

Trabajo Fin de Máster

El problema del viajante. Reparto de costes con juegos cooperativos

Tania Bordón Ramos

Máster en Técnicas Estadísticas

Curso 2022 – 2023

Propuesta de Trabajo Fin de Máster

Título en galego: O problema do viaxeiro. Reparto de custos con xogos cooperativos.

Título en español: El problema del viajante. Reparto de costes con juegos cooperativos

English title: The traveling salesman problem. Cost sharing with cooperative games

Modalidad: Modalidad A

Autor/a: Tania Bordón Ramos, Universidad de A Coruña

Director/a: María Estela Sánchez Rodríguez, Universidad de Vigo

Breve resumen del trabajo:

En este trabajo se aborda el problema del viajante (TSP) desde sus orígenes, examinando los diversos tipos de TSP que existen en la actualidad, así como sus formulaciones y las diversas opciones para resolverlo. Además, se analiza el reparto de costes utilizando soluciones de teoría de Juegos Cooperativos, considerando el juego de ruta fija, el juego de ruta fija con citas y el clásico juego del viajante de comercio. Por último, se presenta un ejemplo práctico que ilustra las diferentes versiones del problema del viajante estudiadas.

Doña M. Estela Sánchez Rodríguez, titular de universidad, de la Universidad de Vigo, informa de que el Trabajo Fin de Máster titulado

El problema del viajante. Reparto de costes con juegos cooperativos

fue realizado bajo su dirección por doña Tania Bordón Ramos para el Máster en Técnicas Estadísticas. Estimando que el trabajo está terminado, da su conformidad para su presentación y defensa ante un tribunal.

En A Coruña, a 09 de Julio de 2023.

La directora:
María Estela Sánchez Rodríguez

La autora:
Tania Bordón Ramos

Declaración responsable. Para dar cumplimiento a la Ley 3/2022, de 24 de febrero, de convivencia universitaria, referente al plagio en el Trabajo Fin de Máster de 24 de febrero, de convivencia universitaria, referente al plagio en el Trabajo Fin de Máster (Artículo 11, [Disposición 2978 del BOE núm. 48 de 2022](#)), la autora declara que el Trabajo Fin de Máster presentado es un documento original en el que se han tenido en cuenta las siguientes consideraciones relativas al uso de material de apoyo desarrollado por otros/as autores/as:

- Todas las fuentes usadas para la elaboración de este trabajo han sido citadas convenientemente (libros, artículos, apuntes de profesorado, páginas web, programas, ...)
- Cualquier contenido copiado o traducido textualmente se ha puesto entre comillas, citando su procedencia.
- Se ha hecho constar explícitamente cuando un capítulo, sección, demostración, sea una adaptación casi literal de alguna fuente existente.

Y, acepta que, si se demostrara lo contrario, se le apliquen las medidas disciplinarias que correspondan.

Agradecimientos

En primer lugar, me gustaría agradecer a mi familia y, en particular, a mis padres y mis abuelos, por el apoyo constante recibido, por inculcarme los valores que hacen que sea la persona que soy ahora, trabajadora y responsable. Por ser mi gran pilar. Gracias por darme la oportunidad de poder tener una educación y la posibilidad de seguir, hoy en día, formándome. No tendré tiempo suficiente para agradecerlos todo.

También me gustaría agradecer a mi tutora, Estela, por haberme guiado y ayudado durante todo el proceso de la elaboración de este trabajo, sobre todo en la parte final de este. Asimismo, gracias por adaptarse a mis tiempos, sé que no ha sido nada fácil, pero por fin, ha llegado el final.

Por otro lado, me gustaría hacer un agradecimiento a mis amigas, por aguantar mis peores días, mis audios por WhatsApp interminables y, de normal, repetitivos, y por comprenderme tanto. En especial, a Elena, Claudia y mis "SPICE". Gracias por ayudarme, por leerlos las cosas aún sin entenderlo, por esas videollamadas interminables y por esos ánimos constantes.

A Claudia, mi compañera fiel del máster. Sólo tú y yo sabemos lo que nos ha costado llegar hasta aquí. A pesar de la distancia, has sido un apoyo fundamental en esta etapa y te estaré agradecida siempre.

Por último, y no menos importante, a Montse. Gracias por acompañarme durante esta etapa. Por hacerme el camino más fácil, no sólo con el trabajo de fin de máster, sino también, conmigo misma. No ha sido un año fácil, pero gracias a tus sesiones todo ha sido más llevadero y he podido disfrutar del proceso, algo que hace un año veía improbable. Gracias siempre.

Índice general

Resumen	15
El problema del viajante	18
1.1. Introducción	18
1.2. Tipos de TSP.....	24
1.3. Formulación.....	25
1.3.1. El problema del viajante simétrico	29
1.4. Métodos de resolución del TSP	33
1.5. Concorde TSP Solver.....	41
Reparto de costes con juegos cooperativos	47
2.1. Introducción	47
2.2. Definiciones previas	49
2.3. Los juegos cooperativos en el caso del TSP	58
2.3.1 El juego del viajante con ruta fija.....	66
2.3.2 El juego del viajante con ruta fija y con citas.....	75
2.3.3 El juego del viajante de comercio.....	83
Caso práctico: Reparto de costes mediante Juegos Cooperativos: El problema del viajante de Castilla y León.....	87
3.1. Introducción	87
3.2. El problema del viajante de Castilla y León.	88
3.2.1. El juego del viajante de Castilla y León con ruta fija.	98
3.2.2. El juego del viajante de Castilla y León con ruta fija y con citas.	104
3.2.3. El juego del viajante de comercio de Castilla y León.	109
3.3. Conclusiones.....	110

Bibliografía.....	112
Anexo I.....	117
Anexo II.....	120
Anexo III.....	121
Anexo IV	122

Índice de figuras

Figura 1. Relación entre un problema NP-duro y NP-completo.....	20
Figura 2. Tipos de grafos	26
Figura 3. Matriz de costes simétrica	27
Figura 4. Grafo no dirigido correspondiente a la matriz de costes de la Figura 3.	28
Figura 5. Matriz de costes asimétrica	29
Figura 6. Grafo dirigido correspondiente a la matriz de costes de la figura 5	29
Figura 7. Grafo no dirigido. Ejemplo del problema del viajante simétrico.....	31
Figura 8 Programa concorde. Fuente: Applegate D., Bixby R.E., Chvátal V. y Cook W. J.	42
Figura 9. Salida de código del servidor neos	43
Figura 10. Evolución del número de ciudades resueltas en un TSP a lo largo de la historia. Fuente: Elaboración propia en base a los datos expuestos por Infantes Durán (2018)	45
Figura 11. Grafo no dirigido.....	50
Figura 12. Costes marginales de las ciudades.	51
Figura 13. Grafo de un juego TU. Fuente: Derks y Kuipers(1997)	59
Figura 14. Core obtenido del Caso 1. Fuente: elaboración propia mediante TUGLabWeb.	61
Figura 15. Ejemplo propuesto por Fishburn and Pollak (1983)	70
Figura 16. Representación gráfica del Core del juego propuesto. Fuente: Fishburn and Pollak (1983)	72
Figura 17. Grafo asociado a la ruta.....	80
Figura 18. Grafo asociado a la ruta restringida a s	80
Figura 19. Representación del core del ejemplo del juego del viajante de comercio.....	86
Figura 20. Grafo de la ruta del viajante de Castilla y león.	90

Índice de tablas

Tabla 1. Posibles rutas y sus costes de la figura 7.	32
Tabla 2. Número de ciudades resueltas en cada tsp. fuente: infantes durán (2018)	45
Tabla 3. Rutas y costes obtenidos de la figura 11.	50
Tabla 4. Nomenclatura del problema del viajante de castilla y león	89

Tabla 5. Costes del problema del viajante de Castilla y León.	90
Tabla 6. Desigualdades triangulares de la matriz de costes del problema del viajante de Castilla y León.	92
Tabla 7. Costes individuales del problema del viajante de Castilla y León.	92
Tabla 8. Costes de las coaliciones formadas por dos provincias del problema del viajante de Castilla y León.	94
Tabla 9. Costes de las coaliciones formadas por tres provincias del problema del viajante de Castilla y León.	95
Tabla 10. Costes de las coaliciones formadas por cuatro provincias del problema del viajante de Castilla y León.	96
Tabla 11. Costes de las coaliciones formadas por cinco provincias del problema del viajante de Castilla y León.	97
Tabla 12 . Costes de las coaliciones formadas por tres provincias del problema del viajante de castilla y león con ruta fija.	99
Tabla 13. Costes de las coaliciones formadas por cuatro provincias del problema del viajante de castilla y león con ruta fija	100
Tabla 14. Costes de las coaliciones formadas por cinco provincias del problema del viajante de Castilla y León con ruta fija.	100
Tabla 15. Costes de las coaliciones formadas por tres provincias del problema del viajante de Castilla y León con ruta fija y ruta óptima.	102
Tabla 16. Costes de las coaliciones formadas por cuatro provincias del problema del viajante de castilla y león con ruta fija y ruta óptima	103
Tabla 17. Costes de las coaliciones formadas por cinco provincias del problema del viajante de castilla y león con ruta fija y ruta óptima	103
Tabla 18. Costes de las coaliciones formadas por dos provincias del problema del viajante de castilla y león con ruta fija y con citas.	106
Tabla 19. Costes de las coaliciones formadas por tres provincias del problema del viajante de castilla y león con ruta fija y con citas	107
Tabla 20. Costes de las coaliciones formadas por cuatro provincias del problema del viajante de castilla y león con ruta fija y con citas.	107
Tabla 21. Costes de las coaliciones formadas por cinco provincias del problema del viajante de castilla y león con ruta fija y con citas.	108

Resumen

Resumen en español

En principio, el problema del viajante, así como otros temas dentro del campo de las matemáticas combinatorias, puede parecer un tema complejo, abstracto y de difícil aplicación a la “vida real”. Sin embargo, a lo largo de toda nuestra vida, hemos ido aplicando el problema del viajante a nuestro día a día, muchas veces sin ser conscientes de ello. Un ejemplo de ello puede ser a la hora de organizar un viaje, cuyo objetivo, suele ser, obtener un recorrido mediante el cual recorramos todos los lugares a visitar en el menor tiempo posible o en un tiempo determinado. Es cierto que en este caso no utilizamos ninguna herramienta matemática para ello, pero la idea básica de lo que es y en lo que consiste el problema del viajante está presente en estas decisiones. Por otro lado, en lo que respecta a los juegos cooperativos, como bien dice la palabra, consiste en la cooperación de los jugadores para obtener un beneficio común. Un ejemplo ilustrativo de este concepto es la aplicación BlaBlaCar. Esta plataforma facilita la colaboración entre un grupo de personas que comparten el mismo destino, permitiéndoles compartir un vehículo y los costes del trayecto.

Como vemos, son muchas ocasiones en las que, sin ser conscientes de ello, estamos ante un problema del viajante. Es por ello por lo que hemos querido realizar un estudio exhaustivo sobre el problema del viajante de comercio y el reparto de costes de este mediante los juegos cooperativos.

En el presente trabajo comenzaremos, en los primeros epígrafes, con una breve introducción sobre el problema del viajante y las diferentes versiones de este en la actualidad, así como sus formulaciones. Analizaremos también, las herramientas utilizadas para su resolución y los diferentes resultados obtenidos a lo largo de la historia.

Seguidamente, se estudiarán los distintos tipos de juegos del problema del viajante: el juego del viajante de comercio, el juego del viajante de comercio con ruta fija y el juego del viajante de comercio con ruta fija y con citas, así como un pequeño ejemplo práctico de cada uno de ellos donde obtendremos algunas de las principales soluciones de los juegos cooperativos.

Para finalizar, realizaremos un caso práctico. Estudiaremos el caso del problema de un viajante recorriendo distintas provincias de Castilla y León con el objetivo de minimizar el coste de dicha ruta. Para ello, calcularemos las principales soluciones de los juegos cooperativos y, tras el análisis de los resultados obtenidos, realizaremos las conclusiones pertinentes.

English Summary

Initially, the traveling salesman problem, as well as other topics in the field of combinatorial mathematics, may appear complex, abstract, and challenging to apply in real life. However, throughout our lives, we have been applying the traveling salesman problem in our daily routines, often without even realizing it. An example of this is when organizing a trip, where the objective is typically to plan a route that allows us to visit all desired locations in the least amount of time or within a specified timeframe. While we may not explicitly employ mathematical tools in such cases, the fundamental idea and essence of the traveling salesman problem are present in these decision-making processes.

On the other hand, regarding cooperative games, as the term implies, they involve the cooperation of players to achieve a common benefit. An illustrative example of this concept is the BlaBlaCar application. This platform facilitates collaboration among a group of individuals who share the same destination, enabling them to share a vehicle and the costs of the journey.

As we can see, there are many instances where we unknowingly encounter the traveling salesman problem. This is why we have conducted a comprehensive study on the traveling salesman problem and the distribution of costs through cooperative games.

In this work, we will begin with a brief introduction to the traveling salesman problem and its versions, as well as their formulations. Additionally, we will analyze the tools employed for its resolution and explore the diverse outcomes obtained throughout history.

Next, we will explore the different types of games based on the traveling salesman problem: the traveling salesman game, the traveling salesman game with a fixed route, and the traveling salesman game with a fixed route and appointments. Furthermore, we will provide a brief practical example for each of these games, where we will obtain some of the main solutions from cooperative games.

To conclude, we will present a practical case study. We will examine the scenario of a salesman traveling through various cities of Castilla y León with the objective of minimizing the cost of the route. To achieve this, we will calculate the basic solutions using cooperative games, and after analyzing the results obtained, we will draw the relevant conclusions.

Capítulo 1

El problema del viajante

1.1. Introducción

En base a lo que comenta Davendra (2010), el problema del viajante (conocido en inglés como “Traveling Salesman Problem” con las siglas TSP) consiste en encontrar, dado un número de ciudades y un coste de viaje (o distancia) entre cada una de ellas, una ruta por la cual el viajante visite cada ciudad regresando al lugar de partida, pasando una vez por cada una de las ciudades y minimizando el coste del trayecto (o distancia) de dicha ruta.

A pesar de que, a simple vista, el problema del viajante parece un problema de fácil resolución, esto queda lejos de la realidad, tanto que, hoy en día, en base a lo que se define en teoría de la complejidad, se considera el TSP como un problema *NP-duro*, es decir, no se puede encontrar su solución óptima en un tiempo polinómico debido a la dificultad para su resolución cuando el tamaño del problema es muy grande. Así, una de las opciones para resolver el TSP es mediante los diferentes algoritmos existentes.

Podemos definir un algoritmo como un conjunto de reglas o instrucciones definidas, no-ambiguas, ordenadas y finitas que permiten, normalmente y entre otros, solucionar un problema, procesar datos o llevar a cabo otras tareas o actividades. A la hora de resolver un problema, uno de los aspectos que más nos interesa es que dicho problema se resuelva de manera óptima y, en el menor tiempo posible. Así, el tiempo que tarda un algoritmo en procesar o ejecutar el código y obtener una solución, es lo que definimos como la complejidad de tiempo.

Definimos formalmente la complejidad de tiempo de un algoritmo como $O(g(n))$ si existe una constante $c > 0$ y $N \in \mathbb{N}$ tal que su tiempo de computación $f(n)$ satisface $f(n) \leq cg(n)$ para todo $n \geq N$, siendo n el tamaño del problema a resolver.

A lo largo de la historia, se han utilizado los algoritmos para resolver diferentes tipos de problemas, entre ellos los problemas de decisión binaria. Los problemas de decisión

binaria se pueden definir como problemas dónde las únicas respuestas posibles son “sí” o “no”.

En la teoría de la complejidad, existen diferentes clases de complejidad de los problemas de decisión. En este trabajo, explicaremos brevemente dos tipos: *clase de complejidad P* y *clase de complejidad NP*.

Dado un problema de decisión P diremos que pertenece a la *clase de complejidad P* si existe un algoritmo que resuelva cualquier problema en tiempo polinomial. En este tipo de problemas, su función de complejidad ($f(n)$) satisface que $f(n) = O(n^c)$. Si esta igualdad se cumple, podemos concluir con que el algoritmo es eficiente. Un ejemplo de este tipo de clase es el problema del recorrido de Euler. Un recorrido de Euler es un recorrido o camino que recorre o visita cada arista del grafo exactamente una vez.

Por otro lado, un problema P pertenece a la *clase de complejidad NP* (siendo dichas siglas la abreviatura de tiempo polinómico no determinista) si existe un algoritmo que permite verificar cualquier ejemplo en tiempo polinómico por una máquina de Turing¹ no determinista. Cuando el problema es NP podemos recurrir a algoritmos aproximados o a algoritmos heurísticos (que serán explicados, ambos, detalladamente en los siguientes epígrafes). Dentro de la *complejidad de tiempo polinómico no determinista*, podemos encontrar dos clases de problemas: NP-completo y NP-duro.

Podemos decir que un problema P es NP-completo si cualquier problema en NP puede ser reducido a P en tiempo polinomial. Un ejemplo de un problema NP-completo es lo que en teoría de grafos se conoce como el problema del ciclo o camino hamiltoniano.

El problema del ciclo o camino hamiltoniano en un grafo G es un camino que visita todos los vértices del grafo una sola vez coincidiendo el primer y último vértice visitado. Así, ante una posible solución para verificar si es un ciclo hamiltoniano tenemos una complejidad de tiempo igual a $O(n)$ siendo n el conjunto total de nodos. Sin embargo, si utilizamos un algoritmo de fuerza bruta, la complejidad de tiempo sería igual a $O(n!)$. A

¹ Dispositivo que manipula símbolos sobre una tira de cinta de acuerdo con una tabla de reglas.

pesar de todos los estudios realizados, aún no se conoce ningún algoritmo en tiempo polinomial que sea capaz de resolver el problema del ciclo o camino hamiltoniano.

Diremos que un problema es NP-duro si cualquier problema en la clase NP se puede reducir a él en tiempo polinomial.

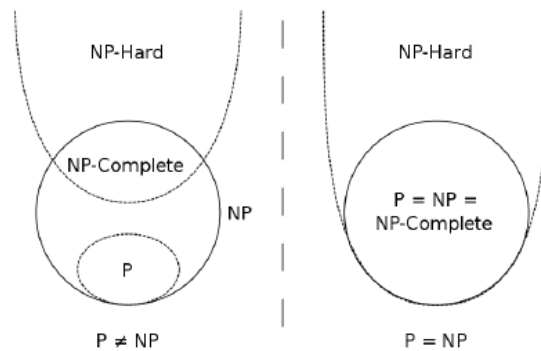


FIGURA 1. RELACIÓN ENTRE UN PROBLEMA NP-DURO Y NP-COMPLETO

Un ejemplo de problema NP-completo es el problema de la mochila. Dicho problema, es uno de los 21 problemas NP-completos de Richard Karp (1972) en su artículo <<Reducibility Among Combinatorial Problems>>. El autor definió el problema como: dada una mochila con una capacidad de 15 kg, que puedo llenar con cajas de distinto peso y valor, ¿qué cajas elijo de modo que maximice mis ganancias y no me exceda de los 15kg de peso permitidos?

Asimismo, no siempre los NP-duros son NP completos (véase *Figura 1*). Un ejemplo de ello es el problema de la parada de la máquina de Turing. Dicho problema consiste en, dada una máquina de Turing $\mathcal{M}$ y una palabra ℓ , el problema intenta determinar si la máquina de Turing terminará en un número finito de pasos cuando esta es ejecutada usando la palabra ℓ como dato de entrada. El matemático Alan Turing demostró en su artículo «On Computable Numbers, with an Application to the Entscheidungsproblem» en 1936, que ninguna máquina de Turing puede resolver dicho problema.

Ante estas definiciones, nos puede surgir la siguiente pregunta ¿entonces, qué diferencia hay entre las clases NP-completo y NP-duro? La diferencia entre ambas clases es que un problema puede ser NP-duro sin pertenecer a la clase NP.

Una de las preguntas abiertas más importantes en la actualidad es descubrir el paradigma de si $P=NP$. Lo cierto es que, a pesar de todos los descubrimientos e investigaciones realizadas, en teoría de complejidad computacional todavía no se conoce si es cierta dicha igualdad. A pesar de que muchos investigadores creen que ambas clases deben ser distintas, la pregunta sigue sin resolverse tanto que, este es conocido como uno de los siete problemas del milenio del Clay Mathematics Institute por el cual se ofrece un millón de dólares por su resolución.

El problema del viajante es de gran relevancia, no solo por su complejidad, sino también por sus numerosas aplicaciones en distintos sectores. Uno de los sectores más destacados es el sector de la logística. En este sector es de destacar un caso particular: los problemas de rutas de vehículos (en inglés, Vehicle Routing Problem con sus siglas VRP). En este caso, se trata de que el vehículo aprovisiona desde un almacén u oficina central a los distintos puntos de demanda, minimizando la distancia total recorrida. Así, como vemos, este problema puede ser aplicado al reparto de cualquier mercancía.

Por otro lado, otro sector de gran relevancia es el sector de la industria, donde podemos destacar la aplicación del problema del viajante al diseño de circuitos o a la producción de circuitos electrónicos. En este caso, el objetivo del TSP es minimizar la cantidad de cable eléctrico que se utiliza para conectar las clavijas que se asignan a las distintas componentes. Veremos esta aplicación con más detalle a continuación.

Otra aplicación sería en el problema de secuenciación de una máquina. Esta tiene que realizar un número de trabajos con un orden. El objetivo del problema sería que esta secuenciación se realizase en el menor tiempo posible, teniendo en cuenta que dicha máquina tendrá ciertas condiciones en lo que concierne a su temperatura, presión, etc. El coste, al que denominaremos c_{ij} representará el tiempo que invierte la máquina en cambiar del estado exigido para realizar el trabajo i (c_i) al estado requerido para llevar a cabo el trabajo j (c_j).

Asimismo, es frecuente que en todos estos tipos de problemas se cree un almacén ficticio donde convertir todas estas variantes de problemas en un TSP estándar.

Otro sector de gran importancia es el sector de la robótica, en la aplicación del problema del viajante al problema de placa de circuitos impresos (en inglés, Printed Circuit Board con las siglas PCB). Una placa de circuitos impresos es un soporte físico en donde se instalan componentes electrónicos y eléctricos, los cuales se interconectan entre ellos según se desea.

El problema PCB lo podemos dividir en dos subproblemas: encontrar el orden óptimo de perforar las placas y obtener los caminos o el orden óptimo para comunicar los chips. En lo que respecta al problema de perforación de las placas, tenemos que considerar las posiciones o puntos a perforar como las ciudades (en el caso del problema del viajante) y las distancias entre ellas como el tiempo que necesita la máquina en trasladarse de un punto o posición a otra. Así, el punto inicial y final será un punto adicional donde permanece la perforadora mientras descansa. En base a esto, podemos llegar a la conclusión de que, si estas máquinas no están bien programadas, el tiempo que tarde en recorrer un punto u otro puede no ser óptimo con la consecuencia de que la producción de placas bajaría en un período de tiempo.

Por otro lado, respecto al problema de conexión de los chips, el objetivo es minimizar la cantidad de cable necesaria para unificar todos los puntos de una placa sin interferencias. Teniendo en cuenta que los chips son de pequeño tamaño, no existe la posibilidad de poner más de dos cables en un pin². Así, la equivalencia de este problema al problema del viajante radica en que los pines serían las ciudades y la cantidad de cable necesaria para unir estos sería la distancia entre las ciudades.

El origen del problema del viajante no está del todo claro. Este fue definido en el siglo XIX por los matemáticos W.R. Hamilton y T. Kirkman. Sin embargo, la forma general del problema del viajante, en base a las investigaciones realizadas, se cree que fue

² Un pin es una terminal o patilla de cada uno de los contactos metálicos de un conector o de un componente fabricado de un material conductor de la electricidad.

estudiado por primera vez en Viena y Harvard por ciertos matemáticos durante la década de 1930, destacando los estudios realizados por Karl Menger.

Karl Menger fue uno de los matemáticos más destacados en este campo. Este definió el problema del viajante, considerando el algoritmo de fuerza bruta como una de las herramientas para solucionar el TSP. Karl Menger observó también, la no optimalidad de la heurística de los vecinos más cercanos de la que hablaremos más adelante.

Durante los años 1950 y 1960, el problema del viajante fue cogiendo popularidad entre los científicos provenientes de Europa y Estados Unidos. Una contribución que destacar es la realizada por los autores Dantzig, Fulkerson y Johnson (1954), quienes expresaron el TSP como un problema de Programación Lineal en enteros y desarrollaron para su resolución el método de planos de corte³.

Otro hallazgo para destacar fue el realizado por el informático Richard Manning Karp. Este autor demostró, en 1972, que el problema del camino hamiltoniano era un problema NP-completo. Esta demostración implica que el TSP sea un problema NP-duro debido a la dificultad computacional para encontrar soluciones óptimas. Asimismo, uno de los últimos hallazgos fue en los años noventa, cuando Applegate D., Bixby R.E., Chvátal V. y Cook W. J. desarrollaron el programa Concorde. El programa Concorde está formado por múltiples trabajos, siendo, algunos de los trabajos actuales, apoyados por el Consejo de Investigación de Ciencias Naturales e Ingeniería de Canadá (con sus siglas en inglés NSERC) y por el Departamento de Combinatoria y Optimización de la Universidad de Waterloo.

El objetivo de este capítulo es revisar el problema del viajante, siendo este un problema destacado de la optimización combinatoria, minimizando la distancia o el coste de la ruta realizada por el agente, pasando por cada ciudad una vez y saliendo y regresando del mismo lugar de partida. Para ello, utilizaremos algunas herramientas como el programa creado por Applegate D., Bixby R.E., Chvátal V. y Cook W. J : El programa Concorde. Asimismo, a pesar de la existencia de diferentes tipos del problema del viajante de

³ Método utilizado para encontrar soluciones enteras de un problema lineal.

comercio, en este trabajo nos centraremos en una: el problema del viajante simétrico, del que hablaremos a continuación.

1.2. Tipos de TSP

A lo largo de la historia se han desarrollado diferentes tipos de problemas del viajante. Entre ellos, los principales son: el problema del viajante simétrico (con sus siglas en inglés, sTSP) y el problema del viajante asimétrico (aTSP).

