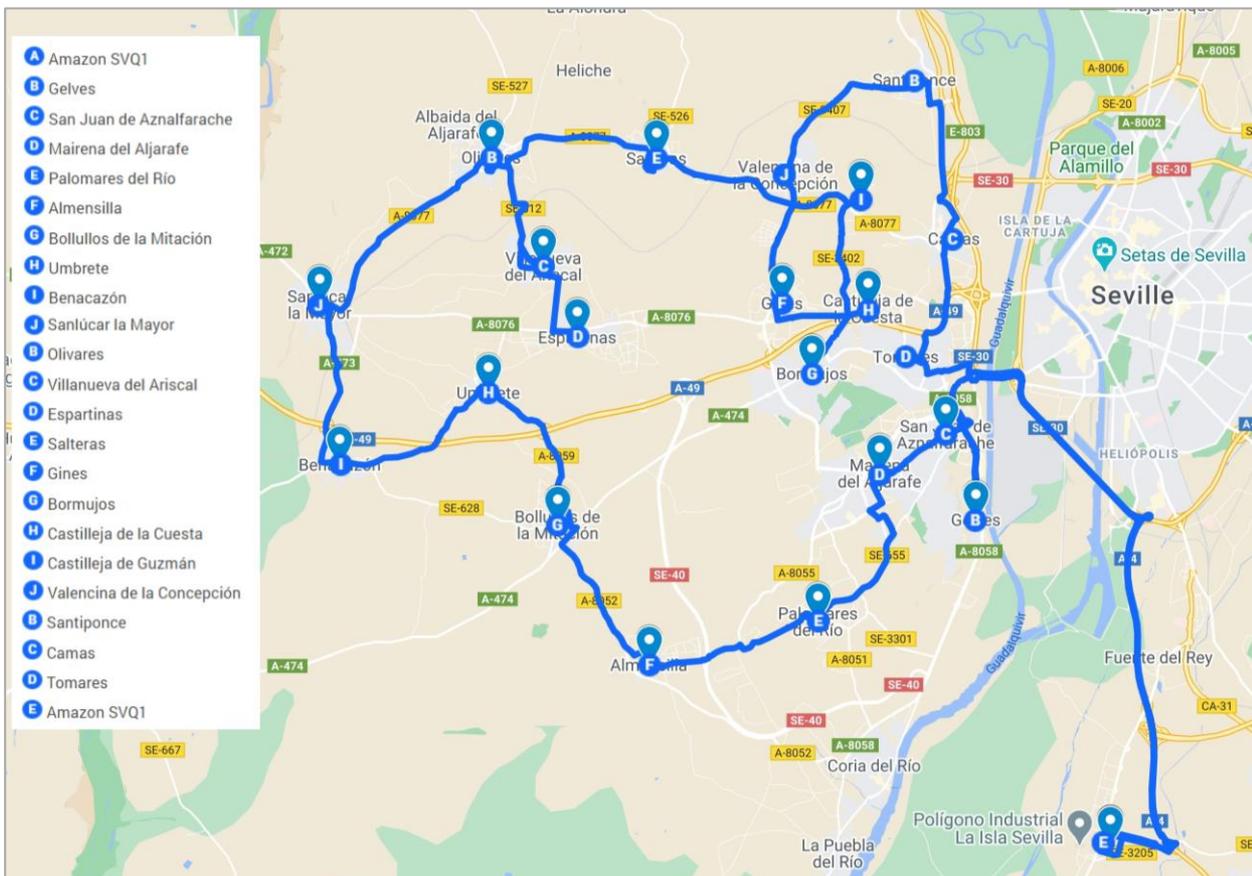


Figura 29: Resolución del problema de los 23 nodos con el algoritmo de Branch and Bound.

5.1.2 Caso de 12 nodos

Con el objetivo de comparar también los algoritmos para problemas más pequeños, se estudia su comportamiento para el caso de doce nodos.

Este es un subproblema del problema anterior para el que solo se tienen en cuenta las once primeras ciudades: Gelves, San Juan de Aznalfarache, Camas, Santiponce, Tomares, Sanlúcar la Mayor, Castilleja de la Cuesta, Palomares del Río, Castilleja de Guzmán, Mairena del Aljarafe y Bormujos. Además del centro logístico de Amazon: Amazon SVQ.

El comportamiento del algoritmo en este punto es proporcional al anterior, sobre todo, en cuanto a tiempo se refiere. Este ha sido de 102.1581888 segundos y a alcanzado el óptimo con una distancia virtual de 166.7 km.

La secuencia obtenida es

['Amazon SVQ1', 'San Juan de Aznalfarache', 'Mairena del Aljarafe', 'Palomares del Río', 'Gelves', 'Santiponce', 'Camas', 'Castilleja de Guzmán', 'Castilleja de la Cuesta', 'Bormujos', 'Sanlúcar la Mayor', 'Tomares', 'Amazon SVQ1']

5.1.3 Caso de 16 nodos

Para cuando el número de pueblos es 15, el problema muestra una singularidad, existen casos en los que la obtención del óptimo no llega hasta que se han explorado todos o casi todos los subproblemas TSP_{iq} . Esto provoca que la complejidad del algoritmo en su peor caso se mayor, aumentando así el tiempo de ejecución.