En lo que respecta al problema del viajante simétrico (sTSP), este presenta un conjunto de ciudades o nodos definido por $V = \{1, \dots, n\}$, un conjunto de arcos o aristas $A = \{(i, j): i, j \in V, i \neq j\}$ con un coste o distancia asociada al arco igual a $c_{ij} = c_{ji}$ o $d_{ij} = d_{ji}$, respectivamente. El sTSP se centra en encontrar un coste o distancia mínima con la que visitar cada ciudad una vez minimizando el tour realizado. En este caso, cada ciudad ($i \in V$), estará definida por las coordenadas (x_i, y_i) teniendo una distancia euclídea (distancia ordinaria entre dos puntos de un espacio euclídeo) entre las ciudades i y j igual a d_{ij} , obteniendo así un problema del viajante euclídeo.

Por otro lado, como bien comentamos anteriormente, otro tipo de problema del viajante es el problema del viajante de comercio asimétrico (aTSP). Dicho problema se asemeja bastante al caso del problema del viajante simétrico, pero con un conjunto de arcos o aristas definidas por $E = \{(i, j): i, j \in V, i < j\}$ y con la diferencia de que las distancias o los costes entre las ciudades i y j pueden ser diferentes, es decir, al menos existen dos ciudades tales que $d_{ij} \neq d_{ji}$ ó $c_{ij} \neq c_{ji}$.

Asimismo, para facilitar la lectura y comprensión del lector, en este trabajo cuando hablemos de costes entre las ciudades, estaremos haciendo referencia a la distancia entre ellas.

1.3. Formulación

En esta sección, explicaremos en detalle la formulación de los diferentes tipos de TSP explicados anteriormente. El problema del viajante en general se puede definir como un grafo no dirigido u orientado ($G = (V, A)$) en el caso de que el problema del viajante sea simétrico (sTSP) o en un grafo dirigido ($G = (V, E)$) en el caso de ser asimétrico (aTSP).

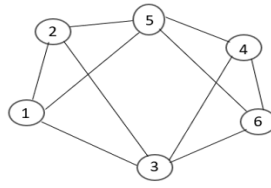
Un *grafo no orientado o dirigido* (Véase *Figura 2.a*) en V se puede definir como un conjunto de arcos o pares no ordenados de elementos distintos de V . Si denotamos por A al conjunto de arcos o aristas que une a dos jugadores o ciudades distintas $i, j \in V$ (por lo que $(i:j) = (j:i)$, ya que son pares no ordenados), entonces un grafo no orientado es cualquier subconjunto del conjunto

$$A = \{(i,j): i,j \in V, i \neq j\}$$

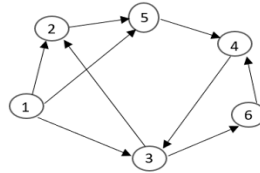
Denotamos por $g(V)$ el conjunto de grafos no orientados en V . Para cada $S \in 2^V$ y cada $A \in g(V)$ $i, j \in S$ con $i \neq j$ están conectados en S a través de A si existe una secuencia de jugadores distintos $\{n^k\}_{k=0}^{k'} \subset S$ tal que $n^0 = i, n^{k'} = j$ y $[n^{k-1}:n^k] \in A$ para todo $k = 1, \dots, k'$. Un componente conexo es un subconjunto maximal de jugadores conectados a través de un grafo no dirigido u orientado. Denotamos por $S|A$ la clase de las componentes conexas inducidas por A en S . Asimismo, al ser un grafo no dirigido, los arcos representan relaciones simétricas y el sentido de estos no está definido.

Por otro lado, un grafo dirigido u orientado (Véase *Figura 2.b*) se puede definir como:

Un *grafo dirigido u orientado* es un grafo en el cual los arcos tienen un sentido definido. Formalmente, un grafo dirigido está definido por un par de conjuntos G , donde, si el problema es asimétrico, este es definido como $G = (V, E)$. Donde $E \in g(V)$ $i, j \in S$ con $i < j$ y están conectados en S a través de E .



a) Grafo no orientado o dirigido



b) Grafo orientado o dirigido

FIGURA 2. TIPOS DE GRAFOS

Como bien comentamos en la sección anterior, el conjunto $V = \{1, \dots, n\}$ es el conjunto de ciudades o vértices en el caso del sTSP y aTSP (representados en la *Figura 2* por vértices que van desde 1 hasta 6). Asimismo, tal y como explicamos en el epígrafe anterior, nos podemos encontrar con dos tipos de problema del viajante: el simétrico y asimétrico. El problema del viajante simétrico presenta un conjunto de arcos definidos por $A = \{(i, j): i, j \in V \text{ con } i \neq j\}$ y, por otro lado, para el problema asimétrico tenemos un conjunto de arcos definidos por $E = \{(i, j): i, j \in V \text{ con } i < j\}$ (representando, dichos arcos, los caminos o la unión entre los vértices o ciudades, y siendo estos dirigidos (E) o no (A)) (véase *Figura 2.a* y *2.b*). Asimismo, podemos definir una ruta o camino como una secuencia de arcos en los que todos los vértices son distintos. Así, formalmente, en el caso de un grafo no dirigido u orientado $G = (V, A)$, un camino es una sucesión de nodos $(v_0, v_1, v_2, \dots, v_i)$ y arcos $A_1, A_2, \dots, A_i$ tal que $A_i = (i_i, i_{i+1}) \in A$ de tal forma que el nodo origen es igual al nodo final del camino, es decir, $i_0 = i_i$.

Ambos problemas presentan similitudes, tanto que comparten la misma matriz de costes $c = (c_{ij})$. Dicha matriz satisface la desigualdad triangular siempre que $c_{ij} \leq c_{ik} +$

c_{kj} para todo i, j, k . La desigualdad de Minkowski o desigualdad triangular proviene del teorema de desigualdad del triángulo de la geometría euclidiana que permite establecer que la suma de la longitud de dos lados cualesquiera del triángulo es mayor o igual que la del tercer lado. La matriz de costes cumple esta desigualdad siempre y cuando el coste de ir desde la *ciudad i* a la *ciudad j* directamente, sea menor o igual que la suma de los costes de ir de la *ciudad i* a la *ciudad k* y de la *ciudad k* a la *ciudad j*.

Así, se considera una restricción natural para el TSP que las distancias entre estas ciudades formen un conjunto de regularidades formales y sistemáticas que satisfagan dicha desigualdad, es decir, que la conexión desde la *ciudad i* a la *ciudad j* no sea superior que la ruta o el camino de la *ciudad i* a la *ciudad j* pasando por la *ciudad k*. Ahora bien, veamos si dicha desigualdad se cumple en todos los casos.

Por ejemplo, imaginemos que tenemos un problema con cinco ciudades, a las que denominaremos *ciudad g*, *ciudad h*, *ciudad i*, *ciudad j* y *ciudad k*. Estas ciudades presentan una matriz de costes igual a:

$$\begin{array}{c} g \quad h \quad i \quad j \quad k \\ \begin{array}{c} g \\ h \\ i \\ j \\ k \end{array} \begin{pmatrix} 0 & 15 & - & - & 25 \\ 15 & 0 & 30 & - & - \\ - & 30 & 0 & 45 & 30 \\ - & - & 45 & 0 & 10 \\ 25 & - & 30 & 10 & 0 \end{pmatrix} \end{array}$$

FIGURA 3. MATRIZ DE COSTES SIMÉTRICA

Con un grafo no dirigido u orientado igual al expuesto en la *Figura 4*.

Dónde la primera columna y fila corresponden al coste de la *ciudad g* con el resto de las ciudades, así, por ejemplo, el coste de ir desde la *ciudad g* a la *ciudad h* es igual a 15 unidades ($c_{gh} = c_{hg} = 15$). La segunda columna y fila corresponden al coste de la *ciudad h* con respecto al resto de ciudades, la tercera columna y fila corresponden al coste de la *ciudad i* con respecto al resto y, así sucesivamente, hasta la última fila y columna que corresponden al coste de la *ciudad k*.

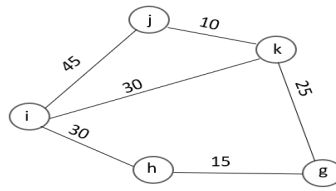


FIGURA 4. GRAFO NO DIRIGIDO CORRESPONDIENTE A LA MATRIZ DE COSTES DE LA FIGURA 3.

Por otro lado, como podemos observar, no todas las ciudades son visitadas, es decir, por ejemplo, la *ciudad i* no tiene un camino o ruta directo a la *ciudad g*. Para poder visitar dicha ciudad debe visitar, primeramente, otra ciudad.

Teniendo en cuenta los costes asociados a los caminos que unen cada ciudad, vemos que esta matriz de costes asociada al problema planteado no cumple con la desigualdad triangular. Un ejemplo de ello es el siguiente caso: $c_{ij} \leq c_{ik} + c_{kj}$ con un resultado igual a $c_{ij} = 45 > 40$, con $c_{ik} = 30$ y $c_{kj} = 10$.

Por el contrario, si el coste de ir de la ciudad *i* a la ciudad *j* fuese igual a 15 ($c_{ij} = 15$) y el resto de los valores se mantuviesen, se cumpliría la desigualdad triangular.

También puede darse el caso de que estemos ante un problema del viajante asimétrico, es decir que $d_{ij} \neq d_{ji}$ ó $c_{ij} \neq c_{ji}$. Esta tipología de TSP presenta una matriz de costes asimétrica (siendo la matriz de costes asimétrica de este ejemplo la representada en la *Figura 5* y un grafo dirigido u orientado igual al mostrado en la *Figura 6*).

Sin embargo, aunque hemos hecho referencia al problema del viajante asimétrico, en este trabajo entraremos más en detalle en el caso de que el problema del viajante presente un grafo simétrico y, por tanto, no dirigido.

	<i>g</i>	<i>h</i>	<i>i</i>	<i>j</i>	<i>k</i>
<i>g</i>	0	30	—	—	30
<i>h</i>	15	0	15	—	—
<i>i</i>	—	30	0	45	30
<i>j</i>	—	—	30	0	10
<i>k</i>	25	—	25	20	0

FIGURA 5. MATRIZ DE COSTES ASIMÉTRICA

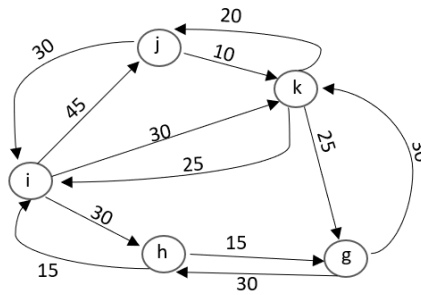


FIGURA 6. GRAFO DIRIGIDO CORRESPONDIENTE A LA MATRIZ DE COSTES DE LA FIGURA 5

Por otro lado, como podemos observar, estamos ante el caso de problemas de planos cartesianos para los cuales los vértices son puntos $V = \{P_i, P_j\}$ en el plano, siendo la ubicación de la ciudad *i* y la ciudad *j* iguales a $P_i = (x_i, y_i)$ y $P_j = (x_j, y_j)$, respectivamente, y el coste $c_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$ que hace referencia a la distancia Euclídea.

1.3.1. El problema del viajante simétrico

El problema del viajante de comercio simétrico se puede plantear como un problema de programación lineal y entera. En este caso, nos basaremos en la formulación propuesta por el autor *Tucker* en el capítulo trece del libro de Papadimitriou y Steiglitz publicado en 1981. Entonces, podemos definir el TSP como:

$$\text{minimizar } \sum_{i,j=0}^n c_{ij}x_{ij} \quad (1.1)$$

sujeto a:

$$\sum_{i \neq j, i=0}^n x_{ij} = 1 \quad j = 0, \dots, n \quad (1.2)$$

$$\sum_{i \neq j, j=0}^n x_{ij} = 1 \quad i = 0, \dots, n \quad (1.3)$$

$$u_i - u_j + nx_{ij} \leq n - 1 \quad 1 \leq i \neq j \leq n \quad (1.4)$$

$$0 \leq x_{ij} \leq 1 \quad x_{ij} \text{ con } i, j = 0, \dots, n, \forall i, j$$

Así, como vemos, en esta formulación los autores presentan una variable binaria denominada x_{ij} . Dicha variable tendrá un valor igual a uno si el arco (i, j) está en el camino y cero en caso contrario, es decir:

$$x_{ij} = \begin{cases} 1 & \text{si el camino va desde la ciudad } i \text{ a la ciudad } j \\ 0 & \text{cualquier otro caso} \end{cases}$$

Asimismo, definen c_{ij} como el coste de ir desde la ciudad i a la ciudad j . Teniendo que ser, dicho coste, mayor o igual a cero ($c_{ij} \geq 0$) ya que este representa la distancia entre las ciudades i y j .

Como podemos observar, ambas variables son las que forman la función objetivo (1.1), donde se minimiza el coste del camino realizado, siendo este igual a cero en el caso de que el arco (i, j) no esté dentro del camino. Esta función objetivo se encuentra sujeta a diferentes restricciones. La primera y segunda restricción, (1.2) y (1.3) respectivamente,

hacen referencia a que de cada nodo tiene que entrar y salir a un único arco. Por último, en la restricción (1.4) disponemos de unas variables reales de libre disposición definidas formalmente como u_i con $i = 1, \dots, n$, a la que los autores describen como condiciones *Miller-Tucker*. Estas condiciones evitan la posibilidad de encontrarnos con una solución del problema formada por subcircuitos y, a su vez, reducen el número de restricciones.

Supongamos que un agente debe visitar cuatro ciudades, empezando y terminando en la misma ciudad y pudiendo ir tanto desde la ciudad A a la ciudad B como a su camino inverso. Teniendo en cuenta que los caminos que unen las ciudades presentan ciertos costes. El objetivo del agente es minimizar el coste que obtendría al realizar dicha ruta y encontrar una ruta o camino óptimo. Así, este problema se puede plantear mediante un grafo simétrico y no dirigido, $G = (V, A)$, igual a:

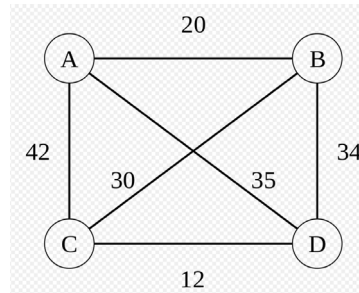


FIGURA 7. GRAFO NO DIRIGIDO. EJEMPLO DEL PROBLEMA DEL VIAJANTE SIMÉTRICO

Como bien comentamos, ir de una ciudad a otra presenta cierto coste, siendo, por ejemplo, el coste de ir desde la *ciudad A* a la *ciudad B* (c_{AB}) igual a veinte unidades e igual al que se obtiene si vamos desde la *ciudad B* a la *ciudad A* como consecuencia de la simetría que presenta el grafo y, por ende, los costes asociados. Así, los costes c_{ij} que presentan las ciudades son iguales a:

$$c_{AB} = c_{BA} = 20$$

$$c_{AC} = c_{CA} = 42$$

$$c_{AD} = c_{DA} = 35$$

$$c_{CB} = c_{BC} = 30$$

$$c_{CD} = c_{DC} = 12$$

$$c_{BD} = c_{DB} = 34$$

Ahora supongamos que fijamos un nodo inicio u origen: la *ciudad D* y que cada ciudad sólo puede ser visitada una vez y, la ruta, debe comenzar y terminar en el mismo nodo o ciudad inicial. En base a esto, obtendríamos las siguientes rutas con sus respectivos costes:

RUTA	COSTE
D-C-B-A-D	97
D-A-B-C-D	97
D-B-A-C-D	108
D-C-A-B-D	108
D-A-C-B-D	141
D-B-C-A-D	141

TABLA 1. POSIBLES RUTAS Y SUS COSTES DE LA FIGURA 7.

En este caso obtendríamos dos caminos de menor coste: D-C-B-A-D y D-A-B-C-D que presentan un coste igual a 97 unidades. Supongamos que el agente decide ir primero por la ciudad C y seguir la primera ruta obtenida. Así, en base a lo descrito anteriormente los valores de las variables x_{DC} , x_{CB} , x_{BA} , x_{AD} , que van desde el arco D hasta el arco A, son iguales a uno debido a que el camino o ruta obtenido pasa por esos tramos yendo desde la *ciudad C a la A*, empezando y terminando por el nodo origen: *la ciudad D*. Por el contrario, las ciudades o nodos que no son visitados en la ruta obtenida tienen un valor igual a cero, $x_{AC} = x_{BD} = 0$.

Finalmente, teniendo en cuenta la formulación detallada anteriormente (1.1) y los resultados expuestos en la Tabla 1, el coste total a minimizar del camino obtenido es igual a:

$$\min \sum_{i,j=0}^n c_{ij}x_{ij} = (1 * 12 + 1 * 30 + 1 * 35 + 0 * 42 + 0 * 34) = 97 \text{ unidades}$$

1.4. Métodos de resolución del TSP

Como bien comentamos en epígrafes anteriores, el TSP es un problema que está considerado como un problema NP-duro. Sin embargo, esto no implica que el TSP no tenga solución en todos los casos. Realmente, el TSP tiene solución ya que se puede evaluar todas las posibles soluciones y escoger la de menor coste, pero el problema radica en que el número de posibles soluciones aumenta cuando aumenta el tamaño del problema o el número de ciudades que va a visitar el agente (n) y esto hace que su resolución sea computacionalmente más costosa. Así, el número total de rutas posibles que cubren todos los lugares es un posible conjunto de soluciones factibles y se pueden calcular mediante el método directo. Este método consiste en evaluar de forma directa las $(n - 1)!$ soluciones o rutas posibles.

Así, teniendo en cuenta el ejemplo realizado en el epígrafe anterior, formado este por cuatro ciudades (A, B, C y D) obtendríamos un total de $\frac{(n-1)!}{2}$ rutas posibles ya que en el TSP simétrico para cada ruta elegida su inversa tiene el mismo coste. Por tanto, con $n = 4$ ciudades, obtenemos $\frac{(n-1)!}{2} = \frac{(4-1)!}{2} = 3$ soluciones o rutas posibles para evaluar.

Sin embargo, si el número de ciudades a visitar por el agente aumentase hasta, por ejemplo, un total de 10 ciudades, deberíamos evaluar 181440 rutas diferentes.

Como vemos, para un tamaño del problema no muy grande, el número de ciudades a evaluar aumenta de forma considerable. Por lo tanto, sería interesante buscar algoritmos que disminuyan el número de pruebas a realizar. Ante esto, los autores Held y Karp (1962) desarrollaron un nuevo algoritmo proveniente de la teoría de grafos y de la programación dinámica. Este algoritmo está diseñado para encontrar un recorrido mínimo donde el agente pase por todos los nodos y regrese al nodo de partida. El algoritmo de Held-Karp presenta una complejidad temporal exponencial igual a $O(n^2 2^n)$. En el campo de la informática, la complejidad temporal hace referencia a la complejidad

computacional que describe la cantidad de tiempo que lleva ejecutar un algoritmo y se expresa usando la notación de la O grande o notación de Landau, siendo n el tamaño de entradas en unidades de bits. La notación de la O grande o de Landau es una notación para la comparación asintótica de funciones, lo que permite establecer la cota inferior, superior y ajustada asintótica, siendo la cota superior asintótica una función que sirve de cota superior de otra función cuando el argumento tiende a un valor igual al infinito. Por tanto, siguiendo con el ejemplo de diez ciudades, con este algoritmo tendríamos una complejidad temporal exponencial igual a $O(10^2 2^{10}) = O(102400)$, mucho mejor que el tiempo de ejecución o rendimiento que se obtiene con el método directo o fuerza bruta $O\left(\frac{(n-1)!}{2}\right)$, siendo con un tamaño igual a diez ciudades igual a $O\left(\frac{(10-1)!}{2}\right) = O(181440)$, del que hablaremos más detalladamente en el siguiente epígrafe.

Por tanto, a la luz del resultado obtenido anteriormente, podríamos afirmar que el algoritmo propuesto por Held y Karp es mejor que el método directo, pero ¿en base a que criterios definiríamos que un algoritmo es bueno o malo? La respuesta a esta pregunta fue resuelta por el matemático Edmonds J. en 1965, donde dicho autor, en el artículo “Paths, Trees and Flowers”, comentó la posibilidad de establecer una teoría matemática de algoritmos combinatorios eficientes. Para Edmonds, un algoritmo es eficiente (entendiendo esto como un algoritmo bueno), cuando el tiempo para alcanzar su máximo es $O(n^c)$. Por tanto, ante esta definición y lo propuesto por el autor Edmonds J., el algoritmo propuesto por Held y Karp no es eficiente.

Sin embargo, a pesar de todos estos hallazgos, hoy en día, sigue sin encontrarse un algoritmo “bueno” o eficiente para resolver el problema del viajante. Es por ello por lo que el problema del viajante es un ejemplo de interés dentro de la teoría de la complejidad y uno de los problemas más estudiados en el campo de las matemáticas combinatorias en las últimas décadas. Esta teoría es una rama de la teoría de la computación que se centra en clasificar los problemas computacionales en función de su dificultad y en la relación entre las clases de complejidad. En este caso, un problema se clasifica como difícil si su solución implica una gran cantidad de recursos computacionales (tiempo, por ejemplo), sin tener en cuenta el algoritmo utilizado para

su resolución. Por tanto, ante esta definición, podemos clasificar los problemas en función de si existe o no un buen algoritmo para su resolución; así, los algoritmos “buenos” o “más eficientes” son clasificados como P, dicho termino procede de tiempo polinómico. Nos referiremos como algoritmo “más eficiente” al algoritmo que necesite menos tiempo para resolver el problema. Por otro lado, los problemas para los que no se puede encontrar una solución óptima en un tiempo polinómico se clasifican como NP, siendo este el acrónimo de tiempo polinómico no determinista.

Así, teniendo en cuenta la dificultad para la resolución de algunos problemas como el problema del viajante, muchos autores han desarrollado diferentes técnicas con diferentes algoritmos para la resolución de estos problemas, comenzando desde uno de los primeros algoritmos utilizados para resolver el TSP, el algoritmo del vecino más próximo hasta los últimos algoritmos desarrollados. El *algoritmo del vecino más próximo* es utilizado para obtener una solución rápida a través de la obtención de un camino corto, pero que generalmente, de todas sus posibilidades, no suele ser el que presenta el menor coste posible. Para ello, el algoritmo elige la ciudad o vértice más próximo a la ciudad inicial. Una vez elegida esta, repite el mismo procedimiento (buscar la ciudad más próxima al nodo escogido) hasta que no queden ciudades a visitar.

Estos algoritmos se pueden clasificar en algoritmos precisos y algoritmos aproximados o heurísticos. En este trabajo, entraremos en detalle en algunos algoritmos precisos o exactos actuales y en algunos algoritmos heurísticos como Chained Lin- Kernighan.

Asimismo, con el paso de los años se han desarrollado diferentes herramientas de computación para resolver algunos problemas de optimización como puede ser el caso del problema del viajante de comercio. Una de ellas es el programa Concorde, nombrado anteriormente y que será explicado en detalle en los siguientes epígrafes, el cual utiliza uno de los algoritmos precisos y heurísticos explicados a continuación.

1.4.1. Algoritmos precisos o exactos

A la hora de resolver un problema mediante un algoritmo, la solución más rápida y directa puede ser intentar todas las permutaciones posibles, siendo estas unas

disposiciones de sus miembros en una secuencia u orden lineal y, en el caso de que el conjunto esté ordenado, estas serían una variación del orden o posición de los elementos de un conjunto ordenado y, de todas esas permutaciones, ver cuál es la menor mediante el uso de la fuerza bruta. Así, el uso de la fuerza bruta consiste en enumerar sistemáticamente todos los posibles candidatos para la solución de un problema, con la finalidad de chequear si dicho candidato satisface la solución al mismo. El tiempo de ejecución de un algoritmo preciso o exacto es igual a $O(n!)$. Existen varios algoritmos exactos, entre los que destacan:

Programación Lineal

La programación lineal es un método de resolución del TSP. Este método nació alrededor del año 1826 y con el paso de los años se realizaron varias modificaciones. Esta consiste en maximizar o minimizar una función lineal que denominamos función objetivo estando, las variables de dicha función, sujetas a unas restricciones expresadas mediante un sistema de ecuaciones o inecuaciones lineales. El método que se usa para resolver este tipo de problemas es el método simplex. El método simplex fue desarrollado por el matemático y físico, y el considerado como “el padre de la programación lineal”, G. Dantzig en 1947 y consiste, básicamente, en buscar el máximo o mínimo de una función lineal sobre un conjunto de variables que satisfaga un conjunto de inecuaciones lineales. Asimismo, este método se aplica principalmente al método de planos de corte en la programación entera (siendo esta un problema de optimización matemática en el que algunas o todas las variables, así como la función objetivo y las restricciones deben ser enteras y lineales).

El *método de planos de corte* fue propuesto por R. Gomory en los años 50 y se utiliza para resolver problemas de programación entera y programación entera mixta. Sin embargo, muchos expertos, incluido el propio autor, consideran que este método es poco práctico debido a la inestabilidad numérica, así como por su ineficacia ya que para avanzar hacia la solución era necesario realizar varios cortes. La situación cambió cuando, a mediados de los 90, el autor G. Cornuéjols y sus compañeros de trabajo demostraron que el método de planos de corte era muy efectivo al combinarlo con el

método de ramificar y cortar (en inglés branch and cut), mediante el cual se superaban las inestabilidades numéricas encontradas al utilizar únicamente el método de planos de corte. Actualmente, muchos investigadores utilizan el método propuesto por R. Gomory, debido a que estos cortes se generan de forma eficiente mediante un cuadro simplex. En lo que respecta a su aplicación para la resolución del problema del viajante, los autores Dantzig, Fulkerson y Johnson en 1954 propusieron el empleo de esta técnica. Estos propusieron una formulación del problema del viajante. Los autores se basan en la minimización de la función objetivo $c^t x$, siendo $c^t = (c_{ij})$. Dicha función objetivo se encuentra sujeta a las restricciones $x \in S$, siendo S un conjunto finito de vectores m -dimensionales, que, en el caso del problema del viajante, corresponde con el número total de caminos que unen las ciudades. Así, estos autores definen cada ruta como un vector con una dimensión igual a $\frac{n(n-1)}{2}$ en la que el componente x_{ij} toma el valor uno y cero, si el arco al que corresponde está en el camino o no, respectivamente. Sin embargo, la idea principal del método de planos de corte es resolver un problema sencillo, y poco a poco, ir añadiendo ciertas restricciones hasta obtener una solución que pertenezca al conjunto de vectores m -dimensionales, es decir, a S . Así, en el caso del problema del viajante simétrico (con sus siglas en inglés, sTSP) que ya hemos definido en epígrafes anteriores y, basándonos en la formulación propuesta por los autores nombrados anteriormente, el problema a resolver es el siguiente:

$$\text{Min} \sum_{i,j=0}^n c_{ij} x_{ij} \quad (1.21)$$

s. a:

$$0 \leq x_{ij} \leq 1, \text{ para todo arco } ij \quad (1.22)$$

$$\sum_{i,j \in V}^n (x_{ij} : V \text{ es un extremo de } ij) = 2 \text{ para todo vértice } V \quad (1.23)$$

Como podemos observar, las restricciones de ruptura de subcircuito no se han incluido en la definición del problema y también se permite que las variables de decisión tomen valores no enteros. La omisión de la restricción de ruptura de subcircuito se debe a que

al añadir esta restricción el problema estaría compuesto por demasiadas restricciones e implicaría una complejidad computacional muy alta.

Para solucionarlo, los autores propusieron ir añadiendo restricciones al problema a medida que hiciesen falta, es decir, estos propusieron resolver primero el problema expuesto anteriormente y, una vez resuelto, buscar subcircuitos y plantear un nuevo problema compuesto por las restricciones anteriores más la de rotura de subcircuito que está asociada al conjunto de vértices (V) donde se encuentra el subcircuito. Ante ello, existe la posibilidad de que nos encontremos con soluciones que no sean circuitos, pero, lo que sí es seguro es que cualquier solución factible del problema del viajante es también solución factible del problema del viajante simétrico.

Ramificación y acotación

El método de diseño de algoritmos Ramificación y acotación (conocido en inglés como Branch and Bound) se interpreta como un árbol de soluciones, donde cada rama nos lleva a una posible solución posterior a la actual. La característica principal de este método es que el algoritmo empleado se encarga de detectar y, en consecuencia, eliminar, qué ramificación de las soluciones obtenidas no es óptima y así no invertir más tiempo en los casos que están lejos de la solución óptima. Así, los pasos a seguir para aplicar el algoritmo son:

1. Resolvemos el problema original relajado (es decir, sin considerar la restricción que establece que las variables tienen que ser enteras). Entonces, si el problema no es factible, paramos el proceso y, en el caso de que la solución sea entera entonces estamos ante la solución óptima; si no es el caso, consideramos una cota superior que denominaremos por z^* (referente al óptimo de z) como $+\infty$ y la cota inferior como una función objetivo del problema relajado. Seguidamente, una vez realizado este primer paso, continuamos con el segundo.
2. En este segundo paso procedemos a ramificar. Esto se realiza escogiendo una variable que tiene que ser entera pero que no lo es y, mediante ramificación, generamos dos problemas. Estos problemas se colocan en una lista de

problemas, de forma ordenada, que serán resueltos en serie o en paralelo. A continuación, realizamos el paso tres.

3. Una vez ramificado el problema y encontrado una solución, tenemos varios casos:
 - Si la solución de un subproblema es entera, se poda (es decir, se elimina). Así, si el valor de la función objetivo es menor que la cota superior (z^*) actual, entonces lo actualizamos con el nuevo valor, disponiendo así, de un candidato a óptimo y actualizamos todas las cotas superiores. En el caso contrario, se descarta esta solución. Una vez actualizado, realizaríamos lo descrito en el paso cuatro.
 - Si, por el contrario, la solución del subproblema no es entera y el valor de la función objetivo es superior al de la cota inferior de su nodo, entonces actualizamos dicha cota y procedemos a realizar el paso dos de nuevo.
 - Si la solución no es entera y el valor de la función objetivo es superior al de la cota superior de su nodo, entonces podamos dicha rama. Una vez podada, realizaríamos lo descrito en el paso cuatro.
 - Por último, si la solución no es factible, entonces podamos dicha rama y procedemos a realizar lo descrito a continuación.
4. Si todos los problemas de la lista han sido resueltos, el procedimiento de ramificación y acotación concluye con un candidato que será el óptimo. En caso contrario, se procede a resolver los problemas pendientes de procesar y volveríamos a realizar el paso tres.

1.4.2. Algoritmos aproximados o heurísticos

Un algoritmo heurístico es un algoritmo que produce soluciones sin ninguna garantía de optimalidad, pero, como resultado, suelen tener un tiempo de ejecución mucho menor que los otros algoritmos. Por tanto, dichos algoritmos se emplean cuando las soluciones que se quieren obtener son costosas computacionalmente. Es por ello por lo que estos

son, a menudo, utilizados para resolver problemas NP-duros, como es el caso del TSP. Hoy en día existen tres categorías de heurísticas: constructivas, de mejora iterativa y de mejoras aleatorias. A continuación, vamos a explicar más detalladamente algunos de estos algoritmos.

Heurística de intercambio par a par

Este algoritmo, que conforman los algoritmos heurísticos de mejora iterativa, también definida como 2-opt, conlleva en cada iteración la eliminación de dos arcos produciendo la creación de fragmentos que serán reconectados mediante unos dos arcos nuevos, produciendo así un camino nuevo más corto.

Chained Lin-Kernighan

Fue desarrollado por Martin, Otto y Felten (1991,1992) y es considerado como uno de los mejores heurísticos para resolver el problema del viajante simétrico.

Como bien sabemos, el objetivo del TSP en los enfoques heurísticos es encontrar un tour de bajo coste en un tiempo computacional razonable; para ello, uno de los mejores métodos para conseguir este objetivo es el algoritmo de Lin y Kernighan cuyo proceso es un “método de mejora del tour” en el que escoge un recorrido e intenta modificarlo para obtener un tour de menor coste. La idea en la que se basa este algoritmo es que, mientras haya tiempo para realizar más cálculos, al generar un nuevo recorrido inicial y aplicar el algoritmo, siempre hay una posibilidad de encontrar un tour de menor coste que el escogido anteriormente.

Sin embargo, la puesta en práctica de este algoritmo terminó tras la publicación del algoritmo Chained Lin-Kernighan por los autores Martin, Otto y Felten (1991,1992), quienes argumentaron que la aplicación de este algoritmo es ineficiente a la hora de probar las soluciones localmente óptimas obtenidas mediante el algoritmo de Lin-Kernighan. Ante ello, los autores propusieron una alternativa a esta práctica. La alternativa es “perturbar” el tour de Lin-Kernighan y aplicar el algoritmo de Lin-Kernighan. Si con ello obtenemos un tour mejor, descartamos el tour anterior y

trabajamos con el nuevo; en caso contrario, continuamos con el tour más antiguo y lo volvemos a “perturbar” hasta encontrar uno mejor.

Por último, según lo expuesto por los autores Applegate, Cook y Rohe en 2003, el algoritmo Chained Lin-Kernighan tiene un componente muy importante: el tour o camino inicial. Realmente, para ejemplos pequeños el tour inicial no es un problema, debido a que el algoritmo es lo suficientemente potente como para superar casi cualquier camino que se le dé, pero, en el caso de ejemplos grandes, la elección del camino inicial puede tener un gran impacto en el desempeño del algoritmo.

Ambos métodos son utilizados por uno de los principales solucionadores y, según algunos autores, por uno de los programas más rápidos y eficaces para resolver un TSP, incluso si este está compuesto por un gran número de agentes: El Concorde TSP Solver.

1.5. Concorde TSP Solver

El Concorde TSP Solver es un programa que fue creado Applegate D., Bixby R.E., Chvátal V. y Cook W. J. para resolver el problema del viajante de comercio (TSP) y algunos problemas de optimización de red relacionados con el TSP. El código de este programa está escrito en el lenguaje de programación ANSI C⁴ y es de uso gratuito para la investigación académica. Según los autores Mulder y Wunsch (2003), Concorde es, de entre todos los programas existentes en la actualidad, el más rápido para resolver un TSP, incluso para ejemplos con un gran número de agentes.

Concorde, para su funcionamiento, utiliza el método de planos de corte. Mediante este método resuelve iterativamente las relajaciones de la programación lineal del problema del viajante de comercio. La interfaz del programa muestra cómo va progresando este al final de cada iteración en los planos de corte mediante el coloreado de los bordes según sus valores LP actuales. Una vez el programa obtiene el recorrido óptimo para el

⁴ Estándar publicado por el Instituto Nacional Estadounidense de Estándares (ANSI) para el lenguaje de programación C.

TSP, éste se muestra mediante unos bordes rojos (Observar la imagen de la derecha mostrada en la *Figura 8*).

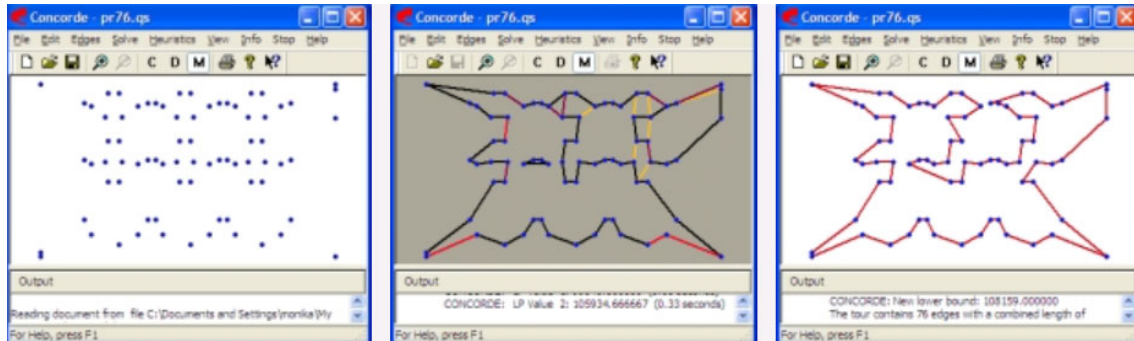


FIGURA 8 PROGRAMA CONCORDE. FUENTE: APPLGATE D., BIXBY R.E., CHVÁTAL V. Y COOK W. J.

Retomando el ejemplo realizado en el *epígrafe 1.3*, supongamos que un comerciante tiene que visitar cuatro establecimientos repartidos por su municipio y que, los costes a los que incurre este, reflejados en la *Figura 7*, se basan en las distancias que tiene que recorrer entre un establecimiento y otro con el objetivo de minimizar dichas distancias. Este problema se puede resolver mediante el servidor NEOS para Concorde. Dicho servidor es una aplicación cliente-servidor basada en Internet que permite el acceso gratuito a una biblioteca de solucionadores (solvers) de optimización. Dentro de este servidor, existen diferentes formatos de datos con los que se puede trabajar, en nuestro caso, resolveremos dicho problema mediante el formato de la librería TSPLIB. Dicho formato está basado en un archivo de texto (.txt). El formato consta de una serie de definiciones de nodos, seguida de una lista de arcos o aristas, que especifican pares de nodos y una etiqueta de borde opcional. Así, el resultado obtenido es el mostrado en la *Figura 9*.

Como vemos en la *Figura 9*, el comerciante debe realizar la ruta A-B-C-D-A (siendo estos nodos los correspondientes a 0-1-2-3-0 en la salida obtenida con el servidor NEOS) con un coste o distancia total de 97 unidades. En este caso, no hemos fijado la ciudad D como nodo inicial, y, por tanto, el servidor ha establecido como este la ciudad A. Así, si comparamos el resultado obtenido con NEOS y el obtenido en el *epígrafe 1.3*,

este resultado o camino es el inverso del obtenido anteriormente: D-C-B-A-D pero con el mismo coste o distancia.

```

NEOS Server Version 6.0
Job#       : 12461511
Password   : cwHyNxTI
User       :
Solver     : co:concorde:TSP
Start      : 2022-11-25 13:08:30
End        : 2022-11-25 13:08:38
Host       : athene.la.asu.edu

Disclaimer:

This information is provided without any express or
implied warranty. In particular, there is no warranty
of any kind concerning the fitness of this
information for any particular purpose.

Announcements:
*****
/home/neos5/bin/concorde.cplex -s 99 -f sample.tsp
Host: athene Current process id: 17568
Using random seed 99
Problem Name: Ejemplo matriz de costes simétrica ejemplo del comerciante
Formato TSPLib para matriz de costes simétrica ejemplo del comerciante
Problem Type: TSP
Number of Nodes: 4
Explicit Lengths (CC_MATRIXNORM)
Optimal Solution: 97.00
Total Running Time: 0.00 (seconds)

***

*** You chose the Concorde(CPLEX) solver ***

*** Cities are numbered 0..n-1 and each line shows a leg from one city to the next
followed by the distance rounded to integers***

4 4
0 1 20
1 2 30
2 3 12
3 0 35

```

FIGURA 9. SALIDA DE CÓDIGO DEL SERVIDOR NEOS

Asimismo, como podemos observar en el ejemplo mostrado anteriormente, el programa ha necesitado menos de cinco segundos, exactamente 0, para resolver este problema con un tamaño igual a 4 ciudades. Sin embargo, el tiempo que necesita el programa Concorde para resolver un ejemplo dado, puede variar considerablemente dependiendo del número de ciudades y la complejidad de éste. Así, Concorde se ha llegado a utilizar para obtener las soluciones óptimas para un conjunto completo de 110 ejemplos. Dichos ejemplos se encuentran dentro de la librería TSPLIB. La librería TSPLIB fue publicada por Gerhard Reinelt en 1991. Esta está compuesta por instancias de pruebas de distinta dificultad, fuentes y tipos.

Actualmente, el mayor problema de TSP resuelto por el programa Concorde se constituye de un recorrido óptimo de 85.900 ciudades. Dicho problema, basado en el diseño de un microchip, fue resuelto en 2006 por varios autores, entre ellos Cook, y es, hoy en día, el ejemplo con el mayor número de ciudades resuelto.

Así, para otros ejemplos con millones de ciudades, los autores Rego, Gamboa, Glover y Osterman en 2011, consideraron que la solución de un problema con tantas ciudades puede ser encontrada pero que ésta solo tiene garantizada entre un dos y un tres por ciento del recorrido óptimo.

A lo largo de la historia, son numerosos los problemas que han sido resueltos mediante este solver. En la *Tabla 2 e Figura 10*, en base a lo expuesto por Infantes Durán (2018), podemos ver una evolución a lo largo de la historia del TSP. En ellas podemos apreciar los diferentes autores que han resuelto cada TSP con sus respectivos números de ciudades. Asimismo, exponemos en la *Ilustración 3* la evolución del problema del viajante de comercio en la historia, donde se aprecia en el eje de abscisas los años a los que corresponde cada trabajo y en el eje de ordenadas el número de ciudades resueltas en cada TSP.

Año	Autores	número ciudades
1954	G.Dantzig, R.Fulkerson, S.Johnson	49
1971	M.Held,R.M.Karp	57
1971	M.Held,R.M.Karp	64
1975	P.M.Camerini, L. Fratta, F.Maffioli	67
1975	P.Miliotis	80
1977	M. Grötschel	120
1980	H. Crowder and M.W. Padberg	318
1987	M. Padberg and G. Rinaldi	532
1987	M. Grötschel and O. Holland	666

1987	M. Padberg and G. Rinaldi	1002
1987	M. Padberg and G. Rinaldi	2392
1994	Applegate, Bixby, Chvátal y Cook	7397
1998	Applegate, Bixby, Chvátal y Cook	13509
2001	Applegate, Bixby, Chvátal y Cook	15112
2004	Applegate, Bixby, Chvátal, Cook y Helsgaun	24978
2005	Cook, Espinoza y Goycoolea	33810
2006	Applegate, Bixby, Chvátal y Cook	85900

TABLA 2. NÚMERO DE CIUDADES RESUELTAS EN CADA TSP. FUENTE: INFANTES DURÁN (2018)

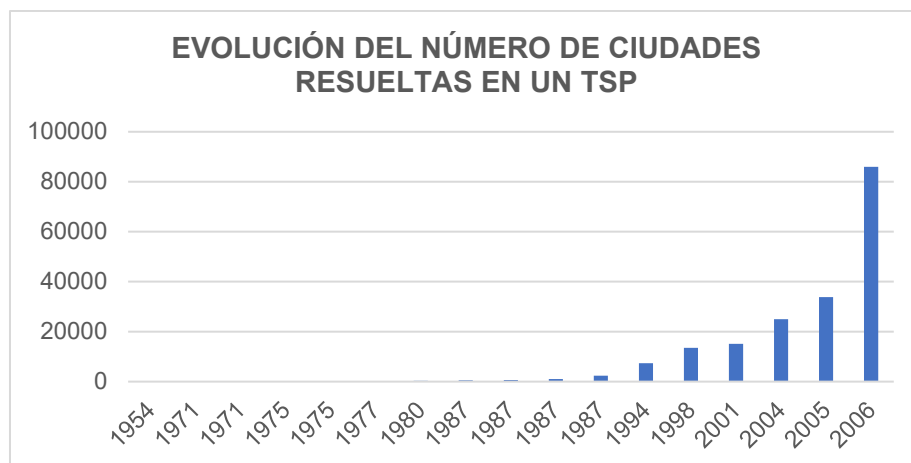


FIGURA 10. EVOLUCIÓN DEL NÚMERO DE CIUDADES RESUELTAS EN UN TSP A LO LARGO DE LA HISTORIA. FUENTE: ELABORACIÓN PROPIA EN BASE A LOS DATOS EXPUESTOS POR INFANTES DURÁN (2018)

En lo que respecta a los tipos de algoritmos que utiliza este programa, Concorde contiene varios solucionadores. Para el caso del TSP, uno de los solucionadores más avanzados y rápidos son los algoritmos basados en la ramificación y acotación definidos por Applegate, Bixby, Chvatal y Cook en 2001 y la implementación Chained Lin-Kernighan realizada por los autores Applegate, Cook y Rohe (2003) explicados, ambos, en el epígrafe anterior.

Capítulo 2

Reparto de costes con juegos cooperativos

2.1. Introducción

“La teoría de juegos es una rama de las matemáticas y de la economía que estudia la elección de la conducta óptima de un individuo cuando los costes y los beneficios de cada opción no están fijados de antemano, sino que dependen de las elecciones de otros individuos” (Sevilla Arias, 2016)

En las últimas décadas se ha profundizado en la teoría de juegos hasta encontrar así juegos que pueden ser divididos en diferentes categorías siendo, una de ellas, los juegos cooperativos.

Definimos como *juego cooperativo* a aquel juego en el cual dos o más jugadores forman una coalición para conseguir el mismo objetivo y obtener así un beneficio común. En el análisis de los juegos cooperativos, se intenta verificar que las coaliciones obtenidas sean estables, es decir que ningún miembro esté inconforme y prefiera actuar individualmente más que cooperando con el resto de los agentes. Así, definimos una coalición como un grupo de jugadores que tienen la posibilidad de tomar decisiones conjuntas para obtener beneficios comunes en un juego en concreto. Formalmente, dado un conjunto de N jugadores, una coalición es cualquier $S \subseteq N$. Llamaremos así, gran coalición a la coalición que viene dada por $S = N$ y coalición nula al conjunto vacío $S = \emptyset$.

La teoría de juegos cooperativos supone que los jugadores adoptan acuerdos entre sí, considerando coaliciones y asignaciones, basándose en nociones de justicia y equidad. Por otro lado, tenemos también los *juegos no cooperativos* donde cada jugador toma las decisiones de manera independiente buscando su beneficio personal. Así, en la teoría de juegos no cooperativos se consideran diferentes estrategias y pagos donde cada jugador adopta la estrategia que maximiza sus pagos individuales.

En un juego cooperativo no es necesario analizar las estrategias de los jugadores, puesto que éstos actuarán de la forma que consigan mayor beneficio, en un juego no cooperativo, en cambio, se tienen en cuenta las distintas estrategias que pueden realizar cada uno de los agentes existiendo, por tanto, una función de pagos asociada a cada jugador que dependerá de las estrategias que decidan emplear.

Dentro de los juegos cooperativos existen dos clases principales: los juegos con utilidad transferibles (juegos TU) y los juegos con utilidad no necesariamente transferible (juegos NTU). En el caso de los *juegos cooperativos TU*, estos corresponden a la situación donde los agentes pueden transferirse de cualquier forma el recurso (ya sean costes o beneficios) entre ellos. Por el contrario, en los *juegos cooperativos NTU*, existe alguna restricción sobre la manera en la que se reparte la utilidad. En nuestro trabajo, nos centraremos en los primeros juegos nombrados, los *juegos cooperativos TU*.

Así, un *juego cooperativo TU* es representado por un par (N, v) tal que el conjunto finito de jugadores o agentes viene dado por $N = \{1, 2, \dots, n\}$ y v es la función característica ($v: 2^N \rightarrow \mathbb{R}$) o de costes (dependiendo de si en el juego TU se trabaja con costes o beneficios) asociada a cada coalición $S \subseteq N$ siendo $v(S)$ el valor de la coalición S y $v(\emptyset) = 0$.

En lo que respecta a la historia de los juegos cooperativos, el precursor de la teoría de juegos fue el economista y matemático francés Cournot, en el siglo XIX, mediante el Equilibrio de Cournot. Otros matemáticos como Zermelo o Borel realizaron otras contribuciones a posteriori anticipando las bases de la teoría de juegos. Sin embargo, esta, realmente, fue fundada por Von Neumann en 1928 mediante la creación del teorema minimax, dando así una noción de lo que hoy en día conocemos como un juego. Concretamente, el punto de partida para la Teoría de Juegos fue la publicación del tratado *Theory of Games and Economic Behavior* (1944) por el propio Von Neumann junto con el economista Morgenstern. Años más tarde, el matemático estadounidense John Nash, desarrolló el Equilibrio de Nash, estableciendo así algunos conceptos importantes para una gama más amplia de juegos. Seguidamente, en los años setenta, los economistas e investigadores Harsanyi y Selten desarrollaron ciertos conceptos que permitieron la aplicación de la Teoría de Juegos al campo de la economía y otras disciplinas.

Muchos de estos autores recibieron premios Nobel por sus aportaciones a esta disciplina; desde el año 1994 donde los autores Nash, Harsanyi y Selten recibieron un premio nobel por sus análisis pioneros del equilibrio en la teoría de los juegos no cooperativos, hasta uno de los premios más recientes galardonado en el año 2012 a los

economistas Roth y Shapley por su trabajo en la teoría de las asignaciones estables y el diseño de mercado.

2.2. Definiciones previas

A continuación, vamos a establecer ciertas definiciones previas sobre algunos conceptos de juegos cooperativos que utilizaremos en epígrafes posteriores y aplicaremos a diferentes ejemplos. Dichas definiciones, han sido obtenidas de la monografía de Sánchez y Vidal (2014).

Definición 2.1. Costes marginales. El coste marginal de un individuo i o, en este caso ciudad i , representa el pago adicional que habría que hacer a este una vez sabemos que las ciudades o individuos restantes van a cooperar conjuntamente.

En lo que respecta al cálculo de los costes marginales de cada individuo o ciudad, estos se calculan eliminando la ciudad i de la ruta o circuito y manteniendo el mismo orden del resto de ciudades, obteniendo así el coste marginal de la ciudad eliminada de la ruta.

Para entender mejor esta definición, consideremos un viajante de comercio que quiere recorrer tres ciudades mediante la ruta de menor coste, partiendo de su ciudad de origen o nodo cero, visitando una vez cada ciudad y teniendo en cuenta los costes asociados a cada una de ellas (ver *Figura 11*). Para ello, el viajante tiene en cuenta las diferentes posibilidades, es decir, todas las rutas posibles, y los costes totales asociados a estas (expuestos en la *Tabla 3*).

Como podemos observar, en base a los datos expuestos en la *Tabla 3*, las rutas de menor coste son las que están enumeradas como 1ª, 2ª, 3ª y 4ª con un coste igual a nueve unidades.

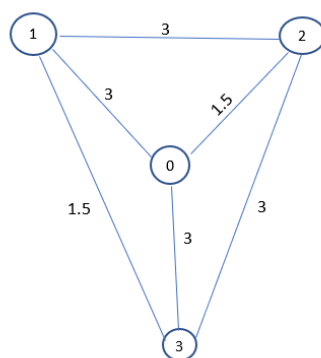


FIGURA 11. GRAFO NO DIRIGIDO

Número	Ruta	Coste
1 ^a	0-2-1-3-0	9
2 ^a	0-2-3-1-0	9
3 ^a	0-1-3-2-0	9
4 ^a	0-3-1-2-0	9
5 ^a	0-1-2-3-0	12
6 ^a	0-3-2-1-0	12

TABLA 3. RUTAS Y COSTES OBTENIDOS DE LA FIGURA 11.

En base a esto, podemos calcular los costes marginales de cada ciudad. Así, si quisiéramos calcular el coste marginal de la ciudad uno y, por ejemplo, de la 1^a y 2^a ruta, procederíamos eliminando la ciudad uno de la ruta y manteniendo el mismo orden del resto de ciudades (véase la imagen de la derecha de la *Figura 12*). Como vemos, el coste marginal obtenido es igual a uno con cinco ($m_1=1.5$), como resultado de la resta del coste total (siendo este igual a nueve con la ruta 0-2-1-3-0 o también mediante la ruta 0-2-3-1-0) y del coste de la ruta eliminando la ciudad *i* (en este caso correspondería a la *ciudad uno*) con un coste igual a siete con cinco unidades, y manteniendo el resto de las ciudades en el mismo orden.

Así, para para obtener los costes marginales de las ciudades dos y tres, realizaríamos los cálculos de forma análoga al caso de la ciudad uno. En base a esto, obtenemos unos

costes marginales de la ciudad dos y tres iguales a $m_2 = 1.5$ y $m_3 = 1.5$, respectivamente.

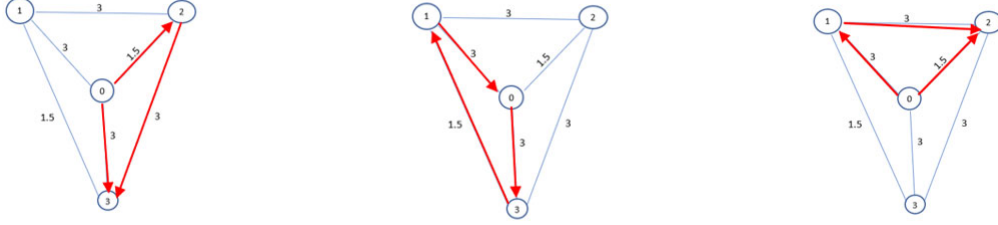


FIGURA 12. COSTES MARGINALES DE LAS CIUDADES.