Este ha sido el caso en el que el número de nodos es 16, para el que el algoritmo ha hallado una solución con un recorrido virtual de 196'45km. En un tiempo de 2785'682 segundos, lo que muestra la gran singularidad del presente caso. Por último, la ruta encontrada es la siguiente, que es la óptima:

['Amazon SVQ1', 'Gelves', 'San Juan de Aznalfarache', 'Mairena del Aljarafe', 'Palomares del Río', 'Almensilla', 'Sanlúcar la Mayor', 'Salteras', 'Gines', 'Bormujos', 'Castilleja de la Cuesta', 'Castilleja de Guzmán', 'Valencina de la Concepción', 'Santiponce', 'Camas', 'Tomares', 'Amazon SVQ1']

5.2 Resultados del algoritmo de Programación Dinámica

Este algoritmo, como ya se adelantó en el Capítulo 3, apartado 3.2.1, su complejidad es bastante alta, es decir, no es de tipo polinomial, lo que implica que en ocasiones se requiera de tiempos desmesurados para llegar a la solución exacta.

5.2.1 Caso de 23 nodos

En este caso se ha podido comprobar la imposibilidad del algoritmo para llegar a una solución en un tiempo razonable (mayor de un día). En línea con lo dicho anteriormente, la complejidad de este algoritmo aumenta exponencialmente a medida que lo hace el número de nodos del problema que intenta resolver. Esto hace que la capacidad computacional del ordenador donde se esté ejecutando deba ser lo suficientemente grande como para que se pueda obtener una solución exacta en un tiempo razonable.

No por ello este algoritmo deja de perder valor, pues a mostrado buenas soluciones para problemas de menor envergadura.

5.2.2 Caso de 12 nodos

Para un caso menor, al contrario que, en el caso de 23 nodos, el resultado es mejor que en el algoritmo anterior, el de Branch and Bound. La solución es la misma, el óptimo, pero en un tiempo cien veces menor: 1.1030643999 segundos.

La secuencia obtenida es:

['Amazon SVQ1', 'San Juan de Aznalfarache', 'Mairena del Aljarafe', 'Palomares del Río', 'Gelves', 'Santiponce', 'Camas', 'Castilleja de Guzmán', 'Castilleja de la Cuesta', 'Bormujos', 'Sanlúcar la Mayor', 'Tomares', 'Amazon SVQ1']

5.2.3 Caso de 16 nodos

Para este caso la solución obtenida es la óptima en un tiempo breve. Concretamente, la solución es la misma que para el algoritmo de Branch and Bound, la óptima. La longitud del camino es de 196,45 km y el tiempo empleado es de 26'186 segundos, es decir, cien veces más rápido que el de Branch and Bound.

5.3 Resultados del algoritmo de Christofides

Este algoritmo destaca por ser el más rápido de los tres, a continuación, se verán los resultados que lo confirman.

5.3.1 Caso de 23 nodos

Para el mayor caso de los dos, el algoritmo de Christofides da la siguiente secuencia:

['Amazon SVQ1', 'Tomares', 'Camas', 'Bormujos', 'Castilleja de la Cuesta', 'Gines', 'Espartinas', 'Salteras', 'Villanueva del Ariscal', 'Olivares', 'Umbrete', 'Benacazón', 'Sanlúcar la Mayor', 'Bollullos de la Mitación', 'Almensilla', 'Palomares del Río', 'Mairena del Aljarafe', 'San Juan de Aznalfarache', 'Castilleja de Guzmán', 'Valencina de la Concepción', 'Santiponce', 'Gelves', 'Amazon SVQ1']

El tiempo que ha tardado en obtener dicho resultado es de 0.0501314 segundos, y la distancia virtual recorrida en km es de 262'7, mayor que el algoritmo de Branch and Bound. Por último, se muestra en la Figura 30. A simple vista se puede ver que la solución es menos bondadosa que en el algoritmo de Branch and Bound, llegando a pasar por un pueblo no visitado sin llegar a visitarlo. No obstante, en otros puntos del mapa, especialmente más hacia el Sur, sí muestra un comportamiento más bondadoso.

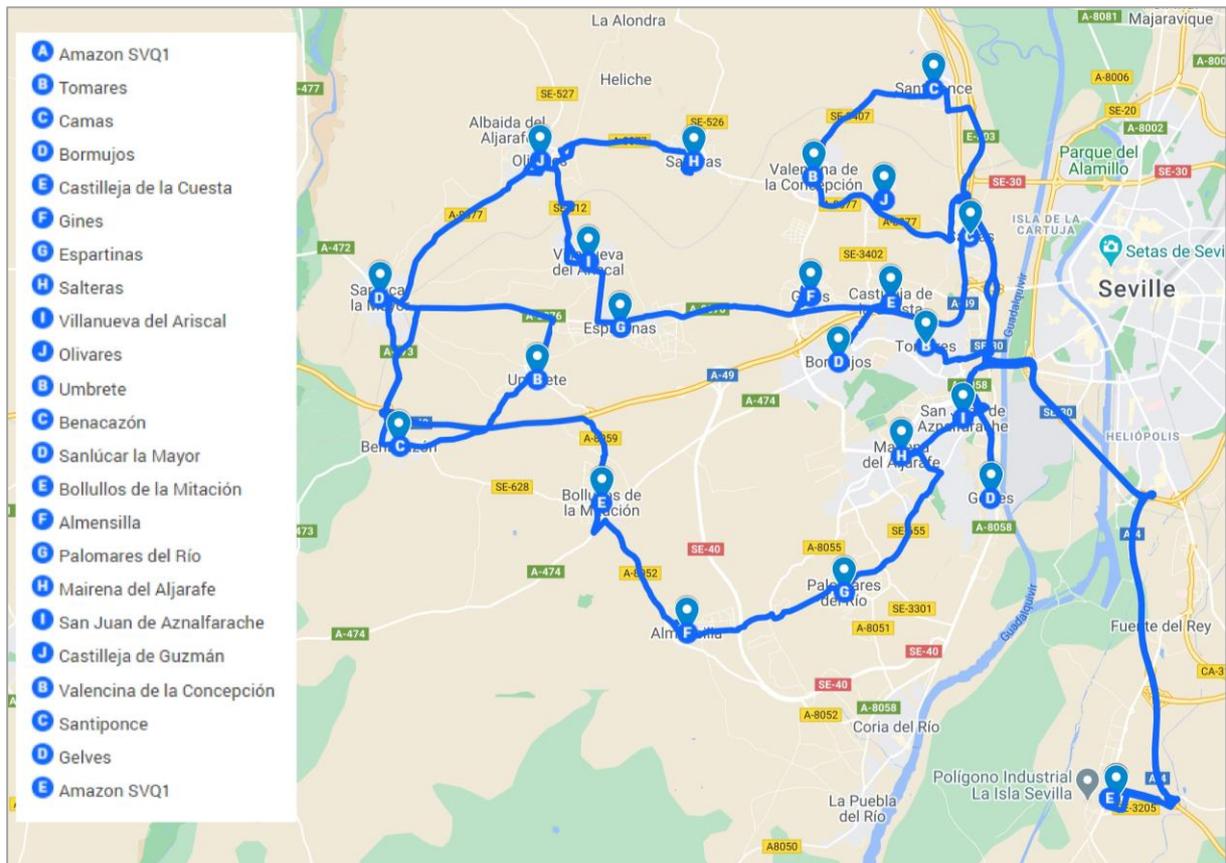


Figura 30: Resolución del problema de los 23 nodos con el algoritmo de Christofides.

5.3.2 Caso de 12 nodos

En este caso la solución de Christofides también vuelve a ser rápida, tarda un tiempo de 0'0495193 segundos con un coste mínimo de 182'2 km. La secuencia obtenida es la siguiente:

['Amazon SVQ1', 'Tomares', 'Castilleja de la Cuesta', 'Bormujos', 'Sanlúcar la Mayor', 'Castilleja de Guzmán', 'Santiponce', 'Camas', 'San Juan de Aznalfarache', 'Mairena del Aljarafe', 'Palomares del Río', 'Gelves', 'Amazon SVQ1'].

Nótese que, efectivamente, el algoritmo de Christofides cumple con la relación de la solución hallada con la solución óptima. Concretamente, para este caso, la secuencia mostrada por el algoritmo es de 1'093 veces superior a la óptima, habiéndola hallado en un tiempo 1'047 veces mayor que para el de Programación dinámica y 2'59 veces mayor que el de Branch and Bound.

5.3.3 Caso de 16 nodos

Por último, se muestra el caso de los 15 pueblos más el centro logístico de Amazon. La solución obtenida es una secuencia cuyo recorrido es de 214'85 km virtuales en un tiempo de 0'1044 segundos. Es decir, se obtiene una solución 1'235 veces peor que la óptima en un tiempo de ejecución 250'92 veces menor que para el de Programación Dinámica. Por otra parte, la solución obtenida sigue esta secuencia:

['Amazon SVQ1', 'Gelves', 'San Juan de Aznalfarache', 'Camas', 'Santiponce', 'Tomares', 'Sanlúcar la Mayor', 'Castilleja de la Cuesta', 'Palomares del Río', 'Castilleja de Guzmán', 'Mairena del Aljarafe', 'Bormujos', 'Valencina de la Concepción', 'Gines', 'Salteras', 'Almensilla', 'Espartinas', 'Bollullos de la Mitación', 'Benacazón', 'Villanueva del Ariscal', 'Olivares', 'Umbrete']

Se muestra claramente la rapidez de este algoritmo a pesar de no llegar al óptimo, lo que supone una elección segura frente a problemas de grandes dimensiones.

5.4 Comparativa

Como se menciona al comienzo del capítulo los tres algoritmos pueden dar una buena imagen del equilibrio entre eficiencia y eficacia, variando según la cantidad de datos procesados. Por ejemplo, para el algoritmo de Programación Dinámica es muy complicado llegar a una solución exacta para grandes entradas del problema en tiempos de amplitud pequeña, ya que el orden de complejidad es exponencial. Sin embargo, tal como se puede apreciar en la Tabla 3 da muy buenos resultados para instancias del problema pequeñas, es decir, es un algoritmo muy bueno para caso en los que el problema contenga muy pocos vértices.

En cuanto al algoritmo de Branch and Bound, es un algoritmo que busca el óptimo para instancias del problema mayores, aunque a un precio temporal elevado. Es decir, para una solución óptima es necesario emplear un tiempo considerablemente alto en la resolución del algoritmo. También se puede ver el caso en el que el algoritmo resuelve una mala instancia de datos (Tabla 3) y en el que el tiempo de ejecución se dispara, como es el caso del subproblema de 16 nodos. Un punto fuerte de este algoritmo es la versatilidad que tiene, se le pueden añadir diferentes tipos de cálculo de límites, estrategias de elección de subproblemas y diferentes formas de ramificación para mejorarlo.

| | 6 nodos | | 12 nodos | | 16 nodos | | 23 nodos | |
|-----------------------|----------------|------------|----------------|------------|----------------|------------|----------------|------------|
| | Distancia (km) | Tiempo (s) |
| Branch and Bound | 78,6 | 0,1009 | 166,700 | 102,1582 | 196,45 | 2785,6820 | 212,7 | 938,870 |
| Programación Dinámica | 78,6 | 0,0408 | 166,700 | 1,1031 | 196,45 | 26,1865 | - | - |
| Christofedes | 81,0 | 0,0390 | 182,200 | 0,0495 | 214,85 | 0,1044 | 262,7 | 0,050 |

Tabla 3: Resultados para diferentes instancias del problema.