Definición 2.2. Juego de costes. Un juego de costes (N, c) , se puede definir como un juego donde N es el conjunto de agentes entre los cuales deben ser divididos los costes definidos en la función de costes denominada como $c: 2^N \rightarrow \mathbb{R}$. Para cada coalición $S \subset N$, $c(S)$ representa el mínimo coste en que incurren los miembros de la coalición S . Así, en un grafo completo $G = (V, A)$ donde V corresponde al conjunto de nodos y A hace referencia al conjunto de arcos o aristas que conectan los nodos, siendo el nodo cero el nodo de partida y N el conjunto de agentes ($N = \{1, 2, \dots, n\}$), cada arco (i, j) tiene asociado un coste, siendo dicho coste mayor o igual a cero, es decir, estos arcos no pueden presentar un peso negativo ($c_{ij} \geq 0$). Teniendo en cuenta esto, la función de coste de una coalición de jugadores en un TSP, denominada $c(S)$, representa el coste mínimo que dichos jugadores obtendrán al visitar todos los nodos en S y regresar al nodo de partida o nodo cero. Este coste debe ser repartido entre todos los jugadores donde cada jugador i tiene un coste asociado c_i . Sin embargo, en este trabajo, cuando hablemos del juego de costes, utilizaremos, para referirnos a este concepto, la nomenclatura (N, v) .

Por otro lado, según lo expuesto por los autores Sánchez y Vidal (2014), a la hora de buscar resultados posibles, debe hacerse un reparto del pago total de $c(N)$ entre los

jugadores. El pago a cada jugador puede representarse mediante una función x donde a cada jugador del conjunto N se le asigna un número real que representa el pago que obtendrá ese jugador en el juego. Esta función puede expresarse mediante el vector de pagos $x = (x_1, x_2, \dots, x_n)$ donde x_i representa el pago al jugador i . A continuación, teniendo en cuenta esto, vamos a definir dos restricciones lógicas para el vector de pagos:

Definición 2.3. Principio de individualidad racional. Para que los jugadores acepten la distribución de costes propuesta por el vector de pagos (x) , tienen que recibir un pago inferior o igual al que recibirían si jugasen solos. Por tanto, el principio de individualidad racional se puede definir como:

$$x_i \leq v(\{i\}), \forall i = 1, 2, \dots, n.$$

Definición 2.4. Principio de eficiencia. Se dice que un vector de pagos $x = (x_1, x_2, \dots, x_n)$ satisface el principio de eficiencia cuando

$$\sum_{i \in N} x_i = v(N)$$

Siendo $v(N)$ el coste total de la gran coalición N . Así, este principio impone que, si se forma la gran coalición N , el coste de esta será repartido en su totalidad por los miembros que la forman.

Definición 2.5. El Core o Núcleo. El Core de un juego de costes (N, v) se define como el conjunto:

$$C(N, v) = \{x \in I(N, v), x(S) \leq v(S), \forall S \subseteq N\}.$$

Donde $I(N, v)$ corresponde con el conjunto de imputaciones del juego de costes (N, v) , siendo dicho conjunto de imputaciones el conjunto de todos los repartos eficientes que verifican además la propiedad de racionalidad individual y cuyo conjunto puede ser vacío. Formalmente, definimos el conjunto de imputaciones del juego (N, v) como:

$$I(N, v) = \{x \in I^*(N, v) : x_i \leq v(i), \forall i \in N\}$$

Siendo $I^*(N, v)$ el conjunto de preimputaciones del juego (N, v) igual a

$$I^*(N, v) = \{x \in \mathbb{R}^N : \sum_{i \in N} x_i = v(N)\}$$

Asimismo, en base a la racionalidad coalicional, que puede definirse formalmente como $x(S) = \sum_{i \in S} x_i \leq v(S)$, debemos proporcionarle a cada coalición S que esté contenida en N un coste que sea, como máximo, el que cada coalición obtiene por sí misma. También, en base a la condición de eficiencia, un reparto " x " será eficiente si distribuye el valor de la gran coalición ($v(N)$) entre los jugadores, es decir, si $x(N) = x_1 + \dots + x_N = v(N)$

Los autores, en su libro Teoría de juegos cooperativos, exponen que "El *core* es, por tanto, el conjunto de vectores de pagos que ofrece a cada coalición que puede formarse sobre N un beneficio, al menos, igual que el que esta coalición puede conseguir por sí misma " En el caso de estar ante un juego de costes, este debe ser como máximo el coste que la coalición tiene por sí misma. Por tanto, los elementos del *Core* son aceptables para todas las coaliciones $S \subseteq N$. Intuitivamente, vemos que esto es una solución para un juego de costes (N, v) . El *Core* de un juego cooperativo satisface, además, ciertas propiedades matemáticas. No entraremos en detalle con estas propiedades, pero, el *Core* de un juego (N, v) presenta un conjunto que es cerrado, acotado y convexo. Así, estos autores en su monografía Juegos coalicionales aclaran que puede darse el caso de que haya juegos en los que el *Core* es vacío y, por tanto, no se pueda obtener un vector de pagos que contente a todas las coaliciones.

Definición 2.6. El least core. Este concepto de solución fue, inicialmente, propuesto por Shapley y Shubik (1954) y más tarde por Maschler et al (1979). Así, en base a lo

expuesto por Schulz y Uhan (2013), el least core de un juego cooperativo es el conjunto de asignaciones de costes x que son soluciones óptimas a la programación lineal:

$$z^* = \min z$$

s. a:

$$x(N) = v(N),$$

$$x(S) \leq v(S) + z \text{ para todo } S \subseteq N, S \neq \emptyset, N$$

El valor óptimo z^* de la programación lineal es el valor del least core del juego. Esta programación lineal puede ser equivalente a

$$z^* = \min_{x: x(N)=v(N)} \max_{S \subset N, S \neq \emptyset, N} e(x, S)$$

Donde $e(x, S) = x(S) - v(S)$ para todo $S \subseteq N$. Siendo, la cantidad $e(x, S)$ interpretada como el exceso de una coalición S bajo la asignación de costes x , es decir, es el coste adicional que paga S cuando el coste se asigna de acuerdo con x . Una asignación de costes en el least core, por lo tanto, minimiza la insatisfacción máxima de cualquier coalición. En este sentido, el least core contiene aquellas asignaciones de costes que son menos objetables. Asimismo, el least core contiene todas las asignaciones de costes que son eficientes y estables que maximizan la ganancia mínima de cualquier coalición. Si el núcleo o core de un juego cooperativo es vacío, es decir no hay un vector de pagos que satisfaga a todas las coaliciones, el valor central mínimo z puede verse como la penalización mínima que debemos aplicar a una coalición por actuar de manera independiente para fomentar la cooperación al garantizar la existencia de una asignación de costos eficiente y estable.

Definición 2.7. El valor de Shapley. El valor de Shapley se calcula promediando los vectores de contribuciones marginales asociados a todas las posibles ordenaciones de los jugadores. Así, el valor de Shapley de un juego es el reparto que viene dado por:

$$Sh(N, v) = \frac{1}{|\Pi^N|} \sum_{\pi \in \Pi^N} (v(Pre^\pi(i) \cup i) - v(Pre^\pi(i)))$$

$$Sh(N, v) = \frac{1}{|N|!} \sum_{S \subset N: i \in S} (|S| - 1)! (|N| - |S|)! (v(S) - v(S \setminus i)) \text{ para cada } i \in N$$

Los coeficientes en las fórmulas anteriores indican que todas las coaliciones del mismo tamaño son igualmente probables. Así, $(|S| - 1)! (|N| - |S|)!$ es el número de ordenaciones para las cuales los predecesores del jugador i son los jugadores de $S \setminus i$.

Así, se puede interpretar la expresión $Sh_i(N, v)$ como la media que tiene el jugador i por participar en el juego (N, v) .

Definición 2.8. El nucleolo. Definimos el nucleolo de un juego como:

$$Nu(N, v) = \{x \in I(N, v): \theta(x) \leq_L \theta(y), \text{ para todo } y \in I(N, v)\}$$

Dado un reparto $x \in \mathbb{R}^{|N|}$, sea $\theta(x)$ la $2^{|N|} -$ tupla cuyas componentes son los excesos, (siendo el exceso de una coalición S para un vector x , la diferencia entre el valor de la coalición y lo que le da la asignación x , matemáticamente, $e(S, x) = x(S) - V(S)$, $\forall S \subset N$) del valor de todas las coaliciones con respecto a x , ordenados de mayor a menor, es decir:

$$\theta_i(x) \geq \theta_j(x) \text{ si } 1 \leq i \leq j \leq 2^{|N|}$$

Definición 2.9. El Core-Center. El Core-Center de un juego con un núcleo no vacío, se puede definir, en base a lo expuesto por González y Sánchez (2007), como la esperanza matemática de la distribución uniforme⁵ del núcleo, conllevando, dicha distribución uniforme, que todos los puntos del Core presenten el mismo valor. Así,

⁵ Es una distribución continua que modela un rango de valores con igual probabilidad. Si la variable es una variable real, la distribución uniforme se especifica mediante cotas inferior y superior.

podemos denotar como $U(A)$ a la distribución uniforme definida sobre el conjunto A y como $E(\mathbb{P})$ a la media o esperanza matemática de la distribución de probabilidad $\mathbb{P}$.

El Core-center se puede interpretar de una manera muy simple. Imaginemos que los jugadores que forman el juego quieren obtener un reparto dentro del Core⁶ de éste, como bien explicamos en la definición de este concepto, en el Core se presentan distintas asignaciones, y unas benefician más a ciertos jugadores que a otros. Entonces, el Core-center se define como el centro del Core si consideramos que su “masa” está uniformemente distribuida.

Así, en base a lo expuesto, el Core-center se puede definir de una manera formal como:

Siendo v un juego que presenta un Core no vacío. Entonces, el Core-Center de v , $\mu(v)$, es definido de la forma $\mu(v) := E(U(C(v)))$.

Definición 2.10. Juego cuasi-equilibrado. Un juego (N, v) , se dice que es de compromiso admisible o cuasi-equilibrado (teniendo en cuenta que $m(N, v)$ es el vector de mínimos derechos y $M(N, v)$ el vector de los pagos de utopía de los jugadores), si cumple las siguientes condiciones:

1. $m(N, v) \leq M(N, v)$
2. $\sum_{i \in N} m_i(N, v) \leq v(N) \leq \sum_{i \in N} M_i(N, v)$

Definición 2.11. El τ – valor (tau-valor). Sea (N, v) un juego TU cuasi-equilibrado. Se define el τ – valor como el reparto que verifica:

$$\tau(N, v) = m(N, v) + \alpha(M(N, v) - m(N, v))$$

Teniendo en cuenta que $\alpha \in \mathbb{R}$ t. q. $\sum_{i \in N} \tau_i(N, v) = v(N)$

El τ – valor elige un elemento en el core-cover de tal forma que cada jugador reciba su mínimo derecho más una cantidad añadida que depende de su pago de utopía. Siendo el *pago de utopía* y el *mínimo derecho del jugador i* iguales a $M_i(N, v) = v(N) - v(N \setminus i)$ y $m_i(N, v) = \max_{S \subset N: i \in S} \{v(S) - \sum_{j \in S \setminus i} M_j(N, v)\}$, respectivamente.

⁶ Véase *definición 2.5*.

El *pago de utopía de un jugador* es la contribución marginal del jugador a la gran coalición. Este pago representa lo máximo que este podría pedir. Así, en el caso de que el jugador pidiera más, entonces sería rentable para la gran coalición no tenerlo en cuenta y formar su subjuego de $n - 1$ jugadores.

Por otro lado, el *mínimo derecho de un jugador* se puede definir como la cantidad que le queda a cada coalición a la que pertenece si el resto de los jugadores de la coalición se llevasen su pago de utopía.

Definición 2.12. El core-cover. Sea $(N, v) \in S^N$, podemos definir el core-cover como el conjunto:

$$CC(N, v) = \{x \in I(N, v) : m(N, v) \leq x \leq M(N, v)\}$$

Asimismo, si se cumplen que las condiciones de que el juego (N, v) sea cuasi-equilibrado, podemos afirmar que el core-cover del juego no es vacío ($CC(N, v) \neq \emptyset$) y que, en consecuencia, existe asignaciones en el core-cover.

Definición 2.13. Familia equilibrada. Una familia $\mathcal{F}$ de coaliciones no vacías de N se dice equilibrada si existen unos pesos positivos $\{\delta_S\}_{S \in \mathcal{F}}$, $\delta_S > 0$ para todo $S \in \mathcal{F}$, tales que $\sum_{S \in \mathcal{F}} \delta_S e^S = e^N$, siendo $e^S \in \mathbb{R}^N$ el vector característico de S , es decir $e_i^S = 1$ si $i \in S$ y $e_i^S = 0$ si $i \notin S$. Equivalentemente también tenemos, $\sum_{S \in \mathcal{F} : i \in S} \delta_S = 1$ para todo $i \in N$.

Definición 2.14. Criterio de Kohlberg para el nucleolo. Consideremos un juego con utilidad transferible $(N, v) \in G^N$ y una imputación $x \in I^*(N, v)$. Sea $F^0 = \{\{i\} : i \in N\}$. Denotemos por $k(x)$ el número de excesos distintos de la imputación x . El reparto x es el nucleolus del juego si y solo si para $1 \leq j \leq k(x)$ se verifica que $\bigcup_{t=1}^j F^t \cup F^0$ es una familia equilibrada débilmente, siendo F^t el conjunto de coaliciones asociadas al t -ésimo exceso en el orden no decreciente. Para que una familia sea equilibrada débilmente, los δ_S asociados a las coaliciones de F^t tienen que ser estrictamente positivos pero los pesos asociados a las coaliciones individuales pueden ser cero, es decir, $\delta_i \geq 0$ para todo $i \in N$.

Definición 2.15. El Prenucleolo. Definimos el prenucleolo del juego (N, v) como:

$$PreNu(N, v) = \{x \in I^*(N, v) : \theta(x) \leq_L \theta(y), \text{ para todo } y \in I^*(N, v)\}$$

2.3. Los juegos cooperativos en el caso del TSP

Encontrar la ruta menos costosa visitando un número dado de ciudades, partiendo y regresando de la misma ciudad, es el objetivo de uno de los problemas de optimización combinatoria más conocidos de la investigación operativa, el problema del viajante de comercio (con sus siglas en inglés TSP).

Fishburn and Pollak (1983) comentan que, en muchos casos, en el problema del viajante de comercio, el coste del viaje debe ser distribuido entre los agentes o ciudades visitadas. Es por ello por lo que, en el TSP, las soluciones de teoría de juegos cooperativos representan un papel muy importante. Uno de los principales motivos es debido a que, mediante la cooperación entre los agentes, se puede llegar a obtener mejores resultados (ya sean menores costes o mayores beneficios) que si actúan cada uno por separado.

Así, en base a la definición realizada anteriormente sobre un juego cooperativo TU, el juego asociado al problema del viajante de comercio estará formado por el coste del jugador cuando actúa de forma individual $v(i)$, por el coste asociado a cuando los jugadores cooperan entre ellos, es decir, a la coalición S que denominaremos formalmente como $v(S)$, $\forall S \subseteq N$ y por el coste que se obtiene cuando todos los jugadores cooperan o el valor de la coalición total, $v(N)$, siendo $N = \{1, 2, 3, \dots, n\}$. Así, dichos costes se exponen en la función característica del juego de costes (N, v) .

Es de gran interés observar cómo, a lo largo de los años, diversos autores han obtenido distintas soluciones de un mismo concepto de juegos cooperativos en las diferentes versiones del viajante de comercio, como también han ido evolucionando la interpretación de las principales soluciones y el cálculo de estas. Un ejemplo de ello puede ser la transformación o modificación que realizaron los autores Potters et al en

1992 con respecto a ciertas definiciones establecidas por Fishburn y Pollak en 1983 sobre el juego del viajante de comercio con ruta fija.

Por ello, en este trabajo hemos querido ilustrar el papel de los juegos cooperativos a lo largo de la historia y en la actualidad, así como su papel en los distintos tipos de TSP, y las distintas soluciones obtenidas mediante las principales soluciones de los juegos cooperativos que veremos en los siguientes epígrafes.

A la hora de realizar los cálculos de algunos conceptos de soluciones de los juegos cooperativos, hemos recurrido a la herramienta TUGlabWeb (enlace en [TUGlabWeb](#)). TUGlabWeb es una herramienta sencilla que nos permite obtener, mediante la introducción de los datos de la función característica del juego cooperativo a analizar diferentes soluciones de juegos cooperativos, así como las representaciones gráficas de estos de juegos formados por, como máximo, cuatro jugadores o agentes.

Ejemplo ilustrativo

Los autores Derks y Kuipers ilustraron en 1997 en uno de sus artículos un ejemplo sobre un juego del viajante TU. Dicho juego es representado mediante un grafo no dirigido y simétrico (véase *Figura 13*) $G = (V, A)$ con un número de agentes $N = \{1,2,3\}$.

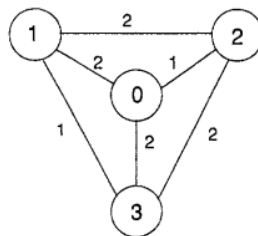


FIGURA 13. GRAFO DE UN JUEGO TU. FUENTE: DERKS Y KUIPERS(1997)

Consideremos que estamos ante un juego de costes, donde el objetivo del viajante de comercio es minimizar el coste (o distancia) al que se enfrenta realizando una ruta partiendo del nodo inicial (nodo 0), recorriendo cada ciudad o lugar de la ruta una vez y regresando al lugar de partida. Entonces, en base al objetivo del juego, las diferentes posibles rutas a realizar por el agente son:

Número	Ruta	Coste
1 ^a	0-2-1-3-0	6
2 ^a	0-2-3-1-0	6
3 ^a	0-1-3-2-0	6
4 ^a	0-3-1-2-0	6
5 ^a	0-1-2-3-0	8
6 ^a	0-3-2-1-0	8

Como vemos, hemos obtenido seis posibles rutas, siendo cuatro de ellas de coste mínimo (igual a seis unidades) y dos de ellas, concretamente la ruta número cinco y seis de un coste mayor (exactamente de ocho unidades). Teniendo en cuenta que buscamos la ruta que minimice el coste de esta, vamos a descartar las dos últimas rutas obtenidas ya que presentan un coste mayor al resto.

El resto de las rutas obtenidas tienen un coste igual a seis. En este caso, vamos a escoger la ruta 0 – 2 – 3 – 1 – 0 a la que denominaremos, a partir de ahora, *Caso 1*.

En el *Caso 1*, estamos, por tanto, ante un juego de costes (N, v) que se compone de un conjunto de jugadores $N = \{1, 2, 3\}$ y cuya función de costes viene dada por:

S	$\{1\}$	$\{2\}$	$\{3\}$	$\{1, 2\}$	$\{1, 3\}$	$\{2, 3\}$	$\{1, 2, 3\}$
$v(S)$	4	2	4	5	5	5	6

que, de forma abreviada podemos definir como:

$$v(S) = [4, 2, 4, 5, 5, 5, 6]$$

Como vemos, los valores de las funciones de costes individuales, $v(i)$, son iguales a $v(1) = 4$, $v(2) = 2$, $v(3) = 4$ y corresponden con el coste al que se incurre realizando el recorrido desde el lugar de partida a cada *ciudad* i , ida y vuelta.

Por otro lado, tenemos los valores de la función característica de los costes de las coaliciones ($v(12)$, $v(13)$, $v(23)$) siendo todas iguales a cinco y, por último, el valor de la coalición total ($v(N)$) siendo este igual a seis.

Así, teniendo en cuenta la definición 2.1 del epígrafe 2.2, los *costes marginales* obtenidos de las tres ciudades son todos iguales a uno. Es decir, $m_1 = m_2 = m_3 = 1$ siendo m_i el coste marginal de la ciudad i .

Asimismo, en base a la definición del Core realizada anteriormente, hemos obtenido el Core del juego del ejemplo ilustrativo, siendo este la envoltura convexa de los siguientes puntos, así como su representación gráfica (véase *Figura 14*).

$$C(N, v) = \text{conv}\{(4,1,1)(3,2,1)(1,2,3)(1,1,4)\}$$

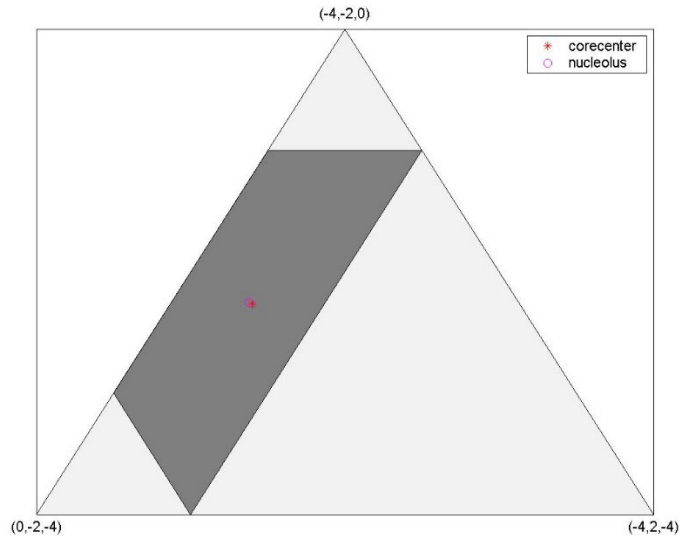


FIGURA 14. CORE OBTENIDO DEL CASO 1. FUENTE: ELABORACIÓN PROPIA MEDIANTE TUGLABWEB.

Por otro lado, es de interés calcular el least core del ejemplo ilustrativo. Teniendo en cuenta el valor de la función de costes definida anteriormente. Ahora, vamos a proceder al cálculo del least core. Primero restamos a cada valor de la función de costes un valor ε :

$$v_\varepsilon(S) = [4 - \varepsilon, 2 - \varepsilon, 4 - \varepsilon, 5 - \varepsilon, 5 - \varepsilon, 5 - \varepsilon; 6]$$

Siendo v_ε igual al valor obtenido de la resta del juego de costes ($v(S)$) y el valor épsilon (ε). Como vemos, hemos restado dicho valor a los costes individuales ($c_i = v_i$) y a las

coaliciones intermedias ($c_{ij} = v_{ij}$) pero no al valor de la coalición o coste total del juego ($c(N) = v(N)$):

$$v_\varepsilon = [v(1) - \varepsilon, v(2) - \varepsilon, v(3) - \varepsilon, v(12) - \varepsilon, v(13) - \varepsilon, v(23) - \varepsilon; v(N)]$$

Así, realizando los cálculos pertinentes, obtenemos un least core del juego de costes que viene dado por la envoltura convexa de los vértices:

$$LC(N, v) = \text{conv}\left\{\left(3, \frac{3}{2}, \frac{3}{2}\right), \left(\frac{3}{2}, \frac{3}{2}, 3\right)\right\}$$

Una de las principales soluciones de los juegos cooperativos es el valor de Shapley. Teniendo en cuenta el valor de la función de costes de nuestro juego TU y utilizando las diferentes ordenaciones de los jugadores, obtenemos los vectores de las contribuciones marginales:

Orden	Jugador		
	1	2	3
123	4	1	1
132	4	1	1
213	3	2	1
231	1	2	3
312	1	1	4
321	1	1	4
	14	8	14

Entonces, el valor de Shapley, que es la media aritmética de los vectores de contribuciones marginales, vendría dado por:

$$Sh(N, v) = \frac{1}{6}(14, 8, 14) = \left(\frac{7}{3}, \frac{4}{3}, \frac{7}{3}\right)$$

A continuación, vamos a calcular otra solución principal de los juegos cooperativos: el nucleolo. Así, el nucleolo del juego (N, v) es:

$$Nu(N, v) = \{(2.25, 1.5, 2.25)\}$$

Sin embargo, tras realizar los cálculos de las soluciones de algunos conceptos de juegos cooperativos nos puede surgir una pregunta ¿Qué sucede en el caso de que el viajante escoja una ruta que no es óptima? Pues bien, ahora, supongamos que, por el contrario, el viajante escoge la siguiente ruta: $0 - 1 - 2 - 3 - 0$ con un coste igual a ocho unidades. Como vemos, estamos ante una ruta con un orden que no es óptimo y con un coste superior a la ruta óptima obtenida en el *Caso 1*. A este caso lo denominaremos *Caso 2*. A continuación, vamos a obtener las principales soluciones de los juegos cooperativos, ya calculadas para el *Caso 1*, y analizaremos las diferencias obtenidas entre ambos casos.

Estamos ante un juego de costes (N, v) que se compone de un conjunto de jugadores $N = \{1, 2, 3\}$ y cuya función de costes viene dada por:

S	$\{1\}$	$\{2\}$	$\{3\}$	$\{1, 2\}$	$\{1, 3\}$	$\{2, 3\}$	$\{1, 2, 3\}$
$v(S)$	4	2	4	5	5	5	8

Como vemos, la única diferencia con el *Caso 1*, es el valor de la función de costes de la coalición total ($v(N)$), que podemos abreviar de la forma $v(S) = [4, 2, 4, 5, 5, 5, 8]$. Los *costes marginales del Caso 2* son los siguientes:

$$m_1 = m_2 = m_3 = 3$$

En este caso, los costes marginales aumentan, siendo la diferencia entre cada caso de dos unidades (en el *Caso 1* los costes marginales fueron de una unidad). Ahora, vamos a proceder al cálculo del Core de la misma forma que en el caso anterior. Como era de esperar, el Core de este juego formado por una ruta con un orden que no es óptimo es igual al vacío y, por tanto, no se obtiene un vector de pagos que satisfaga a todas las coaliciones. Esto puede dar hincapié a que los agentes no cooperen y busquen su beneficio propio de forma individual.

$$C(N, v) = \emptyset$$

En lo que respecta al least core, vamos a realizar el mismo procedimiento que en el *Caso 1*. Teniendo una función de costes y realizando los cálculos pertinentes, obtenemos un least core para el *Caso 2* que viene dado por la envoltura convexa de los vértices:

$$LC(N, v) = \text{conv} \left\{ \left(\frac{5}{2}, \frac{5}{2}, 3 \right), \left(3, \frac{5}{2}, \frac{5}{2} \right) \right\}$$

Ahora, vamos a calcular otra solución principal de los juegos cooperativos: el valor de Shapley. Teniendo en cuenta el valor de la función de costes expuesta anteriormente y, utilizando las diferentes ordenaciones de los jugadores obtenemos los vectores de las contribuciones marginales:

Orden	Jugador		
	1	2	3
123	4	1	3
132	4	3	1
213	3	2	3
231	3	2	3
312	1	3	4
321	3	1	4
	18	12	18

Entonces, el valor de Shapley, que es la media aritmética de los vectores de contribuciones marginales, vendría dado por:

$$Sh(N, v) = \frac{1}{6}(18, 12, 18) = (3, 2, 3)$$

Por último, hemos calculado el nucleolo del juego (N, v) del *Caso 2*; este tiene un valor igual a:

$$Nu(N, v) = \{(3, 2, 3)\}$$

Como vemos, la diferencia más destacable entre ambos casos es el resultado obtenido a la hora de calcular el Core del juego. Para el *Caso 1*, hemos obtenido un Core igual a la envoltura convexa de los puntos $\{(4, 1, 1)(3, 2, 1)(1, 2, 3)(1, 1, 4)\}$ y, por tanto, un vector de pagos que satisface a todos los jugadores. Por el contrario, para el *Caso 2* (recordemos que esta ruta tiene un orden no óptimo) no hemos obtenido un reparto eficiente, es decir, hemos obtenido un Core o núcleo vacío. Sin embargo, este resultado puede depender, no solo del orden de la ruta, sino también de la versión del juego del viajante de comercio a la que estemos calculando este concepto.