Si se pretende hallar soluciones en un corto período de tiempo, el mejor algoritmo es el de Christofides, para todas las estancias del problema ha dado soluciones menores que tres media partes de la solución óptima. Para el de 12 nodos la solución ha sido de 1,09 veces la solución óptima, mientras que, para la instancia completa del problema, la solución dada por Christofides era de 1,235 veces el óptimo (Tabla 4). Esto muestra un gran equilibrio entre eficacia y eficiencia, que se mantienen constante y polinomial a lo largo del dominio de los posibles problemas.

| | 6 nodos | 12 nodos | 23 nodos | 16 nodos |
|---------------------|-------------|-------------|------------|-------------|
| Christofides/Óptima | 1,030534351 | 1,092981404 | 1,09366251 | 1,235072873 |

Tabla 4: Relación del Problema de Christofides con la solución óptima del problema

5.5 Conclusión

Con el presente Proyecto, se ha pretendido dar una solución y enfocar desde un punto de vista de la optimización combinatoria el problema actual frente al que se encuentra el nuevo centro logístico de Amazon en Sevilla, debido a la congestión y a la alta densidad de tráfico en la zona oeste del área metropolitana de la capital hispalense.

Para ello se ha comenzado hablando de la propia empresa, de la evolución de sus ingresos, costes y beneficios. Se ha profundizado en el origen de sus ingresos y gastos, así como en los pilares básicos sobre los que se fundamenta la cultura de la organización. A continuación, se ha explicado mediante el círculo virtuoso de Bezos la estrategia empresarial por la que se mueve la empresa y en la que se remarca la importancia de la reducción de costes. Este punto ha sido fundamental para la motivación de este proyecto.

Junto con la importancia de la minimización de los gastos de la empresa, se ha mostrado la problemática que sufren los corredores de acceso a la provincia de Sevilla. En ellos se ha visto como los puntos de conexión con el oeste del área metropolitana se encuentran congestionados. También, se ha mostrado el impacto que este puede tener en los costes de un usuario medio, lo que se traduce en costes para las empresas que lo atraviesen.

La necesidad de reducir costes por parte de Amazon y los problemas que se dan en la provincia de Sevilla a nivel de tráfico, han motivado el modelado de un problema del viajante en la zona oeste de la capital sevillana con inclusión del centro logístico de Amazon en Dos Hermanas.

A continuación, se ha profundizado en el estado del arte del problema del viajante, dando una visión general sobre las diferentes aplicaciones que puede tener, mostrando así su gran versatilidad. Del mismo modo, se han explicado algunas versiones del problema que muestran su gran adaptación a las diferentes situaciones en los problemas de optimización combinatoria. En este último aspecto, también se ha hablado sobre problemas de naturaleza similar, lo que refuerza la importancia de esta área de las matemáticas y de su empleo para el modelado del Proyecto.

Una vez se ha dado un conocimiento general sobre el tipo de problema matemático y sus variantes, se han mostrado tres formas de solucionarlo mediante el uso de tres algoritmos: dos heurísticas y un algoritmo de fuerza bruta, cada uno con sus propios puntos fuertes y débiles. Explicado el funcionamiento de estos algoritmos, han sido aplicados a la realidad del conjunto de carreteras sevillano.

Para ello, ha sido fundamental el modelado de la zona oeste de Sevilla, empleando una matriz de distancias virtual que tiene en cuenta el tipo de carretera, de manera que esto influye en la

descongestión real de las vías hispalenses. De este modelado ha surgido un grafo de veintitrés nodos que han servido de entrada para el problema del viajante, que a su vez ha sido resuelto mediante los tres algoritmos.

Habiéndose aplicado estos algoritmos al modelado de Sevilla, mediante diferentes instancias del problema. Se han visto las principales diferencias de ellas en la práctica. Con los resultados obtenidos se puede decir que han resuelto satisfactoriamente el problema, teniendo en cuenta su magnitud y la naturaleza de cada algoritmo. El más lento ha sido capaz de encontrar una solución menor que la del resto y óptima para varias instancias del problema, mientras que el más rápido se ha acercado a menos de tres cuartos del primero. Aunque uno de ellos, no haya resuelto el problema completo en un tiempo razonable, sí ha dado buenos resultados para otras instancias menores. El modelado y la aplicación del problema del viajante abren una gran variedad de posibles mejoras al área urbana de Sevilla, así como a los objetivos de reducción de costes de recientes empresas logísticas, como es el caso de Amazon, en la capital hispalense.

6 BIBLIOGRAFÍA

- [1] K. Schept, C. Østergren y P. Reiffer, «Top 100 most valuable brands in 2019,» 2019.
- [2] Beetrack, «Logística de Amazon: analizando al gigante,» Santiago de Chile, 2020.
- [3] «Jeff Bezos - Wikipedia,» [En línea]. Available: https://en.wikipedia.org/wiki/Jeff_Bezos. [Último acceso: 18 10 2020].
- [4] *History of Amazon: From Garage Startup to The Largest E-Commerce Marketplace - Capitalism*, 2020.
- [5] M. Hall, «Amazon.com | History & Facts | Britannica,» 5 2008. [En línea]. Available: <https://www.britannica.com/topic/Amazoncom>.
- [6] M. Research, «Amazon Revenues and Profits Analysis - 2019 Update,» MGM Research, 19 Febrero 2019. [En línea]. Available: <https://mgmresearch.com/amazon-revenues-and-profits-analysis-2019-update/>.
- [7] elmundo.es, «Boeing y Amazon defraudan a Wall Street,» 16 Octubre 2005. [En línea]. Available: https://www.elmundo.es/mundodinero/2005/10/26/mercados_losquemmas/1130337575.html. [Último acceso: 24 Octubre 2020].
- [8] J. Egúsquiza Herrera, S. Gonzales Mendivil y O. Oyardo Manrique, «Planeamiento estratégico Amazon.com.inc. 2019-2023,» Universidad del Pacífico, 2019.
- [9] I. Lunden, «Amazon's share of the US e-commerce market is now 49%, or 5% of all retail spend,» TechCrunch, 3 Julio 2018. [En línea]. Available: <https://techcrunch.com/2018/07/13/amazons-share-of-the-us-e-commerce-market-is-now-49-or-5-of-all-retail-spend/>.
- [10] C. Martín, «Amazon se frota las manos con el Covid.,» hispanidad.com, 30 Octubre 2020. [En

- línea]. Available: https://www.hispanidad.com/confidencial/amazon-se-frota-manos-con-covid-duplica-beneficio-logra-record-en-segundo-trimestre-disparar-40-ventas_12020212_102.html.
- [11] I. Amazon.com, «Amazon.com, Inc. - Amazon.com Announces Third Quarter Results,» Amazon.com, Inc., 29 Octubre 2020. [En línea]. Available: <https://ir.aboutamazon.com/news-release/news-release-details/2020/Amazon.com-Announces-Third-Quarter-Results/default.aspx>.
- [12] J. Kaziukėnas, «Amazon GMV in 2019 - Marketplace Pulse,» 4 Febrero 2020. [En línea]. Available: <https://www.marketplacepulse.com/articles/amazon-gmv-in-2019>.
- [13] A. Pratap, «Costs and Expenses of Amazon - Statstic,» Statstic.com, 23 Agosto 2020. [En línea]. Available: <https://statstic.com/costs-and-expenses-of-amazon/>.
- [14] B. Fox Rubin, «Amazon ya opera siete distintos tipos de tiendas físicas,» CNET en Español, 2 Marzo 2020. [En línea]. Available: <https://www.cnet.com/es/noticias/amazon-siete-tipos-diferentes-de-tiendas-fisicas/>.
- [15] S. Boyd, «Amazon Logistics And Trucking: Virtuous Cycles and The Flywheel Effect,» WorkFutures, 28 Junio 2019. [En línea]. Available: <https://medium.com/work-futures/amazon-logistics-and-trucking-virtuous-cycles-and-the-flywheel-effect-4c1b85a4ca>.
- [16] T. Herrero Alcántara, «El caos organizado del centro logístico de Amazon España,» TICbeat, 5 diciembre 2014. [En línea]. Available: <https://www.ticbeat.com/economia/el-caos-organizado-del-centro-logistico-de-amazon-espana/>.
- [17] M. d. C. Hernández García and J.-Á. Miguel Dávila, "Amazon, el gigante del comercio electrónico y su centro logístico en España," 2015.
- [18] J. Costas, «El Puente del Centenario (Sevilla) se ampliará a seis carriles,» Motor.es, 02 septiembre 2020. [En línea]. Available: <https://www.motor.es/noticias/el-puente-del-quinto-centenario-sevilla-se-ampliara-a-seis-carriles-202070578.html>.
- [19] E. Martos, «La SE-40 convierte a Sevilla en milla de oro del sector logístico,» ABCdesevilla, 19 octubre 2020. [En línea]. Available: <https://sevilla.abc.es/sevilla/sevi-se-40-convierte-sevilla->

milla-sector-logistico-202010190721_noticia.html.

- [20] Ayuntamiento de Sevilla, «El Ayuntamiento culmina el primer diagnóstico integral sobre la movilidad en la ciudad,» www.sevilla.org, 25 abril 2019. [En línea]. Available: <https://www.sevilla.org/actualidad/noticias/el-ayuntamiento-culmina-el-primer-diagnostico-integral-sobre-la-movilidad-en-la-ciudad>.
- [21] Fundación RACC, «La congestión en las vías de acceso a Sevilla,» Barcelona, 2011.
- [22] «TSP History Home,» Natural Sciences and Engineering Research Council of Canada (NSERC); Department of Combinatorics and Optimization at the University of Waterloo, Enero 2007. [En línea]. Available: <http://www.math.uwaterloo.ca/tsp/history/index.html>.
- [23] C. E. Miller, A. W. Tucker y R. A. Zemlin, «Integer Programming Formulation of Traveling Salesman Problems,» *J. ACM*, vol. 7, nº 4, pp. 326-329, 10 1960.
- [24] A. Carpintero Nieto, J. d. D. Díaz Ramírez, V. Franco Otero, P. López Ramos y A. Pérez Montilla, «TSP: Aplicaciones,» Universidad de Cádiz, 2015. [En línea]. Available: <http://knuth.uca.es/moodle/mod/page/view.php?id=3415>.
- [25] R. S. Garfinkel, «Minimizing Wallpaper Waste, Part 1: A Class of Traveling Salesman Problems,» *Operations Research*, vol. 25, nº 5, pp. 741-751, 1977.
- [26] O. B. G. Madsen, «An Application of Travelling-Salesman Routines to Solve Pattern-Allocation Problems in the Glass Industry,» vol. 39, 1988, pp. 249-256.
- [27] T. Pohle, E. Pampalk y G. Widmer, «Generating Similarity-Based Playlists using Traveling Salesman Algorithms,» de *Proceedings of the International Conference on Digital Audio Effects 8th*, Madrid, 2005.
- [28] E. L. Lawler, J. K. Lenstra, A. H. G. R. Kan y D. B. Shmoys, *The Traveling Salesman Problem: A Guided Tour of Combinatorial Optimization*, 4 ed., John Wiley & Sons Inc, 1985, p. 476.
- [29] R. W. Floyd, «Algorithm 97: Shortest Path,» *Commun. ACM*, vol. 5, nº 6, p. 345, 6 1962.
- [30] R. S. Garfinkel y K. C. Gilbert, «The Bottleneck Traveling Salesman Problem: Algorithms and Probabilistic Analysis,» *J. ACM*, vol. 25, nº 3, pp. 435-448, 7 1978.