En la actualidad, existen varias versiones del juego del viajante de comercio (con sus siglas en inglés, TSG), pero las más destacadas son el juego del viajante con ruta fija y el juego del viajante de comercio. La diferencia más importante entre ambos juegos es la forma en la que definen la función de costes para las coaliciones intermedias y la coalición total que, para el juego del viajante con ruta fija, tiene un orden ya determinado. En lo que respecta al tiempo computacional que se requiere para la resolución de ambas versiones, este depende de la complejidad computacional de cada problema. Así, el problema del viajante se ha considerado un tipo de problema de complejidad computacional (véase *epígrafe 1.4*).

En el campo de los juegos cooperativos (JC), en base a lo expuesto por los autores Derks y Kuipers (1997), la complejidad computacional del cálculo de algunos conceptos

principales de los JC (como puede ser el núcleo, el valor de Shapley, etc) del problema del viajante, puede ser, como mínimo, exponencial en el número de agentes. Esto es debido a que la información en la función característica ya es exponencial en el número de agentes y, teniendo en cuenta que cuando se utiliza un algoritmo para obtener la solución de alguno de estos conceptos este debe revisar toda la información disponible, el tiempo que tarda en obtener toda esa información no es precisamente polinomial. Por el contrario, si la función característica del juego no se da como una lista de valores, sino mediante una descripción implícita y esta última es polinomial en el número de agentes, entonces podemos esperar que haya algoritmos de tiempo polinomial para resolver dicho juego. El juego del viajante de comercio es un claro ejemplo para el cual tenemos una descripción polinomial del juego, pero con el que es poco probable que exista un algoritmo polinomial.

Asimismo, en la actualidad, resulta ‘irreal’ que, debido a los cambios que pueden producirse en una ruta, los autores no contemplasen que pudiera haber una modificación o un cambio en la ruta y que el viajante o agente tuviera que visitar únicamente una vez cada destino o ciudad, así como su lugar de origen o nodo cero. Es por ello por lo que el autor Yengin (2012), contempló la posibilidad de que la ruta pueda ser modificada y que el viajante pueda regresar a su lugar de origen o visitar una ciudad más de una vez.

Dicha posibilidad fue desarrollada y definida en una nueva versión del problema del viajante con ruta fija: *el problema del viajante con ruta fija y con citas*. Esta versión se explica de forma breve en el epígrafe 2.3.2, así como el juego de citas y las soluciones obtenidas de esta versión mediante los juegos cooperativos.

2.3.1 El juego del viajante con ruta fija

El juego del viajante con ruta fija fue definido por Fishburn y Pollak (1983), mediante el siguiente ejemplo: “Una persona, financiada por varios patrocinadores, visita, partiendo

de su ciudad, las ciudades de los patrocinadores y regresa al origen. El coste total del viaje debe ser compartido por los patrocinadores”

De manera más sencilla, imaginemos que un proveedor partiendo de su almacén (su ubicación origen) debe visitar un conjunto de clientes (establecidos en diferentes ubicaciones) siguiendo una ruta fijada con anterioridad. El proveedor, debe visitar a cada cliente y regresar a su almacén o ubicación de origen. Una de las posibles consecuencias de que la ruta esté fijada con anterioridad es que dicha ruta puede no ser la de menor coste. Asimismo, dicho coste suele estar definido en términos de distancias entre las ubicaciones. El valor de una coalición es el coste del viaje cuando el viajante, en este caso el proveedor, sigue la ruta establecida y se salta a aquellos clientes que no son miembros de la coalición. Si se da el caso de que el proveedor no visita a un cliente, este va directamente al siguiente cliente que esté dentro de la coalición. En el caso de que todos los clientes ya hayan sido visitados, este vuelve a su ubicación de origen. Cada cliente paga, por separado, el coste atribuido al servicio que le han realizado, siendo la suma de todos esos pagos el montante final o coste total de la ruta. Uno de los objetivos de este problema es encontrar una distribución equitativa del coste total generado por el problema del viajante con ruta fija entre los clientes o ubicaciones visitadas por el agente o proveedor.

Para poder alcanzar este objetivo, los autores Potters et al (1992), propusieron la definición de algunos conceptos clave para entender cómo funciona el juego del viajante con ruta fija, así como, los autores Fishburn y Pollak definieron tres condiciones que, según su criterio, una regla de reparto debe cumplir y que fueron modificadas, con el paso del tiempo, por Potters et al.

Así, formalmente, el juego del viajante con ruta fija presenta un conjunto de ciudades a visitar denominado por $N_0 = \{0, 1, 2, \dots, n\}$, con una matriz de costes “c”, donde c_{ij} es el coste que tiene el agente para viajar de i a j. Asimismo, sería favorable que dicha matriz de costes cumpliera las siguientes propiedades:

$$c_{ii} = 0 \quad \forall i \in N_0 \quad (4)$$

$$c_{ij} \leq c_{ik} + c_{kj} \quad \forall i, j, k \in N_0 \quad (5)$$

Siendo, en términos de juegos, la condición (5) equivalente a que el juego sea N-subaditivo $v(N) \leq v(S) + v(N \setminus S)$ $S \subseteq N$ y a la desigualdad triangular (explicada anteriormente). La condición (4) establece que el coste de ir desde la ciudad i hasta esta misma ciudad sea igual a cero.

Los autores definen un *tour del viajante de comercio con ruta fija* como un tour por todas las ciudades de N_0 . Para cada coalición $S \subseteq N := \{1, 2, \dots, n\}$, cada permutación cíclica σ y cada elemento $i \in S_0 := S \cup \{0\}$ definimos:

$$p(S, i) = \min \{p \in \{1, 2, \dots, n\} \mid \sigma^p(i) \in S_0\}$$

donde σ^p es la composición de σ con si misma p veces. Entonces la permutación

$$i \in S_0 \rightarrow \sigma^{p(S, i)}(i) \in S_0$$

es una permutación cíclica de S_0 denotada por $\sigma|_{S_0}$ que serán las rutas que visiten únicamente los elementos de S y se obtendrán, dichas rutas, eliminando de σ las ciudades de $N \setminus S$ y manteniendo el mismo orden del resto de ciudades. Así, para cada coalición $S \subseteq N$, podemos definir formalmente el juego del viajante de comercio con ruta fija como:

$$v_{c, \sigma}(S) := \sum_{i \in S_0} c_{i, \sigma|_{S_0}(i)}$$

Siguiendo con nuestro ejemplo, es el proveedor (el viajante de comercio) el que decide que circuito Hamiltoniano utilizará para visitar a los clientes. El valor de una coalición se define como los costes asociados al recorrido restringido a los jugadores de S en el mismo orden que en el circuito Hamiltoniano.

Ciertos autores han propuesto diferentes algoritmos para la resolución del juego de esta variante del problema del viajante de comercio. Derks y Kuipers en 1997 proponen un algoritmo eficiente que calcula los elementos en el core en los juegos de ruta fija. En Kuipers et al (2000) y Solymosi et al (1998) se definen $O(n^4)$ algoritmos para calcular el nucleolo de un juego de ruta fija.

Como bien comentamos anteriormente, Fishburn y Pollak propusieron tres condiciones que cualquier regla de reparto debe cumplir:

- a) La suma de los pagos de cada agente debe coincidir con el coste total del viaje (eficiencia).
- b) Ningún agente pagará más de lo que cueste un viaje directo desde el origen del viajante hasta la ciudad del agente y vuelta al origen (racionalidad individual).
- c) Si un *agente i* tiene un coste marginal superior al de *agente j*, entonces la contribución de *i* ha de ser superior que la de *j*.

El problema que observaron Fishburn y Pollak es que todas las condiciones no se pueden dar al mismo tiempo y, por tanto, son incompatibles.

Ejemplo de un juego del viajante con ruta fija

Supongamos que estamos ante un juego de costes que presenta el siguiente grafo (véase *Figura 15*) expuesto por los autores Fishburn y Pollak. Sea $N = \{1,2,3,4\}$, los autores, en este ejemplo, tomaron la ruta, a la que denominaremos formalmente como "*r*", $r = (i_1, i_2, i_3, \dots, i_5) = (0,1,2,3,0)$ obteniendo un coste total de 1500 unidades y unos costes marginales iguales a (640,670,640). Como vemos, este resultado incumple algunas de las condiciones expuestas:

Según la *condición c)* el agente que tenga un coste marginal superior debe pagar más. En este ejemplo, vemos que el agente dos tiene un coste marginal superior a los agentes uno y tres, que tienen el mismo coste marginal.

Según la *condición b)* ningún agente pagará más de lo que cueste un viaje directo desde el origen del viajante hasta su ciudad ida y vuelta; por tanto, el agente dos no debería pagar más de 220.

Ante estos resultados, vemos que es imposible alcanzar el coste total de 1500 debido a que el agente dos no pagará más de 220 y el agente uno y tres deberían pagar menos que este agente (es decir, menos de 220) por tanto, es imposible alcanzar el coste total. Esto nos muestra que las tres condiciones son incompatibles.

Ante ello, los autores Potters et al (1992) reconsideraron el ejemplo y decidieron “transformar” el juego del viajante con ruta fija al juego del *viajante con ruta óptima*. Así determinaron que la ruta de mínimo coste del ejemplo mostrado anteriormente es $r = (0,2,1,3,0)$ con un coste total igual a 890 unidades y costes marginales iguales a $(30,60,30)$. Asimismo, dicho juego presenta unos costes individuales iguales a $v(1) = v(3) = 800$ y $v(2) = 220$, unos costes de las coaliciones intermedias formadas por dos agentes iguales a $v(12) = v(23) = 860$ y $v(13) = 830$ y con el valor de la coalición total $v(N) = 890$.

Como vemos, con esta nueva ruta óptima, las condiciones expuestas por los autores Fishburn y Pollak no se cumplen, pero, en este caso, la suma de los costes marginales no es superior al coste total ($120 < 890$), por lo que es posible tomar dichos costes marginales como una contribución mínima de cada agente.

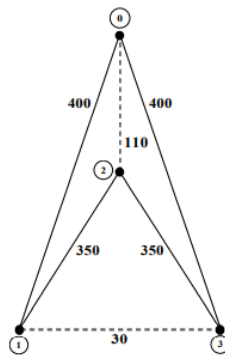


FIGURA 15. EJEMPLO PROPUESTO POR FISHBURN AND POLLAK (1983)

Ante este resultado, se estudió la manera de encontrar alguna regla de reparto que cumpliera las condiciones de eficiencia, racionalidad y una nueva condición que sustituye a la condición c) expuesta por los otros autores:

Condición d) Cada agente, tiene que pagar, como mínimo, su coste marginal.

Así, si la matriz de costes “c” cumple las propiedades (4) y (5), Potters et al (1992) propusieron una regla de reparto diferente a la propuesta por los anteriores autores.

Dicha regla de reparto no suele estar en el núcleo, pero sí cumple las propiedades a), b) y d) y, en base a esta, la regla de reparto viene dada por:

$$\phi_i = \lambda v_{c,\sigma}(i) + (1 - \lambda)(v_{c,\sigma}(N) - v_{c,\sigma}(N \setminus i))$$

Donde λ se elige tal que $\sum_{i \in N} \phi_i = v_{c,\sigma}(N)$ y $v_{c,\sigma}(N)$ corresponde con el valor de la función de costes teniendo en cuenta la matriz de esta y la permutación σ . Así, los autores realizaron el cálculo del núcleo de este juego obteniendo un núcleo que viene dado por:

$$C(N, v_{c,\sigma}) = \{(x_1, x_2, x_3 \in \mathbb{R}^3 / 30 \leq x_1 \leq 800; 60 \leq x_2 \leq 220; 30 \leq x_3 \leq 800; x_1 + x_2 + x_3 = v(N) = 890)\}$$

A continuación, vamos a obtener el reparto mediante la regla propuesta por los autores. Dicho reparto corresponde con una de las principales soluciones de los juegos cooperativos: el τ - *valor*.

$$x_1 = 800\lambda + (1 - \lambda)30$$

$$x_2 = 220\lambda + (1 - \lambda)60$$

$$x_3 = 800\lambda + (1 - \lambda)30$$

$$x_1 + x_2 + x_3 = 890$$

Así, obtenemos un reparto igual a $(x_1, x_2, x_3) = (378.76, 132.48, 378.76)$ que, como podemos observar, en este caso si pertenece al núcleo y cumple las condiciones a), b) y d).

En base a lo expuesto por estos autores, si nos fijamos en el ejemplo ilustrativo mostrado anteriormente, vemos que este cumple también las propiedades de eficiencia, racionalidad individual y la condición de que cada agente tiene que pagar, como mínimo, su coste marginal (las condiciones a), b) y d), respectivamente).

A continuación, vamos a calcular algunos conceptos de soluciones del ejemplo ilustrado por los autores mediante la herramienta TUGLabWeb.

Uno de los objetivos cuando estamos ante un juego del problema del viajante es encontrar un conjunto de vectores de pagos que ofrecer a cada coalición. Dicho conjunto de vectores de pago se puede obtener mediante el Core (véase *definición 2.5*). El Core o núcleo del juego expresado en función de los puntos extremos formado por la ruta elegida por los autores Potters et al (1992) es igual a:

$$C(N, v_{c,\sigma}) = \text{conv}\{(800,60,30)(640,220,30)(30,220,640)(30,60,800)\}$$

Con un Core-Center $(N, v_{c,\sigma})$ igual a $(376.55, 136.91, 376.55)$ que, como vemos, no difiere mucho del reparto obtenido por los autores mediante la regla de reparto propuesta por estos (ϕ_i) . La representación gráfica del Core de este juego viene dada por:

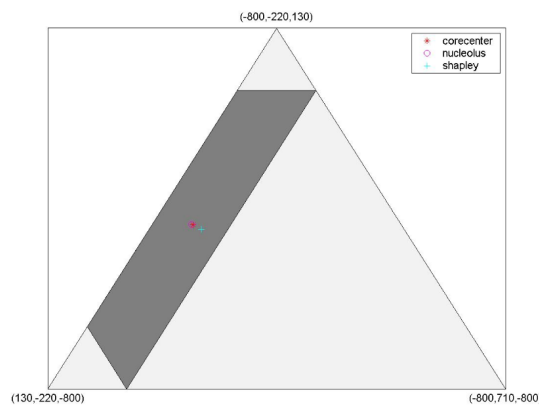


FIGURA 16. REPRESENTACIÓN GRÁFICA DEL CORE DEL JUEGO PROPUESTO. FUENTE: FISHBURN AND POLLAK (1983)

Asimismo, unos de los principales fines cuando estamos ante una decisión conjunta, por ejemplo, en el panorama empresarial a la hora de realizar un reparto de dividendos o de costes en una empresa, es encontrar un reparto que cumpla ciertas propiedades o axiomas favorables como la eficiencia, es decir, que los dividendos se repartan entre todos los agentes que forman el conjunto y que dicho reparto se realice de forma que cada uno obtenga una cantidad en función de su aportación.

Para ilustrar este concepto, vamos a utilizar nuestro ejemplo anterior, en el cual los costes totales se deberían de repartir en función de las distancias que separan al proveedor de sus clientes. Es decir, el cliente que se encuentra más cerca del proveedor deberá pagar un coste menor que el cliente que esta más lejos de este. Dicho reparto, lo podemos obtener mediante una de las principales soluciones de los juegos cooperativos: El *valor de Shapley*. Como bien comentamos en la *definición 2.7* el valor de Shapley se obtiene promediando los vectores de contribuciones marginales asociados a todas las posibles ordenaciones de los jugadores. A continuación, vamos a obtener el valor de Shapley de este juego teniendo en cuenta su función de costes:

$$v(S) = (800, 220, 800, 860, 830, 860, 890)$$

Ahora, utilizando las diferentes ordenaciones de los jugadores y el valor de la función de costes, obtenemos los vectores de las contribuciones marginales:

Orden	Jugador		
	1	2	3
123	800	60	30
132	800	60	30
213	640	220	30
231	30	220	640
312	30	60	800
321	30	60	800
	2330	680	2330

Entonces, el valor de Shapley, que es la media aritmética de los vectores de contribuciones marginales, vendría dado por:

$$Sh(N, v) = \frac{1}{6} (2330, 680, 2330) = (388.33, 113.33, 388.33)$$

Asimismo, es de gran interés conocer el nucleolo. El nucleolo del juego (N, v) es igual a:

$$Nu(N, v) = \{(375, 140, 375)\}$$

Uno de los criterios que se suelen utilizar para comprobar que el conjunto de asignaciones o un reparto " x " pertenece al Nucleolo es el criterio de Kohlberg (véase *definición 2.14*). Para calcular este criterio, obtenemos previamente los vectores de excesos de las coaliciones. Dichos excesos se calculan restando al valor de la coalición el valor del reparto obtenido en el Nucleolo. Así, para que el reparto x obtenido pertenezca al Nucleolo, las familias del juego (N, v) tienen que ser equilibradas (véase *definición 2.13*). Veamos todo esto de forma aplicada en nuestro ejemplo.

Entonces, teniendo en cuenta la definición del vector de excesos realizada en la *definición 2.8* y, que el conjunto de asignaciones x es igual a $x = (375, 140, 375)$, los vectores de excesos de este ejemplo son:

$$\theta(x) = (80, 80, 345, 345, 425, 425)$$

Obteniendo, por ejemplo, el vector de excesos de la coalición individual del agente uno de la forma: $(1) = v(1) - x(1) = 800 - 375 = 425$ unidades y calculando de forma análoga el resto de los vectores de excesos. Así, los valores del vector de excesos $\theta(x)$ obtenidos corresponden a los de las coaliciones: $v(2), v(13), v(12), v(23), v(1)$ y $v(3)$ respectivamente.

Como bien comentamos anteriormente, para que el reparto x forme parte del nucleolo, las familias del juego tienen que ser equilibradas. Para que estas sean equilibradas, deben existir unos pesos $\delta_S > 0$ tales que:

- Cliente o jugador 1: $\delta(1) + \delta(1,2) + \delta(1,3) = 1$
- Cliente o jugador 2: $\delta(2) + \delta(1,2) + \delta(2,3) = 1$
- Cliente o jugador 3: $\delta(3) + \delta(1,3) + \delta(2,3) = 1$

Siendo, las familias del juego (N, v) del ejemplo expuesto iguales a:

$$\mathcal{F}_1 = \{(2)(13)\}$$

$$\mathcal{F}_2 = \{(2)(13)(12)(23)\}$$

$$\mathcal{F}_3 = \{(2)(13)(12)(23)(1)(3)\}$$

Entonces, la familia $\mathcal{F}_1$ es equilibrada con $\delta_s = 1$, la familia $\mathcal{F}_2$ es equilibrada con $\delta_s = 1/2$, y la última familia $\mathcal{F}_3$ es equilibrada con un peso igual a $\delta_s = \frac{1}{3}$. Por tanto, x es el nucleolo del juego (N, v) .

2.3.2 El juego del viajante con ruta fija y con citas

Como bien comentamos en el punto anterior, en el juego del viajante con ruta fija, el agente, que parte de su ciudad o lugar de origen, tiene que visitar un conjunto de ciudades exactamente una vez mediante una ruta ya fijada (ya sea debido a factores externos o no) y, una vez visitada cada ciudad, regresar a su lugar de origen. En el apartado anterior, hemos visto que la ruta establecida es la ruta que el viajante de comercio realiza, y que esta puede ser modificada por motivos económicos, por ejemplo, porque el viajante encuentra una ruta en la que obtiene un menor coste (o un ahorro mayor en el caso de estar ante un problema de ahorros).

Sin embargo, en el caso del *juego del viajante con ruta fija y con citas* puede haber distintos motivos. En este caso, según el autor Yengin (2012), la ruta puede ser modificada por varios factores (no sólo por el coste de esta). Veámoslo de una manera más simple con el ejemplo del proveedor nombrado en el epígrafe anterior.

Supongamos ahora que este tiene que visitar únicamente tres clientes. El proveedor establece que va a realizar la ruta en orden ascendente por el número de cliente, es decir, primero al cliente uno, luego al dos y, por último, al tres. El día de su ruta, el cliente tres le pide al proveedor que, si puede adelantar la cita por un motivo externo urgente, por lo que no puede esperar hasta la hora establecida para recibirlo. Ante este imprevisto, el proveedor modifica su ruta y visita primero, desde su lugar de origen, al cliente tres, luego al uno y por último al cliente dos y regresa a su lugar de origen. Así, como vemos, la ruta ha sido modificada por una causa o un motivo distinto al económico.

Asimismo, en el juego del viajante con ruta fija, hemos asumido que las ciudades o, en el caso de nuestro ejemplo, los clientes, y el lugar de origen han de ser visitados solamente una vez, pero, en el caso del *juego del viajante con ruta fija y con citas* (en inglés, *TSP with appointments*), el lugar de origen puede ser visitado más de una vez. Así, siguiendo con nuestro ejemplo, imaginemos ahora que, el cliente número dos, avisa al proveedor de que no puede asistir a la cita a la hora establecida y que tiene que ser dos horas más tarde de lo previsto; en este caso, el proveedor tiene dos opciones: puede esperar cerca de la localización de este cliente una vez terminado con el anterior o irse a su lugar de origen debido a la larga espera que conlleva.

El objetivo de Yengin (2012) es encontrar una distribución justa del coste total generado por el TSP de ruta fija a través de los patrocinadores o, en el caso de nuestro ejemplo, de los clientes, teniendo en cuenta que los clientes solo comparten el total del coste de la ruta y que cada uno paga, por separado, el coste del servicio realizado por el proveedor.

Este autor, en esta nueva clase de juego, define como “ruta permisible” sobre S (siendo S una coalición concreta) aquella donde el viajante sigue visitando a los agentes que pertenecen a S de acuerdo con las citas originales. Entonces, el viajante sigue la ruta original, saltándose al agente que no está en S y, cuando éste salta al agente, el viajante regresa a su lugar de origen. El valor de la coalición S , es el coste de la ruta permisible sobre S . Esta ruta tiene sentido en algunos tipos de TSP donde el servicio que se presta realiza las citas con dos condiciones:

1. Cuando una cita es cancelada, el agente (en el caso de nuestro ejemplo el proveedor) debe seguir con el calendario establecido, sin modificar o adaptar éste.
2. Las citas están escasamente programadas. Por ejemplo, si nuestro proveedor tiene que realizar un servicio que le conlleva demasiado tiempo para completarlo, si la cita es cancelada y el proveedor tiene que esperar demasiado tiempo a la siguiente cita, el proveedor tiene dos opciones: o esperar cerca de la siguiente localización de la próxima cita o en la misma localización o, por el contrario, puede regresar a su lugar de origen y esperar allí.

En muchos casos, la primera opción presenta un gran coste, siendo dicho coste no reembolsado por los agentes (o clientes), por lo tanto, la mejor opción en base a que el objetivo es minimizar el coste de la ruta es que el viajante regrese a su lugar de origen.

Así, mediante este ejemplo, hemos dado una breve introducción de la definición del juego del viajante con ruta fija y con citas (en inglés, *fixed- route travelling Salesman games with appointments*) realizada por el autor Yengin (2012). A continuación, vamos a explicar detalladamente el problema de citas, el juego de éste y las soluciones que se pueden obtener mediante los juegos cooperativos.

El problema de citas

Los autores Chun et al (2016), definen el problema de citas, basándose en el análisis realizado por Yengin (2012) de la siguiente manera:

Tenemos un conjunto de patrocinadores (o agentes) denominado por $N = (\{1, \dots, n\})$, donde $n \geq 2$ y donde el nodo cero es un agente especial al que ellos denominan “casa”. Para cada $S \subseteq N$, digamos que $S^0 \equiv S \cup \{0\}$ y tenemos una ruta $r = (i_1, i_2, \dots, i_r)$ sobre N como una lista de agentes (incluido el lugar de origen o “casa”) que tienen que ser visitados. Estos agentes han de ser visitados teniendo en cuenta que:

1. La ruta ha de empezar y terminar en el nodo cero.
2. Cada patrocinador ha de ser visitado exactamente una vez.
3. El nodo cero, al que hemos llamado “casa”, puede ser visitados varias veces.

Por cada $i, j \in N^0$, el agente i está directamente conectado con el agente j en una ruta r , denotado como $i \succ_r j$ si el agente j es visitado inmediatamente después del agente i .

Por otro lado, para cada $i, j \in N^0$, tenemos un coste de viaje de ir desde el agente i al j que es denominado como $c_{i,j}$. El coste de visitar el lugar de origen después de cada agente es igual a cero, es decir, $c_{0,0} = 0$. Los autores en su artículo también asumen que para cada $i \in N$, $c_{i,0} = c_{0,i}$ y que para cada $i \in N^0$, $c_{i,0} \equiv c_{0,i}$. Además, también establecen el coste de la ruta r sobre N que es igual a $c(r) = \sum_{j=1}^{T-1} c_{i_j, i_{j+1}}$ siendo la

matriz de costes para N igual a $c = \{c_{i,j} \in \mathbb{R}_+ \mid \{i,j\} \subseteq N^0\}$. Una vez establecido todos estos conceptos, los autores definen un problema de citas como una triple $a = (N, c, r)$, donde, como bien comentamos anteriormente, N es el conjunto de patrocinadores o clientes, c es la matriz de costes para N , y r es la ruta fijada sobre N . Asimismo, los autores definen como $\mathcal{A}^N$ a una clase de problemas de citas para N . Así, para cada $a = (N, C, r) \in \mathcal{A}^N$ donde $r = (i_1, i_2, \dots, i_r)$ y $S \subseteq N$, tenemos una ruta restringida a S para cada $j \in \{1, 2, \dots, T\}$ definida por $r_s = (i_1^s, i_2^s, \dots, i_T^s)$, entonces:

$$i_j^s = \begin{cases} i_j & \text{si } i_j \in S \\ 0 & - \end{cases}$$

El viajante empezando desde su lugar de origen, realiza un calendario para visitar a cada agente o patrocinador perteneciente a S . Así, si $j \in S$ es el primer patrocinador o cliente que el viajante (en el caso de nuestro ejemplo, el proveedor) visita en la ruta r_s , el viajante visita al patrocinador i después de visitar al patrocinador j ($j \succ_{r_s} i$). Si en el caso de que el patrocinador i no pertenezca al conjunto S (es decir, $i \notin S$), entonces el viajante vuelve a su lugar de origen y espera ahí para la siguiente cita. Este mismo procedimiento es el que realiza el viajante hasta que todos los patrocinadores que forman el conjunto S han sido visitados y, una vez visitado todos, vuelve a su lugar de origen.

El juego de citas

Los autores Chun et al (2016), para calcular el coste de la ruta realizada por el viajante de comercio en el caso del problema de citas, utilizan las soluciones obtenidas mediante los juegos cooperativos. Para ello, el problema de citas debe ser trasladado a un juego.

Dado un conjunto de jugadores que llamaremos N y un conjunto de $S \subseteq N$ que llamaremos coalición, un juego con un conjunto de jugadores N es una función de valores reales v definida sobre todas las coaliciones de $S \subseteq N$ que satisfagan $v(\emptyset) = 0$. Siendo, el número $v(S)$ el valor de S . Así, estableciendo que Γ^N es la clase de juegos con un conjunto de jugadores N , tenemos que:

Para cada $a = (N, c, r) \in \mathcal{A}^N$, definimos como el juego de citas a la función $v_a \in \Gamma^N$, donde el valor de cada coalición es definido como el coste de la ruta restringido a esa coalición, es decir, para cada $S \subseteq N$, $v_a(S) = c(r_S)$, teniendo en cuenta que para cada $i \in N$, $v_a(\{i\}) = 2c$.

Ejemplo de un juego del viajante con ruta fija y con citas

Así, si basamos este problema en el ejemplo expuesto en este epígrafe, el juego del viajante con ruta fija y con citas puede ser útil en el caso de que el proveedor, por motivos de logística, quiera restringir la ruta y no visitar a todos los clientes en un mismo día. Este decide que dicha ruta será restringida a cierta coalición a la que denominaremos S , y solo visitará a los que formen esta.

Supongamos que tenemos un conjunto de jugadores o clientes $N = \{1,2,3,4\}$ y una ruta igual a $r = (0,1,2,3,4,0)$ con unos costes iguales a $c_1 = 12$, $c_2 = 6$, $c_3 = c_4 = c_{3,4} = 2$, $c_{1,2} = c_{1,3} = c_{1,4} = 14$, $c_{2,4} = 8$, $c_{2,3} = 9$ y una función característica:

$$v_a = [24, 12, 4, 4, 32, 28, 28, 17, 16, 6, 37, 36, 30, 19, 39]$$

Dicha función característica se obtiene sumando el coste de ir desde el nodo cero al nodo de destino. Al tener la posibilidad de que el vendedor o, en el caso de nuestro ejemplo, el proveedor, vuelva a casa, dicho coste es el total del coste de ida y vuelta de este. Es decir, si el proveedor quiere ir a visitar al cliente uno y, volver a casa, esta ruta tiene un coste de 24 unidades ($v_a(\{i\}) = 2c = 2 * 12 = 24$). Lo mismo ocurre en el caso de querer ir a visitar a clientes que no tienen una ruta que los conecte. Por ejemplo, imaginemos que el proveedor quiere visitar primero al cliente uno, luego al tres y, por ultimo al cuatro; este deberá visitar al cliente uno y, seguidamente, regresar a su almacén o casa (con un coste igual a 24 unidades), luego desde el almacén irá al cliente tres y desde el lugar de origen de este cliente visitará al cliente cuatro y volverá al almacén (con un coste igual a 6 unidades), como vemos, el coste total de esta ruta $r = (0,1,0,3,4,0)$ es igual a 30 unidades y corresponde con el valor de la función característica de la coalición $v(134)$. Así, hemos calculado de forma análoga los costes del resto de coaliciones.

Ahora, el proveedor, por motivos de logística, decide que va a realizar una ruta visitando todos los clientes excepto el cliente número tres, es decir, la ruta r será restringida a la coalición $S = \{1,2,4\}$ y la denominaremos formalmente como r_s . Entonces, teniendo en cuenta que el proveedor no visitará al cliente tres, este deberá volver a casa tras visitar al segundo cliente para, seguidamente, visitar al último y cuarto cliente. La ruta, por tanto, es igual a $r_s = (0,1,2,0,4,0)$. Con unos costes de la ruta (r_s) iguales a $v_a(S) = c(r_s) = c_{0,1} + c_{1,2} + c_{2,0} + c_{0,4} + c_{4,0} = (c_1 + c_{1,2} + c_2) + (2c_4) = c(r_{\{1,2\}}) + c(r_{\{4\}}) = 36$.

Los grafos asociados a este juego $g(a)$ y $g(a_s)$ son respectivamente, las Figuras 17 y 18.

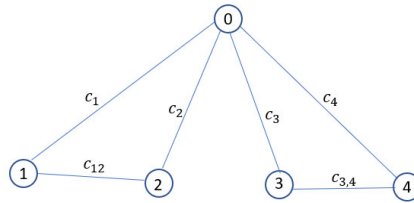


FIGURA 17. GRAFO ASOCIADO A LA RUTA

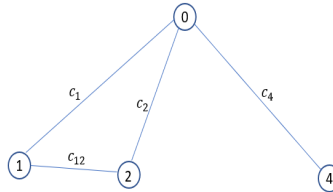


FIGURA 18. GRAFO ASOCIADO A LA RUTA RESTRINGIDA A S

De una manera más sencilla y abreviada, el juego consiste en calcular la ruta establecida teniendo en cuenta la coalición a la que está restringida. Así, el valor de esta es igual al coste de r_s y es la suma del coste de ir desde el lugar de origen hasta la ciudad uno ($c_{0,1}$), de la ciudad uno a la ciudad dos ($c_{1,2}$), de esta última ciudad al lugar

de origen o nodo cero ($c_{2,0}$), del lugar de origen a la ciudad cuatro ($c_{0,4}$) y de esta ciudad al lugar de origen ($c_{4,0}$). Esta operación se puede simplificar sumando los costes de realizar la ruta desde la ciudad uno a la dos y, seguidamente, el coste de realizar la ruta a la ciudad cuatro.

Asimismo, en esta variante del problema del viajante de comercio es de interés calcular las principales soluciones de los juegos cooperativos. Son muchos los autores que han realizado investigaciones sobre la aplicación de estas soluciones al problema del viajante con ruta fija y con citas, conocido, de forma abreviada como el juego de citas. Una de ellas es la realizada por el autor Yengin (2012) sobre una de las principales soluciones de los juegos cooperativos: El valor de Shapley.

Soluciones con los juegos cooperativos del juego de citas

El autor, Yengin comenta en su artículo "Characterizing the Shapley value in fixed-route traveling salesman problems with appointments" que, en el caso de un juego de citas, debido a la estructura de la red, cada agente coopera esencialmente con dos conjuntos de agentes: el conjunto de agentes que le preceden en la ruta y el conjunto de agentes que le siguen. El cálculo del valor de Shapley realizado por este autor en los juegos de citas refleja este hecho. Para cada $a = (N, c, r)$ y cada patrocinador o, en el caso de nuestro ejemplo clientes, $i \in N$, si S_i es el conjunto conexo al que pertenece i , entonces su valor de Shapley en un juego de citas es el promedio de su contribución marginal a la coalición de patrocinadores o clientes que están en el mismo conjunto conexo que ella y la preceden en la ruta.

Efectivamente, en base a esto, podemos deducir que para cada agente solo importa su contribución marginal a estos dos conjuntos de agentes (los agentes que le preceden y los que lo siguen). Sin embargo, la fórmula original del valor de Shapley considera su contribución marginal a cualquier subconjunto de agentes.

Por último, a la hora de obtener el valor de Shapley, debemos de tener en cuenta que el valor de Shapley de cada patrocinador o cliente solo depende del coste de viajar

desde su lugar de origen hasta el nodo cero o “casa”. Una modificación en el coste que conlleva ir desde su lugar de origen hasta el nodo cero solo le afecta a dicho cliente y a los patrocinadores o clientes que están conectados con él. Además, un cambio en el coste del viaje entre dos clientes sólo afectará, por igual, a estos dos.

Siguiendo con el ejemplo expuesto anteriormente, tenemos un valor de Shapley (φ) del juego para cada $v \in \Gamma^N$ igual a:

$$Sh(v_a) = [22, 10.5, 3.5, 3]$$

Por otro lado, y en lo que respecta al Núcleo o Core del juego, en un juego de citas para que este no esté vacío tenemos dos condiciones. La primera, es que no necesitamos que r sea la ruta de menor coste. En segundo lugar, solo necesitamos que, dada una ruta r , para cada par de patrocinadores o clientes conectados, la suma de los costes de conexión al nodo cero o “casa” sea mayor que el coste de conexión de estos entre sí, es decir, formalmente, $c_i + c_j \geq c_{ij}$. Debemos de tener presente que, dichos costes, son los que representan el trayecto de ida y vuelta desde el nodo origen al nodo cero. Comprobemos dicha condición con nuestro ejemplo. Teniendo en cuenta que $c_i = c_1 = 24$, $c_j = c_2 = 12$ y $c_{ij} = c_{12} = 14$, podemos comprobar que se cumple dicha condición y que, por tanto, el núcleo o Core del juego no es vacío.

$$C(N, v_a) = [22, 10.5, 3.5, 3]$$

El tau-valor es otra de las principales soluciones de los juegos cooperativos. Como bien comentamos en la Definición 2.11, esta solución representa el mínimo derecho que recibe cada jugador más una cantidad añadida que depende de su pago de utopía. Así, en nuestro ejemplo, la cantidad mínima que recibirá cada jugador (en nuestro ejemplo, nos referimos a estos como clientes) es:

$$\tau - valor(N, v_a) = [21.875, 10.875, 3.625, 2.625]$$

2.3.3 El juego del viajante de comercio

El problema del viajante de comercio consiste en encontrar la ruta más barata para visitar a un número determinado de agentes (exactamente una vez), saliendo y regresando de su lugar de origen o nodo cero. Una de sus peculiaridades es que el viajante de comercio puede visitar a cualquier agente en el orden que el desee, siempre y cuando los agentes pertenezcan a la misma coalición y el coste de la ruta obtenida sea el mínimo. En lo que respecta a los juegos del viajante de comercio, el valor de la coalición es el coste del recorrido menos costoso entre los miembros de esa coalición.

Yengin (2012), en base a los estudios realizados por Potters et al. en 1992, define formalmente el problema del viajante de comercio mediante la tupla $e = (N, c, r)$, siendo N el conjunto de jugadores, c la matriz de costes de todos los arcos que conectan los vértices en $N \cup 0$ (siendo 0 la ciudad de partida o nodo origen) y r la ruta a realizar por el viajante de comercio.

Supondremos que c satisface la desigualdad triangular y que tenemos un grafo completo con los costes de todos los pares de elementos. Así, para el juego del viajante de comercio, cuando este se salta una ciudad o patrocinador, el viajante va directamente al siguiente patrocinador en S , si este patrocinador pertenece o existe en la ruta. Entonces, si la ruta r obtenida para la gran coalición N es la de menor coste y, las desigualdades triangulares se cumplen para todos los agentes, entonces el Núcleo del juego de comercio asociado a e no está vacío.

A diferencia de las dos versiones anteriores, en esta versión, el viajante no puede visitar a los agentes que forman una coalición más de una vez y tampoco puede regresar al nodo cero o a lo que denominamos anteriormente como “casa” en medio de la ruta sino únicamente al principio y final de esta, una peculiaridad que si presentaba el *juego del viajante con ruta fija y con citas*. Por otro lado, en esta versión, una de las ventajas que presenta el juego del viajante de comercio es que este no presenta una ruta fija, como si ocurre en el *juego del viajante con ruta fija* y en el *juego de citas*, y, por tanto, el

viajante puede obtener la ruta de menor coste entre todas las posibilidades sin tener una ruta fijada de antemano y correr el riesgo de que esta no sea la de menor coste.

Sin embargo, el juego del viajante de comercio presenta un inconveniente con respecto al juego del viajante con ruta fija y es que, para poder definir su función de costes v_c , se ha de resolver numerosos TSP para cada coalición $S \subseteq N$. Como bien comentamos anteriormente, el TSP es un NP-duro, esto implica que el tener que resolver un gran número de TSP para poder definir su función de costes hace que los problemas del viajante de comercio con un elevado número de agentes sean difíciles y costosos de resolver.

El juego del viajante de comercio plantea dos condiciones que han de cumplirse:

1. La suma de los pagos de cada agente debe coincidir con el coste total del viaje (eficiencia).
2. Ninguna coalición $S \subset N$, incluidas las unipersonales, debe pagar más del coste asociado a la ruta óptima entre las ciudades de S .

Ejemplo de un juego del viajante de comercio

Imaginemos que estamos ante un juego que se puede representar con la tupla $e = (N, c, r)$ teniendo en cuenta que $N = \{1, 2, 3\}$ y que los costes asociados a estos agentes y a este juego son iguales a $c_1 = c_{12} = c_{21} = 5$, $c_2 = c_3 = c_{23} = c_{32} = 3$ y que $c_{13} = c_{31} = 6$ con una ruta (r) óptima (de menor coste) igual a $r = (0, 1, 2, 3, 0)$. Entonces, si la ruta obtenida es la de menor coste y la matriz c cumple la desigualdad triangular, en base a lo comentado anteriormente por Potters et al (1992), el juego del viajante de comercio con ruta v_r tiene Núcleo no vacío. Veámoslo de una manera más práctica. Teniendo en cuenta que su función de costes es igual a:

$$v_r = [10, 6, 6, 13, 14, 9, 16]$$

Algunas de las principales soluciones de los juegos cooperativos son el *valor de Shapley* y el *tau-valor*. En el caso del juego del viajante de comercio estas principales soluciones son iguales a:

- El valor de Shapley es igual a $Sh(v_r) = (8.16, 3.67, 4.17)$.
- El tau-valor del juego v_r es igual a $\tau(v_r) = (8.2, 3.6, 4.2)$.

Por otro lado, otra de las soluciones que se pueden obtener en un juego cooperativo es el Nucleolo. El nucleolo del juego v_r tiene un valor igual a:

$$Nu(N, v_r) = (8.33, 3.33, 4.33)$$

El núcleo de un juego ha sido estudiado por muchos autores. Uno de esos estudios fue el que realizaron los autores Potters et al (1992). Estos demostraron que cualquier juego del viajante de comercio que tuviese un número de agentes menor o igual a tres y con una matriz de costes (c) que cumpla las condiciones (4) y (5), explicadas anteriormente en el *epígrafe 2.3.1*, tiene núcleo no vacío. Sin embargo, no todos los autores coinciden.

Los autores Derks y Kuipers (1997) y Sun y Karwan (2015), establecen en sus artículos que el núcleo de un juego del viajante de comercio puede estar vacío. Asimismo, los autores Potter et al. (1992) dieron, también, un ejemplo de un juego de 4 agentes con un núcleo vacío en el que los pesos del arco del gráfico no eran simétricos, es decir, el coste de viajar de i a j no era igual que el coste de viajar de j a i . Sin embargo, incluso si estos pesos fueran simétricos, un TSG (en inglés, Traveling Salesman Game) puede tener un núcleo vacío. Así, el autor Tamir (1989) dio un ejemplo de un TSG simétrico de 6 agentes para el cual el núcleo está vacío. Utilizando un resultado de Fonlupt y Naddef en 1992, este autor, también demostró que un TSG simétrico siempre tiene un núcleo no vacío cuando hay como máximo 4 jugadores. En Kuipers (1991) se demostró que también un TSG simétrico de 5 personas siempre tiene un núcleo no vacío.

Por último, obtendremos el Core-center del núcleo obtenido. Este juego presenta un valor del *Core – Center* igual a:

$$Core - Center (N, v_r) = (8.26, 3.48, 4.26)$$

y una representación gráfica:

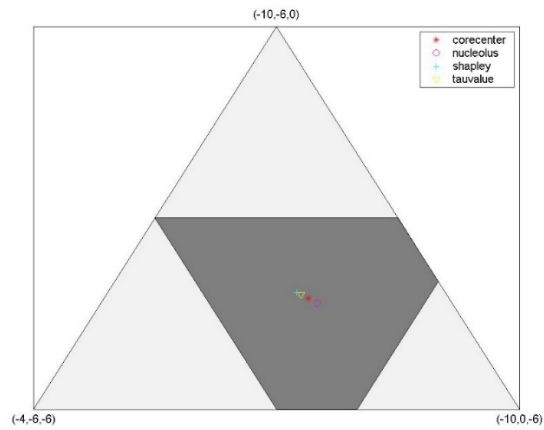


FIGURA 19. REPRESENTACIÓN DEL CORE DEL EJEMPLO DEL JUEGO DEL VIAJANTE DE COMERCIO

Capítulo 3

Caso práctico: Reparto de costes mediante Juegos Cooperativos: El problema del viajante de Castilla y León.

3.1. Introducción

A priori, el problema del viajante de comercio (con sus siglas en inglés, TSP) parece un problema de fácil resolución por la simplicidad de su enunciado. Sin embargo, esto queda lejos de la realidad. Así es que, el problema del viajante de comercio es considerado (dentro del campo de la optimización combinatoria) un problema NP-duro y, por tanto, es computacionalmente complejo a medida que el número de agentes aumenta. A parte de la dificultad de su resolución cuando el número de agentes o ciudades es elevado, la resolución del TSP no consiste solo en encontrar un recorrido de menor coste sino, también, en demostrar que no existe la posibilidad de encontrar un recorrido con un coste menor o más barato.

Así, según lo expuesto Applegate D., Bixby R.E., Chvátal V. y Cook W. J., en la década de 1950 los primeros trabajos sobre el problema del viajante se centraban en encontrar una solución completa del problema. A partir de mediados de la década de 1960, los investigadores comenzaron a investigar una versión más relajada del TSP donde solo se centraban en el recorrido de menor coste. Desde esa fecha hasta la actualidad, son muchos los problemas del viajante de comercio que han sido resueltos mediante los diferentes algoritmos (Véase *epígrafe 1.5*), así como diferentes versiones de este.

El problema del viajante de comercio se puede considerar como un problema con cierta dificultad para su aplicación a la vida real o a los problemas actuales, pero este no es el caso. Un ejemplo de ello es la problemática ocurrida en el año 2021 en Reino Unido denominada “la crisis de los camioneros”.

La crisis de los camioneros principalmente estaba motivada, junto a otros factores económicos externos, por el suceso ocurrido en el año 2020: la salida del Reino Unido

de la Unión Europea, conocida comúnmente como Brexit. Asimismo, muchos trabajadores y expertos del sector comentaban también que otro de los motivos de esta crisis fue la escasez de trabajadores en este sector, causada por las largas jornadas laborales en carretera llegando a estar hasta semanas para poder llevar los suministros a los distintos puntos de Reino Unido. Esta problemática se puede formular como un TSP donde un camionero debe llevar un número concreto de suministros a distintos puntos de su territorio mediante una ruta de menor coste o distancia. Una las posibles soluciones a esta problemática podrían encontrarse mediante los Juegos Cooperativos; planteándose la cooperación entre los distintos camioneros para reducir así las jornadas laborales de estos y los kilómetros en carretera.

En base a esto, en este capítulo, hemos querido introducir y resolver un ejemplo basado en datos reales en el que un comerciante quiere llevar su producto a diferentes provincias de la región de Castilla y León, adaptando el juego de este a las diferentes versiones de uno de los juegos más estudiados en el campo de las matemáticas: El juego del viajante de comercio. Por último, obtendremos las principales soluciones de los juegos cooperativos. Para ello, hemos utilizado la herramienta, TUGLabR propuesta por Núñez, Mirás Calvo, Quinteiro Sandomingo, y Sánchez Rodríguez en 2023.

3.2. El problema del viajante de Castilla y León.

Supongamos que un comerciante tiene que visitar y llevar su producto a siete provincias de un total de nueve que forman la región más extensa de España: Castilla y León. Dicho comerciante sale de su provincia de residencia, Ávila (a la que llamaremos nodo cero o provincia de partida) y desea recorrer el resto de las provincias minimizando la distancia o coste de la ruta. Para facilitar la comprensión del lector y la nomenclatura utilizada, vamos a asignar una abreviatura a cada provincia a la que denominaremos P_i donde P hace referencia a la abreviatura de "Provincia" e i al número de provincia asignado yendo este desde el nodo cero al nodo 6, es decir, $i = \{0, 1, \dots, 6\}$. Toda la nomenclatura que utilizaremos de aquí en adelante para este ejemplo, en lo que a nodos

CAPÍTULO 3. CASO PRÁCTICO: REPARTO DE COSTES MEDIANTE JUEGOS
COOPERATIVOS: EL PROBLEMA DEL VIAJANTE DE CASTILLA Y LEÓN

y para el juego de este se refiere, viene resumida en la *Tabla 4* que se muestra a continuación.

Por otro lado, el comerciante a la hora de realizar la ruta decide realizarla mediante carretera. Dicha ruta, la mediremos en términos de kilómetros, siendo el objetivo principal del comerciante minimizar estos. Para obtener los datos, hemos buscado los caminos entre las diferentes provincias que forman la ruta en la aplicación y herramienta gratuita “*Google Maps*”.

Provincia	Abreviatura
Ávila	P_0
Burgos	P_1
León	P_2
Palencia	P_3
Salamanca	P_4
Segovia	P_5
Soria	P_6

TABLA 4. NOMENCLATURA DEL PROBLEMA DEL VIAJANTE DE CASTILLA Y LEÓN

Y cuyo grafo se muestra en la *Figura 20*. En este caso, hemos elegido las opciones por las que nuestro comerciante iba por carreteras principales, ya que de normal suelen conllevar un menor tiempo y distancia, y no por carreteras secundarias o comarcales. Por ende, algunas de estas opciones podrían conllevar otros costes como, por ejemplo, peajes. Sin embargo, estos no se han tenido en cuenta a la hora de obtener los costes del juego de este ejemplo. Los costes de conexión entre las diferentes provincias de Castilla y León son los mostrados en la *Tabla 5*.

**CAPÍTULO 3. CASO PRÁCTICO: REPARTO DE COSTES MEDIANTE JUEGOS
COOPERATIVOS: EL PROBLEMA DEL VIAJANTE DE CASTILLA Y LEÓN**

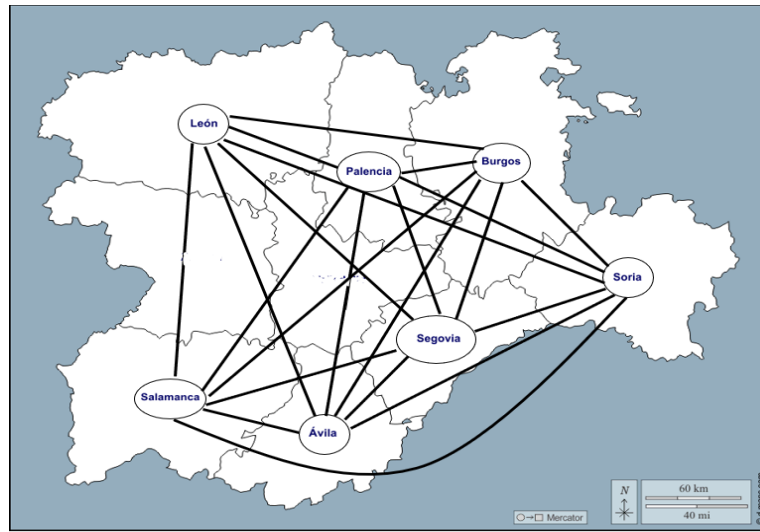


FIGURA 20. GRAFO DE LA RUTA DEL VIAJANTE DE CASTILLA Y LEÓN.

Desde/Hasta	P0	P1	P2	P3	P4	P5	P6
P0	0	263	263	184	107	65	256
P1	263	0	172	90	244	200	142
P2	263	172	0	133	206	294	311
P3	184	90	133	0	165	161	222
P4	107	244	206	165	0	171	324
P5	65	200	294	161	171	0	189
P6	256	142	311	222	324	189	0

TABLA 5. COSTES DEL PROBLEMA DEL VIAJANTE DE CASTILLA Y LEÓN.

El problema del viajante de comercio de Castilla y León tiene asociado un juego cuyo objetivo, como bien comentamos anteriormente, es minimizar los kilómetros o distancia recorrida. Dicho juego está compuesto por un número de agentes igual a seis ($N = \{1,2,3,4,5,6\}$) y por unos valores que corresponden al coste de ir desde el nodo cero a ese nodo en concreto constituyendo así la función característica del juego. Este juego de costes (N, v) se representa mediante un grafo no dirigido donde el nodo 0 representa el nodo de partida del viajante o comerciante siendo este también el último nodo o

provincia que forma la ruta, es decir, el viajante siempre parte de este nodo, recorre todos los nodos que forman la ruta, y regresa al mismo nodo inicial una vez terminada esta. Así, para obtener la función característica, hemos calculado la ruta de mínimo coste del juego. Por ejemplo, el valor del coste de la provincia uno ($v(1) \equiv c(1)$), es decir si la provincia uno no cooperara con el resto de las provincias, es igual a 526, el viajante recorrería 526 kilómetros yendo desde el nodo cero al nodo uno, ida y vuelta. El resto de los valores de los costes individuales los hemos obtenido de forma análoga al caso de la provincia uno.

Es de gran interés, comprobar si la matriz de costes del juego del viajante de Castilla y León cumple con la versión relajada, propuesta por los autores Derks and Kuipers (1997), de las desigualdades triangulares. Para ello, basándonos en la propuesta de estos autores y, siguiendo con la nomenclatura utilizada anteriormente, debe cumplirse que $c_{ij} \leq c_{i0} + c_{0j}$ para todo $i, j, k \in N \cup \{0\}$. Dicha comprobación se encuentra en la *Tabla 6* que exponemos a continuación. Como vemos, la columna uno de la *Tabla 6* hace referencia a la coalición a la que estamos realizando la comprobación de dicha desigualdad, la segunda columna corresponde con las provincias que forman dicha coalición, la tercera y cuarta columna hacen referencia a los costes a los que incurre el proveedor yendo desde la *Provincia i* a la *Provincia j* directamente (c_{ij}) y la suma de los costes de ir desde el nodo cero o *Provincia 0* (correspondiente esta con Ávila) a la *Provincias i* y de la *Provincias j* al nodo cero, respectivamente. Como vemos, se verifica esta versión relajada.