- [31] K. R. Fox, B. Gavish y S. C. Graves, «An n-Constraint Formulation of the (Time-Dependent) Traveling Salesman Problem,» vol. 28, INFORMS, 1980, pp. 1018-1021.
- [32] J.-C. Picard y M. Queyranne, «The Time-Dependent Traveling Salesman Problem and Its Application to the Tardiness Problem in One-Machine Scheduling,» de *Operations Research*, vol. 26, INFORMS, 1978, pp. 86-110.
- [33] D. Feillet, P. Dejax y M. Gendreau, «Traveling Salesman Problems with Profits,» de *Transportation Science*, vol. 39, INFORMS, 2005, pp. 188-205.
- [34] G. Dantzig, R. Fulkerson y S. Johnson, «Solution of a Large-Scale Traveling-Salesman Problem,» *Journal of the Operations Research Society of America*, vol. 2, n° 4, pp. 393-410, 1954.
- [35] T. C. Koopmans y M. Beckmann, «Assignment Problems and the Location of Economic Activities,» *Econometrica*, vol. 25, n° 1, pp. 53-76, 1957.
- [36] A. Schrijver, «Combinatorial Optimization: Polyhedra and Efficiency, Volume 1,» de *Combinatorial Optimization: Polyhedra and Efficiency, Volume 1*, Berlín, SPRINGER-VERLAG, 2003, p. 114.
- [37] W. W. Hardgrave y G. L. Nemhauser, «On the Relation Between the Traveling-Salesman and the Longest-Path Problems,» *Operations Research*, vol. 10, n° 5, pp. 647-657, 10 1962.
- [38] T. H. Cormen, C. E. Leiserson, R. L. Rivest y C. Stein, *Introduction to Algorithms*, Third Edition, Cambridge, UNITED STATES: MIT Press, 2009.
- [39] M. Guan, «Graphic Programming using Odd or Even Points,» de *Chinese Math. I*, 1962, p. 273–277.
- [40] L. Euler, «Solutio problematis ad geometriam situs pertinentis,» de *Commentarii academiae scientiarum Petropolitanae*, 1741, pp. 128-140.
- [41] E. Y. Gabovich, «The small traveling salesman problem,» *Trudy Vychisl. Centra Tartu. Gos. Univ*, vol. 19, pp. 17-51, 1970.

- [42] C.-Y. (. Huang, C.-Y. Lai y K.-T. (. Cheng, «Chapter 4 - Fundamentals of algorithms,» de *Electronic Design Automation*, L. Wang, Y. Chang y K. (. Cheng, Edits., Boston, Morgan Kaufmann, 2009, pp. 173 - 234.
- [43] M. Held y R. M. Karp, «A Dynamic Programming Approach to Sequencing Problems,» vol. 10, Society for Industrial and Applied Mathematics, 1962, pp. 199-200.
- [44] N. Christofides, «Worst-case analysis of a new heuristic for the travelling salesman problem,» Technical Report, Pittsburgh, Pennsylvania, 1976.
- [45] S. Jaeger, «Recovering Dynamic Information from Static, Handwritten Word Images,» 12 1998.
- [46] F. J. Veiga Losada, «Trabajo de Fin de Máster: El problema del viajante.,» Santiago de Compostela, 2013.
- [47] G. D. Valle, «Amazon 2-day shipping: why packages sometimes arrive later,» Vox.com, 24 Abril 2019. [En línea]. Available: <https://www.vox.com/the-goods/2019/4/23/18508093/amazon-prime-two-day-shipping>.
- [48] F. Howard, «La cadena logística de Amazon.com,» Driv.in, 20 Septiembre 2017. [En línea]. Available: <https://blog.driv.in/es/la-cadena-logistica-de-amazon-com/>.
- [49] R. E. Burkard, «Travelling Salesman and Assignment Problems: A Survey,» de *Discrete Optimization I, Annals of Discrete Mathematics*, vol. 4, P. L. Hammer, E. L. Johnson y B. H. Korte, Edits., Elsevier, 1979, pp. 193-215.
- [50] A. H. Land y A. G. Doig, «An Automatic Method of Solving Discrete Programming Problems,» *Econometrica*, vol. 28, nº 3, pp. 497-520, 1960.