Sin embargo, si hacemos una comprobación, por ejemplo, con $c_{06} \leq c_{05} + c_{56}$ es igual a $c_{06} = 256 < 254$, vemos que no se verifica la desigualdad triangular propuesta con anterioridad y, por tanto, la matriz de costes no cumpliría la desigualdad de Minkowski, pero sí la versión de relajada de esta.

**CAPÍTULO 3. CASO PRÁCTICO: REPARTO DE COSTES MEDIANTE JUEGOS
COOPERATIVOS: EL PROBLEMA DEL VIAJANTE DE CASTILLA Y LEÓN**

$v(ij)$	Provincias	c_{ij}	$c_{i0} + c_{0j}$
1,2	Burgos a León	172	526
1,3	Burgos a Palencia	90	447
1,4	Burgos a Salamanca	244	370
1,5	Burgos a Segovia	200	328
1,6	Burgos a Soria	142	519
2,3	León a Palencia	133	447
2,4	León a Salamanca	206	370
2,5	León a Segovia	294	328
2,6	León a Soria	311	519
3,4	Palencia a Salamanca	165	291
3,5	Palencia a Segovia	161	249
3,6	Palencia a Soria	222	440
4,5	Salamanca a Segovia	171	172
4,6	Salamanca a Soria	324	363
5,6	Segovia a Soria	189	321

TABLA 6. DESIGUALDADES TRIANGULARES DE LA MATRIZ DE COSTES DEL PROBLEMA DEL VIAJANTE DE CASTILLA Y LEÓN.

Por un lado, los valores de los costes individuales que conforman el juego son:

Coalición	Valor
$v(1)$	526
$v(2)$	526
$v(3)$	368
$v(4)$	214
$v(5)$	130
$v(6)$	512

TABLA 7. COSTES INDIVIDUALES DEL PROBLEMA DEL VIAJANTE DE CASTILLA Y LEÓN.

Por otro lado, tenemos también las coaliciones formadas por dos agentes o provincias. El valor de estas coaliciones se obtiene escogiendo el camino hacia la provincia que tenga un menor coste. Así, por ejemplo, para la coalición de las provincias una y dos ($v(1,2)$), si el viajante sale de Ávila (su provincia de origen), tiene dos opciones, ir primero hacia la provincia uno o a la provincia dos; en este caso ambos caminos presentan la misma distancia, por tanto, es indiferente que vaya por uno o por otro, pero vamos a suponer que elige ir primero a la provincia uno con una distancia igual a 263 kilómetros. Una vez en la provincia uno, la distancia de ir desde esta provincia a la segunda provincia es igual a 172 kilómetros y, desde la segunda provincia al lugar de partida del viajante el coste o distancia es igual a 263. Por tanto, el viajante, en caso de hacer esta ruta, obtendría una distancia igual a 698 kilómetros.

Supongamos ahora que las provincias uno y tres deciden cooperar para minimizar la distancia de la ruta que va a realizar el viajante. En este caso, el viajante saldría del nodo cero y visitaría primero la provincia tres ya que el recorrido a esta es menor, desde la provincia tres iría a la provincia uno, recorriendo una distancia igual a 90 kilómetros y desde la provincia uno regresaría a la provincia de partida, obteniendo así un coste igual a 537 kilómetros. Como vemos, a la provincia uno le interesa más cooperar con la provincia tres que con la segunda debido a que obtendría una menor distancia o coste de la ruta. Operando del mismo modo, obtendríamos el valor de todas las coaliciones y las cuales están representadas en la *Tabla 8*.

Coalición	Valor
$v(1,2)$	698
$v(1,3)$	537
$v(1,4)$	614
$v(1,5)$	528
$v(1,6)$	661
$v(2,3)$	580
$v(2,4)$	576

**CAPÍTULO 3. CASO PRÁCTICO: REPARTO DE COSTES MEDIANTE JUEGOS
COOPERATIVOS: EL PROBLEMA DEL VIAJANTE DE CASTILLA Y LEÓN**

$v(2,5)$	622
$v(2,6)$	830
$v(3,4)$	456
$v(3,5)$	410
$v(3,6)$	662
$v(4,5)$	343
$v(4,6)$	687
$v(5,6)$	510

TABLA 8. COSTES DE LAS COALICIONES FORMADAS POR DOS PROVINCIAS DEL PROBLEMA DEL VIAJANTE DE CASTILLA Y LEÓN.

Imaginemos ahora, que el viajante decide que quiere visitar en su ruta tres provincias y quiere saber que ruta debe seguir para minimizar el coste de esta y, en consecuencia, el coste que tendría cada ruta (en el hipotético caso de que las provincias deciden cooperar). El comerciante puede conocer dicho valor mediante las coaliciones de tres agentes. Así, por ejemplo, si las provincias uno, dos y tres deciden cooperar, el viajante tendría que realizar la ruta que presenta el orden 0-3-1-2-0, debido a que realizando dicha ruta obtiene una menor distancia o coste (exactamente de 709) que variando el orden de esta. De manera análoga, realizaría el cálculo del resto de coaliciones, obteniendo así los valores expuestos en la *Tabla 9*. El código de salida del cálculo de estos valores con el servidor NEOS se encuentra reflejado en el *Anexo I*.

Coalición	Valor
$v(1,2,3)$	709
$v(1,2,4)$	748
$v(1,2,5)$	700
$v(1,2,6)$	833
$v(1,3,4)$	625

**CAPÍTULO 3. CASO PRÁCTICO: REPARTO DE COSTES MEDIANTE JUEGOS
COOPERATIVOS: EL PROBLEMA DEL VIAJANTE DE CASTILLA Y LEÓN**

$v(1,3,5)$	539
$v(1,3,6)$	672
$v(1,4,5)$	616
$v(1,4,6)$	749
$v(1,5,6)$	659
$v(2,3,4)$	630
$v(2,3,5)$	622
$v(2,3,6)$	874
$v(2,4,5)$	672
$v(2,4,6)$	880
$v(2,5,6)$	828
$v(3,4,5)$	498
$v(3,4,6)$	750
$v(3,5,6)$	660
$v(4,5,6)$	685

TABLA 9. COSTES DE LAS COALICIONES FORMADAS POR TRES PROVINCIAS DEL PROBLEMA DEL VIAJANTE DE CASTILLA Y LEÓN.

Ahora supongamos que el viajante quiere visitar, en la misma ruta, cuatro provincias. Para ello, quiere conocer si estas estuviesen dispuestas a cooperar entre ellas, teniendo en cuenta que estas cooperan si obtienen un menor coste realizando la ruta conjuntamente que por separado. Teniendo en cuenta esto, vamos a calcular las coaliciones formadas por cuatro provincias, realizando el cálculo de la misma forma que los casos anteriores, pero con una provincia más en la ruta. Vamos a calcular, por ejemplo, el coste o distancia de la coalición de las cuatro primeras provincias (Burgos, León, Palencia y Salamanca). El comerciante sale de su lugar de residencia, Ávila, y debe ir a una de estas provincias. Ateniéndose al criterio de obtener el menor coste, decide ir primero a la provincia número cuatro, desde dicha provincia, va hasta la provincia que hemos definido como P_3 , desde esta, la siguiente provincia que tiene una

**CAPÍTULO 3. CASO PRÁCTICO: REPARTO DE COSTES MEDIANTE JUEGOS
COOPERATIVOS: EL PROBLEMA DEL VIAJANTE DE CASTILLA Y LEÓN**

menor distancia es la provincia uno y, finalmente, desde la primera provincia va hasta la segunda y regresa a la provincia de partida o lugar de residencia. Por tanto, la ruta realizada y obtenida a través del servidor NEOS es igual a 0-4-2-1-3-0 con un coste o distancia igual a 759. Mediante este procedimiento, obtendríamos los valores de todas las coaliciones posibles de este juego. El código de salida del cálculo de dichos valores con el servidor NEOS se encuentra reflejado en el *Anexo II*.

Coalición	Valor
$v(1,2,3,4)$	759
$v(1,2,3,5)$	751
$v(1,2,3,6)$	884
$v(1,2,4,5)$	750
$v(1,2,4,6)$	883
$v(1,2,5,6)$	831
$v(1,3,4,5)$	627
$v(1,3,4,6)$	760
$v(1,3,5,6)$	670
$v(1,4,5,6)$	747
$v(2,3,4,5)$	672
$v(2,3,4,6)$	924
$v(2,3,5,6)$	872
$v(2,4,5,6)$	878
$v(3,4,5,6)$	748

TABLA 10. COSTES DE LAS COALICIONES FORMADAS POR CUATRO PROVINCIAS DEL PROBLEMA DEL VIAJANTE DE CASTILLA Y LEÓN.

Por último, imaginemos que las provincias deciden cooperar, pero no todas, solo cinco de ellas. Teniendo en cuenta esto, supongamos que las cinco primeras provincias deciden cooperar para minimizar la distancia o coste de la ruta, obteniendo así una ruta

**CAPÍTULO 3. CASO PRÁCTICO: REPARTO DE COSTES MEDIANTE JUEGOS
COOPERATIVOS: EL PROBLEMA DEL VIAJANTE DE CASTILLA Y LEÓN**

con un orden igual a 0-5-3-1-2-4-0 (calculando dicha ruta bajo el mismo criterio que las rutas de las coaliciones anteriores) con un coste o distancia igual a 801 kilómetros. Así, el resto de las coaliciones formadas por cinco provincias o agentes se obtienen siguiendo el mismo procedimiento que se expone en el *Anexo III* y cuyos valores se muestran en la *Tabla 11*.

Coalición	Valor
$v(1,2,3,4,5)$	801
$v(1,2,3,4,6)$	934
$v(1,2,3,5,6)$	882
$v(1,2,4,5,6)$	840
$v(1,3,4,5,6)$	758
$v(2,3,4,5,6)$	922

TABLA 11. COSTES DE LAS COALICIONES FORMADAS POR CINCO PROVINCIAS DEL PROBLEMA DEL VIAJANTE DE CASTILLA Y LEÓN.

En el caso de que, todas las provincias deciden cooperar, el valor de juego y el coste total de la ruta realizada de m por el comerciante es igual a:

$$v(N) = v(1,2,3,4,5,6) = 932 \text{ kilómetros}$$

Es decir, el proveedor, para llevar su producto a todas las provincias de la ruta, recorrerá un total de 932 kilómetros y realizará la ruta 0-5-6-1-3-2-4-0 siendo esta la ruta de menor coste. Para obtener la ruta de menor coste, hemos utilizado nuevamente el servidor NEOS y cuyo código de salida se muestra en el *Anexo IV*.

Así, la función característica de este juego, teniendo en cuenta que $S = \{1,2,3,4,5,6\}$, está formada por el valor de las coaliciones individuales, de las coaliciones formadas por dos agentes (provincias) o más y por el valor de la coalición total.

El viajante de comercio tiene varias posibilidades para realizar la ruta de mínimo coste obteniendo, en cada una de ellas, diferentes valores y planteamientos del juego, así como diferentes resultados de las principales soluciones de juegos cooperativos.

Es por ello por lo que, a continuación, vamos a aplicar este ejemplo a las diferentes versiones del juego del viajante explicadas en los anteriores epígrafes de este trabajo.

3.2.1. El juego del viajante de Castilla y León con ruta fija.

Supongamos que el viajante de Castilla y León decide realizar una ruta ya fijada de antemano con los clientes. Para ello, obtiene un juego de costes al que denominaremos por (N, v_r) sujeto al valor del juego a la ruta elegida por el viajante (v_r). Los costes de dicho juego son los expuestos anteriormente en el *epígrafe* 3.2 pero con la singularidad de que, en este caso, el coste total del juego cambia debido a que la ruta ya está prefijada; con la posible consecuencia de que esta puede no ser la de menor coste. En este caso, el viajante escoge la ruta $r = (0 - 6 - 5 - 1 - 3 - 2 - 4 - 0)$ con un coste total de 1.181 kilómetros, es decir, $v_r(N) = 1.181 \text{ km}$.

Para el cálculo del valor de la coalición total ($v_r(N)$) hemos utilizado los valores de las coaliciones intermedias reflejados en las *Tablas* 12, 13 y 14 que forman la función característica de este juego, basándonos en la ruta fijada de antemano para el cálculo de estas. Así, para el cálculo de, por ejemplo, la coalición $v(1,3,6)$ hemos tenido en cuenta el orden de la ruta fijada; el proveedor va desde el nodo cero a la provincia seis, luego desde está a la provincia uno y por último a la provincia tres. Asimismo, el valor de las coaliciones individuales y la formada por dos agentes o, en nuestro caso, provincias, no varía respecto al expuesto en la *Tabla* 7 y 8 del *epígrafe* 3.2.

**CAPÍTULO 3. CASO PRÁCTICO: REPARTO DE COSTES MEDIANTE JUEGOS
COOPERATIVOS: EL PROBLEMA DEL VIAJANTE DE CASTILLA Y LEÓN**

Coalición	Valor
$v(1,2,3)$	749
$v(1,2,4)$	748
$v(1,2,5)$	700
$v(1,2,6)$	833
$v(1,3,4)$	625
$v(1,3,5)$	539
$v(1,3,6)$	672
$v(1,4,5)$	616
$v(1,4,6)$	749
$v(1,5,6)$	908
$v(2,3,4)$	630
$v(2,3,5)$	622
$v(2,3,6)$	874
$v(2,4,5)$	672
$v(2,4,6)$	880
$v(2,5,6)$	1002
$v(3,4,5)$	498
$v(3,4,6)$	750
$v(3,5,6)$	790
$v(4,5,6)$	723

TABLA 12 . COSTES DE LAS COALICIONES FORMADAS POR TRES PROVINCIAS DEL PROBLEMA DEL VIAJANTE DE CASTILLA Y LEÓN CON RUTA FIJA.

Coalición	Valor
$v(1,2,3,4)$	799
$v(1,2,3,5)$	751
$v(1,2,3,6)$	884
$v(1,2,4,5)$	750

**CAPÍTULO 3. CASO PRÁCTICO: REPARTO DE COSTES MEDIANTE JUEGOS
COOPERATIVOS: EL PROBLEMA DEL VIAJANTE DE CASTILLA Y LEÓN**

$v(1,2,4,6)$	883
$v(1,2,5,6)$	1080
$v(1,3,4,5)$	627
$v(1,3,4,6)$	760
$v(1,3,5,6)$	919
$v(1,4,5,6)$	996
$v(2,3,4,5)$	672
$v(2,3,4,6)$	924
$v(2,3,5,6)$	1002
$v(2,4,5,6)$	1052
$v(3,4,5,6)$	878

TABLA 13. COSTES DE LAS COALICIONES FORMADAS POR CUATRO PROVINCIAS DEL PROBLEMA DEL VIAJANTE DE CASTILLA Y LEÓN CON RUTA FIJA

Coalición	Valor
$v(1,2,3,4,5)$	801
$v(1,2,3,4,6)$	934
$v(1,2,3,5,6)$	1131
$v(1,2,4,5,6)$	1130
$v(1,3,4,5,6)$	1007
$v(2,3,4,5,6)$	1052

TABLA 14. COSTES DE LAS COALICIONES FORMADAS POR CINCO PROVINCIAS DEL PROBLEMA DEL VIAJANTE DE CASTILLA Y LEÓN CON RUTA FIJA.

Así, la función característica de este juego es la formada por los valores de las *Tablas 7, 8, 12, 13 y 14* expuestos con anterioridad.

El viajante de Castilla y León quiere saber cuál será el vector de pagos de cada cliente, es decir, cuanto debe pagarle cada uno por realizar el servicio. Para ello, decide obtener

una de las principales soluciones de los juegos cooperativos nombrada anteriormente en este trabajo: El valor de Shapley. Así, sin más dilación, el valor de Shapley de este juego⁷ es igual a:

$$Sh(N, v_r) = (209.116, 267.95, 117.616, 99.15, 133.866, 353.3)$$

Dicho juego (N, v_r) presenta un Nucleolo igual a:

$$Nu(N, v_r) = (118.5, 270.00, 147.00, 146.00, 130.00, 369.5)$$

Y un Prenucleolo igual a:

$$PreNu(N, v_r) = (166.50, 240.75, 117.75, 116.75, 188.5, 350.75)$$

Como vemos, las provincias que deben realizar una mayor contribución al comerciante son la provincia seis y dos, Soria y León, respectivamente, que, como era de esperar, son las que se encuentran a mayor distancia de la provincia de origen del viajante de Castilla y León: Ávila.

Por otro lado, es de gran interés conocer el Núcleo o core del juego, siendo este el conjunto de pagos que ofrece a cada coalición y teniendo que ser al menos igual que el que estas pueden conseguir por sí mismas para fomentar así la cooperación y no la búsqueda del beneficio propio. En este caso, como la ruta escogida por el viajante no es óptima (existe otra ruta de menor coste, exactamente de 932 km) el Core de este juego será igual al vacío y, por tanto, el cliente va a preferir no cooperar al obtener un conjunto de pagos menor (o, en este ejemplo, mayores costes) que si realiza la búsqueda por cuenta propia.

Ahora, imaginemos que estamos ante la ruta 0-5-6-1-3-2-4-0, a la que denominaremos como r^* . Para esta ruta, hemos obtenido los siguientes valores de las coaliciones intermedias:

⁷ El valor de Shapley de este juego se ha obtenido mediante la siguiente herramienta: [Valor de Shapley](#)

**CAPÍTULO 3. CASO PRÁCTICO: REPARTO DE COSTES MEDIANTE JUEGOS
COOPERATIVOS: EL PROBLEMA DEL VIAJANTE DE CASTILLA Y LEÓN**

Coalición	Valor
$v(1,2,3)$	749
$v(1,2,4)$	748
$v(1,2,5)$	700
$v(1,2,6)$	833
$v(1,3,4)$	625
$v(1,3,5)$	539
$v(1,3,6)$	672
$v(1,4,5)$	616
$v(1,4,6)$	749
$v(1,5,6)$	659
$v(2,3,4)$	630
$v(2,3,5)$	622
$v(2,3,6)$	874
$v(2,4,5)$	672
$v(2,4,6)$	880
$v(2,5,6)$	828
$v(3,4,5)$	498
$v(3,4,6)$	750
$v(3,5,6)$	660
$v(4,5,6)$	685

TABLA 15. COSTES DE LAS COALICIONES FORMADAS POR TRES PROVINCIAS DEL PROBLEMA DEL VIAJANTE DE CASTILLA Y LEÓN CON RUTA FIJA Y RUTA ÓPTIMA.

Coalición	Valor
$v(1,2,3,4)$	799
$v(1,2,3,5)$	751
$v(1,2,3,6)$	884
$v(1,2,4,5)$	750

CAPÍTULO 3. CASO PRÁCTICO: REPARTO DE COSTES MEDIANTE JUEGOS
COOPERATIVOS: EL PROBLEMA DEL VIAJANTE DE CASTILLA Y LEÓN

$v(1,2,4,6)$	883
$v(1,2,5,6)$	831
$v(1,3,4,5)$	627
$v(1,3,4,6)$	760
$v(1,3,5,6)$	670
$v(1,4,5,6)$	747
$v(2,3,4,5)$	672
$v(2,3,4,6)$	924
$v(2,3,5,6)$	872
$v(2,4,5,6)$	878
$v(3,4,5,6)$	748

TABLA 16. COSTES DE LAS COALICIONES FORMADAS POR CUATRO PROVINCIAS DEL PROBLEMA DEL VIAJANTE DE CASTILLA Y LEÓN CON RUTA FIJA Y RUTA ÓPTIMA

Coalición	Valor
$v(1,2,3,4,5)$	801
$v(1,2,3,4,6)$	934
$v(1,2,3,5,6)$	882
$v(1,2,4,5,6)$	881
$v(1,3,4,5,6)$	758
$v(2,3,4,5,6)$	922

TABLA 17. COSTES DE LAS COALICIONES FORMADAS POR CINCO PROVINCIAS DEL PROBLEMA DEL VIAJANTE DE CASTILLA Y LEÓN CON RUTA FIJA Y RUTA ÓPTIMA

Como podemos observar, las únicas coaliciones que varían respecto a los expuestos en las *Tablas 12, 13 y 14* son en las que las dos Provincias (P_5 y P_6) coinciden (por ejemplo, $v(1,5,6)$). Esto es debido a que la ruta óptima ($r = (0 - 5 - 6 - 1 - 3 - 2 - 4 - 0)$) difiere de la ruta elegida con anterioridad ($r = (0 - 6 - 5 - 1 - 3 - 2 - 4 - 0)$)

únicamente en el orden de visita de estas dos ciudades. Asimismo, el valor de la coalición total es igual a $v(N) = 932 \text{ km}$.

En base a esto, vamos a obtener de nuevo las principales soluciones de los juegos cooperativos. Así, el valor de Shapley del juego (N, v_{r^*}) es igual a:

$$Sh(N, v_{r^*}) = (167.95, 262.783, 116.116, 98.516, 33.60, 253.033)$$

Como podemos observar, si comparamos el valor de Shapley de este juego con ruta óptima con el obtenido anteriormente con la ruta r *no óptima*, vemos que las contribuciones varían notablemente para algunas Provincias como, por ejemplo, Segovia (P_5).

Ahora, entonces, el nucleolo del juego del viajante con ruta fija con la ruta óptima es igual a:

$$Nu(N, v_{r^*}) = (111.333, 257.666, 134.666, 132.00, 64, 232.333)$$

Por otro lado, el Prenucleolo del juego (N, v_{r^*}) es igual a:

$$Pre - Nu(N, v_{r^*}) = (111.333, 257.666, 134.666, 132, 64, 232.333)$$

que, como vemos, $Nu(N, v_{r^*}) = Pre - Nu(N, v_{r^*})$.

Por último, y no menos importante, vamos a obtener el Core-Center del juego:

$$Core - Center(N, v_{r^*}) = (128.373, 259.804, 130.390, 114.703, 52.336, 246.391)$$

3.2.2. El juego del viajante de Castilla y León con ruta fija y con citas.

En este caso, el viajante de Castilla y León decide realizar su ruta estableciendo un calendario previo con sus clientes. Para ello, debe concretar ciertas citas con cada uno de ellos. Dichas citas, determinan el orden mediante el cual el viajante realizará su ruta

no pudiendo modificar el calendario bajo ningún precepto, ni si quiera si el cliente cancela su cita, teniendo que, en este último caso, seguir con su ruta inicial.

Así, en base al problema definido con anterioridad en el *epígrafe* 3.2, estamos ante un juego que está formado por $N = \{1,2,3,4,5,6\}$ agentes o clientes y una ruta $r = (0,5,6,1,3,2,4,0)$ con un valor de la coalición total $v(N) = 932 \text{ km}$. El juego del viajante de comercio con ruta fija y con citas está compuesto por los costes individuales (véase *Tabla 7 del epígrafe* 3.2) y por los costes de las coaliciones intermedias (véase desde la *Tabla 18* hasta la *Tabla 21*). Dichos costes, forman la función característica de este juego. Sin embargo, el proveedor, por motivos externos puede decidir realizar la ruta restringiéndola a ciertas provincias o a un conjunto de agentes a los que denominamos S . Así, para obtener, por ejemplo, el valor de la coalición restringida a los agentes $S = \{1,2,5\}$ a la que denominamos formalmente como $v(1,2,5)$ el proveedor deberá seguir el orden de la ruta " r " pero eliminando de S los agentes que no forman parte de esta coalición. Por tanto, la ruta r será restringida a la coalición $S = \{1,2,5\}$ y la denominaremos formalmente como $r_s = (0,5,0,1,0,2,0)$ con un valor igual a:

$$v(1,2,5) = 2 * c_5 + 2 * c_1 + 2 * c_2 = 2 * 65 + 2 * 263 + 2 * 263 = 1182 \text{ km}$$

Como vemos, el valor de esta coalición es la suma del coste o distancia (expuestos en la *Tabla 5*) de ir desde el nodo cero o nodo de partida del proveedor a Segovia (P_5), de esta provincia regresa al nodo de partida (Ávila), desde Ávila va a Burgos (P_1), de Burgos regresa a Ávila (P_0) y desde esta provincia viaja hasta León (P_2) y regresa a casa. El resto de los valores de las coaliciones intermedias han sido obtenidos de forma análoga.

Coalición	Valor
$v(1,2)$	1052
$v(1,3)$	537
$v(1,4)$	740
$v(1,5)$	656
$v(1,6)$	661

**CAPÍTULO 3. CASO PRÁCTICO: REPARTO DE COSTES MEDIANTE JUEGOS
COOPERATIVOS: EL PROBLEMA DEL VIAJANTE DE CASTILLA Y LEÓN**

$v(2,3)$	580
$v(2,4)$	576
$v(2,5)$	656
$v(2,6)$	1038
$v(3,4)$	582
$v(3,5)$	498
$v(3,6)$	880
$v(4,5)$	344
$v(4,6)$	726
$v(5,6)$	510

TABLA 18. COSTES DE LAS COALICIONES FORMADAS POR DOS PROVINCIAS DEL PROBLEMA DEL VIAJANTE DE CASTILLA Y LEÓN CON RUTA FIJA Y CON CITAS.

Coalición	Valor
$v(1,2,3)$	749
$v(1,2,4)$	1102
$v(1,2,5)$	1182
$v(1,2,6)$	1187
$v(1,3,4)$	751
$v(1,3,5)$	667
$v(1,3,6)$	672
$v(1,4,5)$	870
$v(1,4,6)$	875
$v(1,5,6)$	659
$v(2,3,4)$	630
$v(2,3,5)$	710
$v(2,3,6)$	1092
$v(2,4,5)$	706

**CAPÍTULO 3. CASO PRÁCTICO: REPARTO DE COSTES MEDIANTE JUEGOS
COOPERATIVOS: EL PROBLEMA DEL VIAJANTE DE CASTILLA Y LEÓN**

$v(2,4,6)$	1088
$v(2,5,6)$	1036
$v(3,4,5)$	712
$v(3,4,6)$	1094
$v(3,5,6)$	878
$v(4,5,6)$	724

TABLA 19. COSTES DE LAS COALICIONES FORMADAS POR TRES PROVINCIAS DEL PROBLEMA DEL VIAJANTE DE CASTILLA Y LEÓN CON RUTA FIJA Y CON CITAS

Coalición	Valor
$v(1,2,3,4)$	799
$v(1,2,3,5)$	879
$v(1,2,3,6)$	884
$v(1,2,4,5)$	1232
$v(1,2,4,6)$	1237
$v(1,2,5,6)$	1185
$v(1,3,4,5)$	881
$v(1,3,4,6)$	886
$v(1,3,5,6)$	670
$v(1,4,5,6)$	873
$v(2,3,4,5)$	760
$v(2,3,4,6)$	1142
$v(2,3,5,6)$	1090
$v(2,4,5,6)$	1086
$v(3,4,5,6)$	1092

TABLA 20. COSTES DE LAS COALICIONES FORMADAS POR CUATRO PROVINCIAS DEL PROBLEMA DEL VIAJANTE DE CASTILLA Y LEÓN CON RUTA FIJA Y CON CITAS.

**CAPÍTULO 3. CASO PRÁCTICO: REPARTO DE COSTES MEDIANTE JUEGOS
COOPERATIVOS: EL PROBLEMA DEL VIAJANTE DE CASTILLA Y LEÓN**

Coalición	Valor
$v(1,2,3,4,5)$	929
$v(1,2,3,4,6)$	934
$v(1,2,3,5,6)$	882
$v(1,2,4,5,6)$	1235
$v(1,3,4,5,6)$	884
$v(2,3,4,5,6)$	1140

TABLA 21. COSTES DE LAS COALICIONES FORMADAS POR CINCO PROVINCIAS DEL PROBLEMA DEL VIAJANTE DE CASTILLA Y LEÓN CON RUTA FIJA Y CON CITAS.

Una vez obtenidos los valores tanto de los costes individuales como de todas las coaliciones intermedias, obtenemos, mediante estos valores, la función característica de este juego.

Por otro lado, en esta versión del viajante de Castilla y León, resulta interesante también obtener algunas soluciones principales de los juegos cooperativos. Obtendremos el nucleolo, el valor de Shapley y el Core-Center del juego del viajante con ruta fija y con citas.

Teniendo en cuenta los valores de los costes individuales de los agentes y de las coaliciones intermedias, el valor de Shapley es igual a:

$$Sh(N, v_r) = (159, 287, 32.5, 132, 64, 257.5)$$

Asimismo, el nucleolo del juego del viajante con ruta fija y con citas es igual a:

$$Nu(N, v_r) = (159, 287, 32.5, 132, 64, 257.5)$$

Como vemos, el valor de Shapley está dentro del core del juego. Por último, el Core-Center del juego del viajante de Castilla y León con ruta fija y con citas es igual a:

$$Core - Center (N, v_r) = (159, 287, 32.5, 132, 64, 257.5)$$

Como podemos observar, en este caso, los valores obtenidos del valor de Shapley, el núcleo y el Core-Center del juego son iguales, es decir:

$$Sh(N, v_r) = Nu(N, v_r) = Core - Center (N, v_r) = (159, 287, 32.5, 132, 64, 257.5)$$

Esta peculiaridad ha sido nombrada por varios autores, entre ellos, los autores Sánchez y González (2014). Estos, en su artículo "Understanding the coincidence of allocation rules: symmetry and orthogonality in TU-games" muestran la existencia de una regla de asignación que satisface la eficiencia, la covarianza de traslación y la simetría inversa de los juegos PS (los juegos PS son una clase de juegos donde el Prenucleolo es igual al valor de Shapley) y, que esta implica que el valor de Shapley, el Prenucleolo, el núcleo del juego y el Core-Center de este sean iguales.

3.2.3. El juego del viajante de comercio de Castilla y León.

Por último, supongamos que el viajante de comercio se encuentra ante la tesitura de tener que realizar su ruta mediante un recorrido en el cual no puede volver a su nodo origen o "casa". En base a lo comentado en anteriores epígrafes, podemos comprobar que se trata de un juego del viajante de comercio.

Esta versión del problema del viajante tiene la peculiaridad, respecto a las otras dos versiones ya mencionadas, que la ruta obtenida no es fija y, por tanto, cabe la posibilidad de estar ante una ruta óptima de menor coste. Teniéndolo en cuenta, podemos obtener el juego del viajante de comercio de Castilla y León.

Imaginemos que estamos ante un juego que se puede representar con la tupla $e = (N, c, r)$ teniendo en cuenta que $N = \{1, 2, 3, 4, 5, 6\}$ y que los costes asociados a estos agentes y a este juego son iguales a los expuestos en las tablas del *epígrafe* 3.2 con una ruta (r^*) óptima (de menor coste) igual a $r^* = (0, 5, 6, 1, 3, 2, 4, 0)$. Entonces, resulta

interesante obtener algunas de las principales soluciones de los juegos cooperativos nombradas anteriormente del juego v_{r^*} . En base a estas líneas el valor de Shapley es:

$$Sh(N, v_{r^*}) = (165.25, 260.083, 121.616, 97.15, 34.233, 253.66)$$

Por otro lado, hemos obtenido el Core-Center del juego:

$$Core - Center(N, v_{r^*}) = (114.637, 248.669, 159.851, 111.637, 52.662, 244.541)$$

Por último, el Prenucleolo y el Nucleolo del juego (N, v_{r^*}) son:

$$Pre - Nu(N, v_{r^*}) = (106, 247, 165, 123, 64, 227)$$

$$Nu(N, v_{r^*}) = (106, 247, 165, 123, 64, 227)$$

Como vemos, el Prenucleolo y el Nucleolo coinciden. Así, cuando el Prenucleolo es una imputación, entonces $Pre - Nu(N, v_{r^*}) = Nu((N, v_{r^*}))$.

3.3. Conclusiones.

En este trabajo se ha llevado a cabo un estudio del problema del viajante de comercio y del reparto de costes mediante los juegos cooperativos. A través de una revisión de la literatura existente, en la primera parte del trabajo se ha analizado la información disponible acerca de este tema y se han descrito los diferentes tipos de TSP, centrándonos en el TSP simétrico, así como en su formulación.

Por otro lado, para concluir el *primer capítulo*, se han comentado los diferentes métodos de resolución del problema del viajante.

El *capítulo dos* se ha centrado en el reparto de los costes mediante los juegos cooperativos, haciendo una aplicación de estos al problema que aquí nos concierne: el problema del viajante de comercio. Esta aplicación se ha ampliado a los diferentes juegos de las versiones del TSP nombradas en anteriores epígrafes.

Por último, en el *capítulo tres*, hemos expuesto un caso práctico del reparto de costes mediante juegos cooperativos: el problema del viajante de Castilla y León, y una aplicación de este a las diferentes versiones del TSP, obteniendo diferentes resultados en las distintas versiones del juego de este.

Ante estos resultados, se observa la importancia que tiene la elección de la ruta en las diferentes versiones de este problema. Un ejemplo de ello son los resultados obtenidos en una de las principales soluciones de los juegos cooperativos: el valor de Shapley. Este valor, como bien comentamos a lo largo del documento, refleja el vector de pagos de cada agente, es decir, cuanto debe pagarle cada uno por realizar el servicio. Así, para comparar las soluciones obtenidas, se podrían utilizar los axiomas que verifican dichas soluciones y que, dependiendo de si estamos ante el problema con ruta fija, con ruta fija y con citas o si la ruta es óptima o no, se podría aplicar el modelo correspondiente para el reparto de costes que deberían costear las correspondientes ciudades o, en nuestro ejemplo, Provincias.

Así, tras el análisis de los diferentes tipos de TSP y los resultados obtenidos, creemos que el juego del viajante con ruta fija y con citas se adapta más a la actualidad. Esto es debido a que, hoy en día, en la mayor parte de los casos, el viajante o, como reflejamos en nuestro ejemplo, el proveedor, concreta unas citas con los clientes antes de realizar la ruta, adaptándose a las necesidades de cada cliente, pudiendo ser dichas citas canceladas y, teniendo en cuenta que este puede regresar a su nodo de partida o nodo “casa”. Sin embargo, debemos tener presente la importancia que tiene la elección de la ruta o el orden en el que los clientes serán visitados y, por tanto, la importancia de la aplicación de los juegos cooperativos en este tipo de problemas. Es por ello que una posible línea de trabajo futura podría ser una profundización en el tema de la versión del juego del viajante con ruta fija y con citas, así como la aplicación de los juegos cooperativos en este.

Bibliografía

- Applegate, D., Bixby, R., Chvatal, V., & Cook, W. (2001). *TSP cuts which do not conform to the template paradigm*, 2241. Springer. [TSP cuts which do not conform to the template paradigm | Springer](#)
- Applegate, D., Cook, W., & Rohe, A. (2003). Chained Lin-Kernighan for Large Traveling Salesman Problems. *Inform's Journal on Computing*, 15 (1), 82-92. [Chained Lin-Kernighan for Large Traveling Salesman Problems | Inform's Journal on Computing](#)
- Applegate D., Bixby R.E., Chvátal V. y Cook W. J. *Concorde TSP for Solver*. [Concorde Windows Graphical User Interface](#)
- Chun, Y., Park, N., & Yengin, D. (2016). Coincidence of cooperative game theoretic solutions in the appointment problem. *International Journal of Game Theory*, 45, 699-708. [Coincidence of cooperative game theoretic solutions in the appointment problem | International Journal of Game Theory](#)
- Dantzig, G., Fulkerson, D., & Johnson, S. (1954). Solution of a large-scale traveling salesman problem. *Journal of the Operations Research Society of America*, 2(4), 393-410. [Solution of a large-scale traveling salesman problem | Journal of the Operations Research Society of America](#)
- Davendra, D. (2010). *Traveling Salesman Problem. Theory and applications*. InTechOpen. [Traveling Salesman Problem. Theory and applications | InTechOpen](#)
- Derks, J., & Kuipers, J. (1997). On the Core of Routing Games. *International Journal of Game Theory*, 26(2), 193-205. [On the Core of Routing Games | International Journal of Game Theory](#)
- Edmonds, J. (1965). Paths trees and flowers. *Canadian Journal of Mathematics*, 17, 449-467. [Paths trees and flowers | Canadian Journal of Mathematics](#)
- Fishburn, P., & Pollak, H. (1983). Fixed-Route Cost Allocation. *American Mathematical Monthly*, 90, 366-378. [Fixed-Route Cost Allocation | American Mathematical Monthly](#)
- Fonlupt, J., & Naddef, D. (1992). The traveling salesman problem in graphs with some excluded minors. *Mathematical Programming*, 53, 147-172. [The traveling salesman problem in graphs with some excluded minors | Mathematical Programming](#)
- González Díaz, J., & Sánchez Rodríguez, E. (2007). A natural selection from the core of a TU game: the core-center. *International Journal of Game Theory*, 36, 27-46. [A natural selection from the core of a TU game: the core-center | International Journal of Game Theory](#)

González Díaz, J., & Sánchez Rodríguez, E. (2014). Understanding the coincidence of allocation rules: symmetry and orthogonality in TU-games. *International Journal of Game Theory*, 43, 821-843. [Understanding the coincidence of allocation rules: symmetry and orthogonality in TU-games | International Journal of Game Theory](#)

Held, M., & Karp, R. (1962). A dynamic programming approach to sequencing problems. *Journal of the Society of Industrial and Applied Mathematics*, 10(1), 196-210. [A dynamic programming approach to sequencing problems | Journal of the Society of Industrial and Applied Mathematics](#)

Infantes Durán, M. (2018). *El problema del viajante (TSP)*. Sevilla: Universidad de Sevilla. Departamento de Estadística e Investigación Operativa. [El problema del viajante \(TSP\) | Universidad de Sevilla](#)

Karp, R.M. (1972). Reducibility among Combinatorial Problems. In: Miller, R.E., Thatcher, J.W., Bohlinger, J.D. (eds) *Complexity of Computer Computations*. The IBM Research Symposia Series. Springer, Boston, MA. [Reducibility Among Combinatorial Problems](#)

Kuipers, J. (1993). A note on the 5-person traveling salesman game. *Zeitschrift für Operations Research*, 38, 131-139. [A note on the 5-person traveling salesman game | Zeitschrift für Operations Research](#)

Kuipers, J., Solymosi, T., & Aarts, H. (2000). Computing the nucleolus of some combinatorially-structured games. *Mathematical Programming*, 88, 541-563. [Computing the nucleolus of some combinatorially-structured games | Mathematical Programming](#)

Lawler, E., Lenstra J.K., Rinnooy Kan A. & Shmoys D. (1985). *The traveling salesman problem*. John Wiley and Sons [The traveling salesman problem | John Wiley and Sons](#)

Martin, O., Otto, S., & Felten, E. (1991). Large-step Markov chains for the traveling salesman problem. *Complex System*, 5, pp. 299-326. [Large-step Markov chains for the traveling salesman problem | Complex System](#)

- Martin, O., Otto, S., & Felten, E. (1992). Large-step Markov chains for the TSP incorporating local search heuristics. *Operations Research Letter*, 11(4), 219-224. [Large-step Markov chains for the TSP incorporating local search heuristics | Operations Research Letter](#)
- Mirás Calvo, M.A., Sánchez Rodríguez, E. & Grande Cougil, R.P. (2010). TUGlabWeb. [TUGlabWeb](#)
- Mulder, S., & Wunsch, D. (2003). Million city traveling salesman problem solution by divide and conquer clustering with adaptive resonance neural networks. *Neural Networks: the official journal of the International Neural Network Society*, 16, 827-832. [Million city traveling salesman problem solution by divide and conquer clustering with adaptive resonance neural networks | Neural Networks: the official journal of the International Neural Network Society](#)
- Neumann, J. V. (1928). Zur Theorie der Gesellschaftsspiele. *Mathematische Annalen*, 100, 295-320. [Zur Theorie der Gesellschaftsspiele | Mathematische Annalen](#)
- Núñez Lugilde, I., Mirás Calvo, M.A., Quinteiro Sandomingo, C. and Sánchez Rodríguez, E. (2023). TUGLabR. Working software
- Papadimitriou, C. H., & Steiglitz, K. (1981). *Combinatorial Optimization: Algorithms and Complexity*. General Publishing Company. [Combinatorial Optimization: Algorithms and Complexity | General Publishing Company](#)
- Peleg, B., Shapley, L., & Maschler, M. (1979). Geometric Properties of the Kernel, Nucleolus and Related Solution Concepts. *Mathematics of Operations Research*, 4(4), 303-338. [Geometric Properties of the Kernel, Nucleolus and Related Solution Concepts | Mathematics of Operations Research](#)
- Potters, J., Curiel, I., & Tijs, S. (1992). Traveling Salesman Games. *Mathematical Programming*, 53, 199-211. [Traveling Salesman Games | Mathematical Programming](#)
- Rego, C., Gamboa, D., Glover, F., & Osterman, C. (2011). Traveling Salesman problem heuristics: leading methods, implementations and latest advances. *European Journal of Operational Research*, 211, 427-441. [Traveling Salesman problem heuristics: leading methods, implementations and latest advances | European Journal of Operational Research](#)

Sánchez Rodríguez, E., & Vidal Puga, J. (2014). Juegos coalicionales. Universidad de Vigo.

Schulz, A., & Uhan, N. (2013). Approximating the least core value and least core of cooperative games with supermodular costs. *Discrete Optimization*, 10, 163-180. [Approximating the least core value and least core of cooperative games with supermodular costs | Discrete Optimization](#)

Servidor NEOS. [Servidor NEOS](#)

Sevilla Arias, A. (2016, septiembre 30). *Teoría de juegos*. Economipedia. [Teoría de juegos | Economipedia](#)

Shapley, L., & Shubik, M. (1954). A Method for Evaluating the Distribution of Power in a Committee System. *The American Political Science Review*, 48, 787–792. Retrieved from [A Method for Evaluating the Distribution of Power in a Committee System | The American Political Science Review](#)

Solymosi, T., Aarts, H., & Driessen, T. (1998). On computing the nucleolus of a balanced connection game. *Mathematics of Operations Research*, 23, 983-1009. Retrieved from [On computing the nucleolus of a balanced connection game | Mathematics of Operations Research](#)

Sun, L., & Karwan, M. (2015). On the core of traveling salesman games. *Operations Research Letters*, 43, 365-369. [On the core of traveling salesman games | Operations Research Letters](#).

Tamir, A. (1989). On the core of a traveling salesman cost allocation game. *Operations Research Letters*, 8, 31-34. [On the core of a traveling salesman cost allocation game | Operations Research Letter](#)

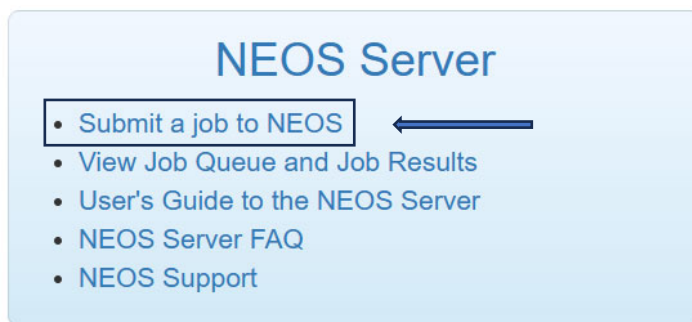
Von Neumann, J., & Morgenstern, O. (1944). *Theory of Games and Economic Behavior*. Princeton University Press.

Yengin, D. (2012). Characterizing the Shapley value in Fixed-route traveling salesman problems with appointments. *International Journal of Game Theory*, 41, 271-299. [Characterizing the Shapley value in fixed-route traveling salesman problems with appointments | SpringerLink](#)

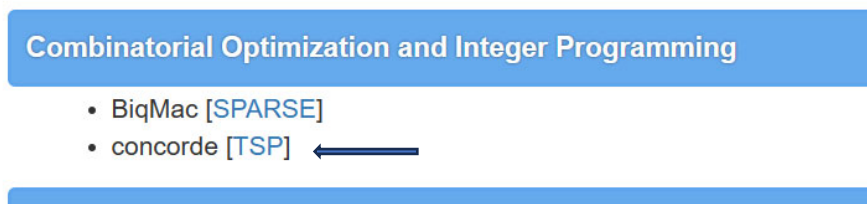
Anexo I

En este Anexo, se podrá encontrar el procedimiento del cálculo mediante el servidor NEOS paso a paso, así como un ejemplo de las salidas de código referente a los cálculos de las coaliciones formadas por tres agentes del Capítulo 3. Es importante tener en cuenta que, lo que el servidor NEOS define como uno, corresponde con la provincia o el agente que forma la coalición a la que restringimos la ruta. Por ejemplo, en el caso de la coalición formada por $(1,2,4)$ el servidor cuando nombra en la ruta al agente tres, este valor corresponde con la provincia número 4.

Como bien comentamos anteriormente, para obtener los valores de las coaliciones intermedias de los agentes, hemos utilizado el servidor NEOS. Para ello, primero nos dirigimos a la página del servidor [Servidor NEOS](#). Una vez ahí, nos dirigimos al apartado “Submit a job to NEOS”:



Una vez dentro de este apartado (al que podemos acceder mediante este hipervínculo [Submit job to NEOS](#)), debemos dirigirnos dentro de la especificación “Problem Type” al problema de optimización combinatoria [\[TSP\]](#):



Seguidamente, debemos introducir los datos mediante un archivo de texto en formato TSPLIB y añadimos el archivo en el siguiente enlace: [Elegir archivo](#).

```

NAME : Ejemplo matriz de distancias de las provincias
COMMENT : Formato TSPLib para matriz de distancias de las provincias (simétricas)
TYPE: TSP
DIMENSION :7
EDGE_WEIGHT_TYPE : EXPLICIT
EDGE_WEIGHT_FORMAT : FULL_MATRIX
EDGE_WEIGHT_SECTION :
0      263  263  184  107  65   256
263    0   172  90   244  200  142
263    172  0   133  206  294  311
184    90   133  0   165  161  222
107    244  206  165  0   171  324
65     200  294  161  171  0   189
256    142  311  222  324  189  0
EOF

```

Concorde data(TSPLIB format file)

Or, enter the complete path to the symmetric TSPLIB file

No se ha seleccionado ningún archivo

Una vez introducido estos, simplemente debemos de marcar la casilla “Short Priority: submit to higher priority queue with maximum CPU time of 5 minutes”, añadir nuestro correo electrónico y clicar en el Submit to NEOS. Al cabo de unos minutos (incluso, muchas veces, segundos) obtendremos en nuestra pantalla el resultado, así como un correo donde se resume este.

Additional Settings

☐ Dry run: generate job XML instead of submitting it to NEOS

☒ Short Priority: submit to higher priority queue with maximum CPU time of 5 minutes

E-Mail address:

Please do not click the 'Submit to NEOS' button more than once.



Submit to NEOS

Clear this Form

Asimismo, para el cálculo de los valores de las coaliciones intermedias debemos realizar el mismo proceso, pero, en este caso, adjuntando la matriz de costes en la que sólo estén los costes de los agentes que forman dicha coalición. Por ejemplo, para el caso de $v(1,3,4)$ con $S = \{1,3,4\}$ añadiríamos en el servidor el archivo de texto formado por:

```

NAME : Ejemplo matriz de distancias de las provincias
COMMENT : Formato TSPlib para matriz de distancias de las provincias (simétricas).Coalicion v(1,3,4)
TYPE: TSP
DIMENSION :4
EDGE_WEIGHT_TYPE : EXPLICIT
EDGE_WEIGHT_FORMAT : FULL_MATRIX
EDGE_WEIGHT_SECTION :
0 263 184 107
263 0 90 244
184 90 0 165
107 244 165 0
EOF

```

Otro ejemplo, podría ser la coalición **v(2,3,5)** donde la ruta está restringida a $S = \{2,3,5\}$, y cuyo código de salida es:

```

/home/neos6/bin/concorde.cplex -s 99 -f sample.tsp
Host: sokrates Current process id: 83722
Using random seed 99
Problem Name: Ejemplo matriz de distancias de las provincias
Formato TSPlib para matriz de distancias de las provincias (simétricas).Coalicion v(2,3,5)
Problem Type: TSP
Number of Nodes: 4
Explicit Lengths (CC_MATRIXNORM)
Optimal Solution: 622.00
Total Running Time: 0.00 (seconds)

```

```

*** ***

```

```

*** You chose the Concorde(CPLEX) solver ***

```

```

*** Cities are numbered 0..n-1 and each line shows a leg from one city to the next
    followed by the distance rounded to integers***

```

```

4 4
0 1 263
1 2 133
2 3 161
3 0 65

```

Anexo II

En este Anexo, se podrán encontrar un ejemplo de las salidas de código referentes a los cálculos de las coaliciones formadas por cuatro agentes del Capítulo 3. Es importante recordar que, como bien se comentó en el Anexo I, lo que el servidor NEOS define como uno, corresponde con la provincia o el agente que forma la coalición a la que restringimos la ruta. Así, por ejemplo, para la coalición $v(1,2,3,5)$ con la ruta restringida a $S = \{1,2,3,5\}$, el servidor cuando define en la ruta el agente cuatro, este valor hace referencia a la provincia 5 que es el agente número cuatro de esa coalición. El código de salida obtenido para esta coalición es:

```
/home/neos5/bin/concorde.cplex -s 99 -f sample.tsp
Host: neos Current process id: 65583
Using random seed 99
Problem Name: Ejemplo matriz de distancias de las provincias
Formato TSPLib para matriz de distancias de las provincias (simétricas). Coalición v(1,2,3,5)
Problem Type: TSP
Number of Nodes: 5
Explicit Lengths (CC_MATRIXNORM)
Optimal Solution: 751.00
Total Running Time: 0.00 (seconds)

***

*** You chose the Concorde(CPLEX) solver ***

*** Cities are numbered 0..n-1 and each line shows a leg from one city to the next
followed by the distance rounded to integers***

5 5
0 4 65
4 3 161
3 1 90
1 2 172
2 0 263
```

Se ha realizado el mismo procedimiento para la coalición $v(2,3,4,6)$ donde la ruta está restringida a $S = \{2,3,4,6\}$, tenemos:

```
/home/neos6/bin/concorde.cplex -s 99 -f sample.tsp
Host: sokrates Current process id: 64625
Using random seed 99
Problem Name: Ejemplo matriz de distancias de las provincias
Formato TSPLib para matriz de distancias de las provincias (simétricas). Coalicion v(2,3,4,6)
Problem Type: TSP
Number of Nodes: 5
Explicit Lengths (CC_MATRIXNORM)
Optimal Solution: 924.00
Total Running Time: 0.00 (seconds)

***

*** You chose the Concorde(CPLEX) solver ***

*** Cities are numbered 0..n-1 and each line shows a leg from one city to the next
followed by the distance rounded to integers***

5 5
0 3 107
3 1 206
1 2 133
2 4 222
4 0 256
```


Anexo III

En este anexo adjuntaremos un ejemplo de uno de los resultados obtenidos en el servidos NEOS de las coaliciones formadas por cinco agentes. Teniendo en consideración el procedimiento , nombrado en el Anexo I y Anexo II, que realiza el servidor NEOS. Así, el código de salida obtenido para la coalición $v(1,2,3,4,5)$ donde la ruta está restringida a $S = \{1,2,3,4,5\}$, es:

```
/home/neos6/bin/concorde.cplex -s 99 -f sample.tsp
Host: sokrates Current process id: 71925
Using random seed 99
Problem Name: Ejemplo matriz de distancias de las provincias
Formato TSPlib para matriz de distancias de las provincias (simétricas). Coalicion v(1,2,3,4,5)
Problem Type: TSP
Number of Nodes: 6
Explicit Lengths (CC_MATRIXNORM)
Optimal Solution: 801.00
Total Running Time: 0.00 (seconds)

***

*** You chose the Concorde(CPLEX) solver ***

*** Cities are numbered 0..n-1 and each line shows a leg from one city to the next
    followed by the distance rounded to integers***

6 6
0 5 65
5 3 161
3 1 90
1 2 172
2 4 206
4 0 107
```

Anexo IV

En este anexo se adjunta el resultado obtenido con el servidor NEOS de la coalición total $v(1,2,3,4,5,6)$.

```
/home/neos5/bin/concorde.cplex -s 99 -f sample.tsp
Host: neos Current process id: 70801
Using random seed 99
Problem Name: Ejemplo matriz de distancias de las provincias
Formato TSPlib para matriz de distancias de las provincias (simétricas)
Problem Type: TSP
Number of Nodes: 7
Explicit Lengths (CC_MATRIXNORM)
Optimal Solution: 932.00
Total Running Time: 0.00 (seconds)
```

```
*** ***
```

```
*** You chose the Concorde(CPLEX) solver ***
```

```
*** Cities are numbered 0..n-1 and each line shows a leg from one city to the next
    followed by the distance rounded to integers***
```

```
7 7
0 5 65
5 6 189
6 1 142
1 3 90
3 2 133
2 4 206
4 0 107
```