ANEXO I: MATRIZ DE COSTES

| | CodMun | 41038 | 41044 | 41086 | 41021 | 41089 |
|---------------|----------------------------|-------------|--------|----------------|-------|------------|
| CodMun | Municipio | Amazon SVQ1 | Gelves | San Juan de Az | Camas | Santiponce |
| 41038 | Amazon SVQ1 | 0 | 24 | 20 | 23,8 | 25,7 |
| 41044 | Gelves | 24 | 0 | 8,4 | 13 | 13,8 |
| 41086 | San Juan de Aznalfarache | 20 | 8,4 | 0 | 13,2 | 15 |
| 41021 | Camas | 23,8 | 13 | 13,2 | 0 | 10,8 |
| 41089 | Santiponce | 25,7 | 13,8 | 15 | 10,8 | 0 |
| 41093 | Tomares | 20 | 11 | 6,8 | 5,6 | 12,5 |
| 41087 | Sanlúcar la Mayor | 41,7 | 35,1 | 29,8 | 26,1 | 31,9 |
| 41029 | Castilleja de la Cuesta | 22,9 | 11,5 | 12,4 | 7,2 | 13 |
| 41070 | Palomares del Río | 35 | 17,4 | 13,2 | 31,9 | 29,2 |
| 41028 | Castilleja de Guzmán | 29 | 14,8 | 18,4 | 7,6 | 13,4 |
| 41059 | Mairena del Aljarafe | 24,4 | 11,2 | 4 | 10 | 18,4 |
| 41017 | Bormujos | 25,7 | 13,2 | 10 | 9,6 | 15,3 |
| 41096 | Valencina de la Concepción | 33,4 | 26,6 | 22,8 | 11,8 | 10,4 |
| 41047 | Gines | 27,4 | 15,3 | 16,1 | 11,8 | 16,2 |
| 41085 | Salteras | 39,2 | 32,4 | 39,2 | 17,8 | 19 |
| 41010 | Almensilla | 43,2 | 26,8 | 20,8 | 28,9 | 33,4 |
| 41040 | Espartinas | 34,8 | 28 | 23,5 | 19,8 | 25,6 |
| 41016 | Bollullos de la Mitación | 35,4 | 16,8 | 23,3 | 19,8 | 25,4 |
| 41015 | Benacazón | 38,1 | 31,6 | 20,4 | 22,5 | 28,3 |
| 41098 | Villanueva del Ariscal | 39,3 | 32,6 | 43,5 | 24,6 | 29,9 |
| 41067 | Olivares | 52,3 | 45,7 | 40,2 | 27,6 | 28,8 |
| 41094 | Umbrete | 32,7 | 26,2 | 20,8 | 17,1 | 22,9 |

| | CodMun | 41093 | 41087 | 41029 | 41070 | 41028 |
|---------------|----------------------------|---------|---------------|---------------------|--------------|-------------------|
| CodMun | Municipio | Tomares | Sanlúcar la M | Castilleja de la Cu | Palomares de | Castilleja de Guz |
| 41038 | Amazon SVQ1 | 20 | 41,7 | 22,9 | 35 | 29 |
| 41044 | Gelves | 11 | 35,1 | 11,5 | 17,4 | 14,8 |
| 41086 | San Juan de Aznalfarache | 6,8 | 29,8 | 12,4 | 13,2 | 18,4 |
| 41021 | Camas | 5,6 | 26,1 | 7,2 | 31,9 | 7,6 |
| 41089 | Santiponce | 12,5 | 31,9 | 13 | 29,2 | 13,4 |
| 41093 | Tomares | 0 | 25,3 | 6,4 | 21 | 12,2 |
| 41087 | Sanlúcar la Mayor | 25,3 | 0 | 25,1 | 37,55 | 31,3 |
| 41029 | Castilleja de la Cuesta | 6,4 | 25,1 | 0 | 12,8 | 6,1 |
| 41070 | Palomares del Río | 21 | 37,55 | 12,8 | 0 | 28 |
| 41028 | Castilleja de Guzmán | 12,2 | 31,3 | 6,1 | 28 | 0 |
| 41059 | Mairena del Aljarafe | 9,8 | 32,5 | 12,2 | 11,6 | 21,3 |
| 41017 | Bormujos | 9,4 | 25,7 | 4,4 | 17,8 | 10 |
| 41096 | Valencina de la Concepción | 16,7 | 29,2 | 12 | 29,4 | 4,8 |
| 41047 | Gines | 10,7 | 25,4 | 4,9 | 23,2 | 9,2 |
| 41085 | Salteras | 22,4 | 22 | 18 | 35,2 | 12 |
| 41010 | Almensilla | 26,4 | 33,55 | 24,9 | 9,6 | 28,6 |
| 41040 | Espartinas | 10,7 | 13 | 14,8 | 34,4 | 19,4 |
| 41016 | Bollullos de la Mitación | 18,7 | 20,4 | 15,1 | 22,6 | 20,8 |
| 41015 | Benacazón | 21,4 | 10 | 17,8 | 33 | 23,5 |
| 41098 | Villanueva del Ariscal | 23,5 | 16,4 | 19,4 | 37,2 | 24 |
| 41067 | Olivares | 33,2 | 13 | 27,8 | 48,7 | 21,8 |
| 41094 | Umbrete | 16 | 13,6 | 12,4 | 27,8 | 17,8 |

| | CodMun | 41059 | 41017 | 41096 | 41047 | 41085 |
|---------------|----------------------------|---------------|----------|-----------------|-------|----------|
| CodMun | Municipio | Mairena del A | Bormujos | Valencina de la | Gines | Salteras |
| 41038 | Amazon SVQ1 | 24,4 | 25,7 | 33,4 | 27,4 | 39,2 |
| 41044 | Gelves | 11,2 | 13,2 | 26,6 | 15,3 | 32,4 |
| 41086 | San Juan de Aznalfarache | 4 | 10 | 22,8 | 16,1 | 39,2 |
| 41021 | Camas | 10 | 9,6 | 11,8 | 11,8 | 17,8 |
| 41089 | Santiponce | 18,4 | 15,3 | 10,4 | 16,2 | 19 |
| 41093 | Tomares | 9,8 | 9,4 | 16,7 | 10,7 | 22,4 |
| 41087 | Sanlúcar la Mayor | 32,5 | 25,7 | 29,2 | 25,4 | 22 |
| 41029 | Castilleja de la Cuesta | 12,2 | 4,4 | 12 | 4,9 | 18 |
| 41070 | Palomares del Río | 11,6 | 17,8 | 29,4 | 23,2 | 35,2 |
| 41028 | Castilleja de Guzmán | 21,3 | 10 | 4,8 | 9,2 | 12 |
| 41059 | Mairena del Aljarafe | 0 | 9,4 | 20,8 | 14,6 | 26,6 |
| 41017 | Bormujos | 9,4 | 0 | 13,8 | 7,6 | 19,6 |
| 41096 | Valencina de la Concepción | 20,8 | 13,8 | 0 | 8,6 | 21,8 |
| 41047 | Gines | 14,6 | 7,6 | 8,6 | 0 | 13,6 |
| 41085 | Salteras | 26,6 | 19,6 | 21,8 | 13,6 | 0 |
| 41010 | Almensilla | 17,8 | 21 | 33,2 | 24 | 36,2 |
| 41040 | Espartinas | 22 | 15 | 20,2 | 10,4 | 7,8 |
| 41016 | Bollullos de la Mitación | 23,4 | 16 | 24,2 | 14,6 | 27,2 |
| 41015 | Benacazón | 25,8 | 18,8 | 26,8 | 17,4 | 29,2 |
| 41098 | Villanueva del Ariscal | 26,6 | 19,6 | 22,6 | 15,2 | 22,6 |
| 41067 | Olivares | 36,6 | 28,6 | 17,6 | 23,4 | 17,6 |
| 41094 | Umbrete | 20,8 | 11 | 21,8 | 12 | 27 |

| | CodMun | 41010 | 41040 | 41016 | 41015 | 41098 |
|---------------|----------------------------|--------------|------------|-----------------|-----------|---------------|
| CodMun | Municipio | Almensevilla | Espartinas | Bollullos de la | Benacazón | Villanueva de |
| 41038 | Amazon SVQ1 | 43,2 | 34,8 | 35,4 | 38,1 | 39,3 |
| 41044 | Gelves | 26,8 | 28 | 16,8 | 31,6 | 32,6 |
| 41086 | San Juan de Aznalfarache | 20,8 | 23,5 | 23,3 | 20,4 | 43,5 |
| 41021 | Camas | 28,9 | 19,8 | 19,8 | 22,5 | 24,6 |
| 41089 | Santiponce | 33,4 | 25,6 | 25,4 | 28,3 | 29,9 |
| 41093 | Tomares | 26,4 | 10,7 | 18,7 | 21,4 | 23,5 |
| 41087 | Sanlúcar la Mayor | 33,55 | 13 | 20,4 | 10 | 16,4 |
| 41029 | Castilleja de la Cuesta | 24,9 | 14,8 | 15,1 | 17,8 | 19,4 |
| 41070 | Palomares del Río | 9,6 | 34,4 | 22,6 | 33 | 37,2 |
| 41028 | Castilleja de Guzmán | 28,6 | 19,4 | 20,8 | 23,5 | 24 |
| 41059 | Mairena del Aljarafe | 17,8 | 22 | 23,4 | 25,8 | 26,6 |
| 41017 | Bormujos | 21 | 15 | 16 | 18,8 | 19,6 |
| 41096 | Valencina de la Concepción | 33,2 | 20,2 | 24,2 | 26,8 | 22,6 |
| 41047 | Gines | 24 | 10,4 | 14,6 | 17,4 | 15,2 |
| 41085 | Salteras | 36,2 | 7,8 | 27,2 | 29,2 | 22,6 |
| 41010 | Almensevilla | 0 | 24,6 | 10,2 | 22,6 | 27,4 |
| 41040 | Espartinas | 24,6 | 0 | 15,6 | 17,4 | 5,2 |
| 41016 | Bollullos de la Mitación | 10,2 | 15,6 | 0 | 13 | 18 |
| 41015 | Benacazón | 22,6 | 17,4 | 13 | 0 | 22 |
| 41098 | Villanueva del Ariscal | 27,4 | 5,2 | 18 | 22 | 0 |
| 41067 | Olivares | 37,9 | 14,2 | 28,5 | 20,6 | 5,6 |
| 41094 | Umbrete | 17,8 | 14 | 8,4 | 11,6 | 10,4 |

| | CodMun | 41067 | 41094 |
|---------------|----------------------------|----------|---------|
| CodMun | Municipio | Olivares | Umbrete |
| 41038 | Amazon SVQ1 | 52,3 | 32,7 |
| 41044 | Gelves | 45,7 | 26,2 |
| 41086 | San Juan de Aznalfarache | 40,2 | 20,8 |
| 41021 | Camas | 27,6 | 17,1 |
| 41089 | Santiponce | 28,8 | 22,9 |
| 41093 | Tomares | 33,2 | 16 |
| 41087 | Sanlúcar la Mayor | 13 | 13,6 |
| 41029 | Castilleja de la Cuesta | 27,8 | 12,4 |
| 41070 | Palomares del Río | 48,7 | 27,8 |
| 41028 | Castilleja de Guzmán | 21,8 | 17,8 |
| 41059 | Mairena del Aljarafe | 36,6 | 20,8 |
| 41017 | Bormujos | 28,6 | 11 |
| 41096 | Valencina de la Concepción | 17,6 | 21,8 |
| 41047 | Gines | 23,4 | 12 |
| 41085 | Salteras | 17,6 | 27 |
| 41010 | Almensevilla | 37,9 | 17,8 |
| 41040 | Espartinas | 14,2 | 14 |
| 41016 | Bollullos de la Mitación | 28,5 | 8,4 |
| 41015 | Benacazón | 20,6 | 11,6 |
| 41098 | Villanueva del Ariscal | 5,6 | 10,4 |
| 41067 | Olivares | 0 | 23 |
| 41094 | Umbrete | 23 | 0 |