Trabajo Fin de Grado

Grado en Ingeniería de las Tecnologías Industriales

Estudio Experimental de diferentes modelos matemáticos para resolver el problema TSP

Autor: Ana Corral Sánchez

Tutor: José Manuel García Sánchez

Dep. Organización Industrial y
Gestión de Empresas I

Universidad de Sevilla

Sevilla, 2020

Proyecto Fin de Carrera
Ingeniería de las Tecnologías Industriales

Estudio Experimental de diferentes modelos matemáticos para resolver el problema TSP

Autor:

Ana Corral Sánchez

Tutor:

José Manuel García Sánchez

Profesor titular

Dpto. de Organización Industrial y Gestión de Empresas I

Escuela Técnica Superior de Ingeniería

Universidad de Sevilla

Sevilla, 2020

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

Presidente:

Vocales:

Secretario:

Acuerdan otorgarle la calificación de:

El Secretario del Tribunal

Sevilla, 2020

INDICE GENERAL

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

1. OBJETIVO DEL TRABAJO

Actualmente, existen diversos métodos y estrategias para la resolución del Problema del Viajante de Comercio (TSP), es decir, "The Travelling Salesman Problem" como sus iniciales indican, ya que quizás es el problema de optimización combinatoria más ampliamente investigado.

Para la solución de este problema existen diferentes estrategias, tanto métodos aproximados, donde entrarían las heurísticas y las metaheurísticas, como el uso de métodos exactos, donde se sitúa este trabajo. El objetivo principal de este trabajo será analizar y comparar cuatro modelos matemáticos diferentes que permitan la resolución óptima de los diferentes problemas TSP o una solución factible pasado un tiempo razonable. Se utilizará una batería de problemas que contemplará tanto escenarios simétricos como asimétricos respecto a la distancia entre nodos.

Los modelos matemáticos se implementarán en una herramienta informática de optimización llamada Lingo, en la cual se formularán los modelos con su propio lenguaje, y los enlazaremos con los problemas situados en ficheros de texto (".txt") que contendrán todos los datos necesarios para su resolución.

En cuanto a los modelos, tendremos tres versiones principales. Por un lado, el modelo propuesto por Miller, Tucker y Zemlin (1960) [5], por otro lado, el modelo propuesto por Desrochers y Laporte (1988) [9], que es muy similar al modelo MTZ, y como tercer modelo el propuesto por Sarin (1992) [2]. Finalmente, el último modelo de estudio será un modelo intermedio entre el propuesto por Miller, Tucker y Zemlin y el propuesto por Desrochers y Laporte, el cual recibirá el nombre del Modelo Intermedio.

La finalidad del trabajo es obtener datos concluyentes y reales, que sirvan de ayuda a la hora de escoger uno de los cuatros métodos dependiendo del tipo de problema al que nos estemos enfrentando.

2. INTRODUCCIÓN

El problema del viajante de comercio, más conocido como el TSP (Travelling Salesman Problem), es uno de los problemas más conocidos de la historia, especialmente tienen una gran relevancia y aplicación en el ámbito de la logística y la distribución, así como en otros entornos dentro del campo de la optimización. Los problemas TSP se puede emplear en cualquier situación que requiera seleccionar nodos en cierto orden que reduzca los costes.

Estos problemas se basan generalmente en un viajante, el cual quiere visitar n ciudades, pasando una sola vez por cada una de ellas, empezando por una ciudad origen en la que se acabará el propio recorrido. El fin a la hora de resolver estos problemas, es decidir cuál es el recorrido que debe seguir el viajero para que el coste total, es decir, ya sea por costes monetarios o sea distancia total recorrida, sean los mínimos.

El problema del TSP trata de un viajero y de distintas ciudades, pero se podría hacer de forma genérica y ser modelado como un grafo de manera que las ciudades sean los diferentes vértices del grafo o los nodos, los caminos entre las ciudades sean los arcos que se forman al ir de un vértice a otro, y los costes la distancia que hay entre ellos. De esta forma, se utilizada el problema del TSP de forma más generalizada, donde los vértices podrían ser localidades, puntos, empresas o tiendas entre otros, y el viajante, que es el que realiza los distintos caminos o arcos en este caso, podría ser cualquier entidad, sea una persona o un vehículo. [5]

Los problemas a la hora de resolverlos es que tienen una gran complejidad ya que pertenecen a la clase de problemas NP-completos (problemas que requieren un número de pasos exponencial respecto al tamaño del problema para garantizar optimalidad) [12]. Esto implica que el tiempo de ejecución de cualquier algoritmo, para poder resolverlos, aumenta de forma exponencial con respecto al número de ciudades. Por lo tanto, un problema TSP con un gran número de nodos, es muy difícil de resolver sin la utilización de herramientas informáticas.

Como la demanda del TSP es bastante elevada gracias a sus múltiples utilidades en diversos ámbitos, en los últimos años se han desarrollado grandes evoluciones para poder resolverlos con mayor facilidad. Hay muchas estrategias y métodos diferentes, los cuales explicaremos en el desarrollo del trabajo.

En este trabajo analizaremos el Problema del viajante, explicando previamente en el capítulo 3 su historia, definición y aplicación en la vida real. Posteriormente, en el capítulo 4, se detallarán las diferentes estrategias y métodos de resolución que tiene, centrándonos en los modelos matemáticos ya que será la forma de resolución que vamos a usar. Primero, desarrollaremos estos modelos matemáticamente explicando y detallando sus elementos, variables, restricciones y función objetivo en el capítulo 5. Una vez planteados los cuatro modelos que vamos a utilizar, añadiremos en el capítulo 6 todos los problemas que vamos a desarrollar, tanto los simétricos como los simétricos, para poder entender mejor su estructura. En el capítulo 7, se explicará todo lo relacionado con la herramienta informática que vamos a disponer para la resolución de los problemas TSP, es decir, Lingo, tanto su lenguaje como la descripción de los cuatro modelos finales. Finalmente, una vez explicado todo lo anterior, comenzará el apartado de la experimentación, es decir, el capítulo 8, donde detallaremos el procedimiento que se ha llevado a cabo, los resultados obtenidos por los cuatro modelos, y las distintas comparaciones de los modelos basándonos en diferentes criterios. Se añadirán los capítulos 9 y 10 para hacer una breve conclusión del trabajo y añadir las referencias.

3. El PROBLEMA DEL VIAJANTE GENERALIZADO

3.1. Historia del TSP

Los orígenes del problema del viajante de comercio no son totalmente claros. El primer conocimiento que se tiene de él fue a partir de 1832 en Alemania [10] en un libro titulado “El problema del viajero”. En este libro se desarrolla qué se debe hacer para obtener éxito en los negocios, el cual, podría ser considerado como la primera referencia bibliográfica del problema del TSP al comentar que con una elección apropiada del tour a realizar, se puede ahorrar mucho coste, es decir, tiempo y distancia, y que el aspecto más importante es cubrir tantas ciudades como sean posibles sin visitar una de ellas dos veces. En cuanto a tratamientos matemáticos, no contenía nada, pero sí cinco rutas que recorren regiones de Alemania y Suiza, una de las cuales será la solución óptima al problema del TSP planteado [4].

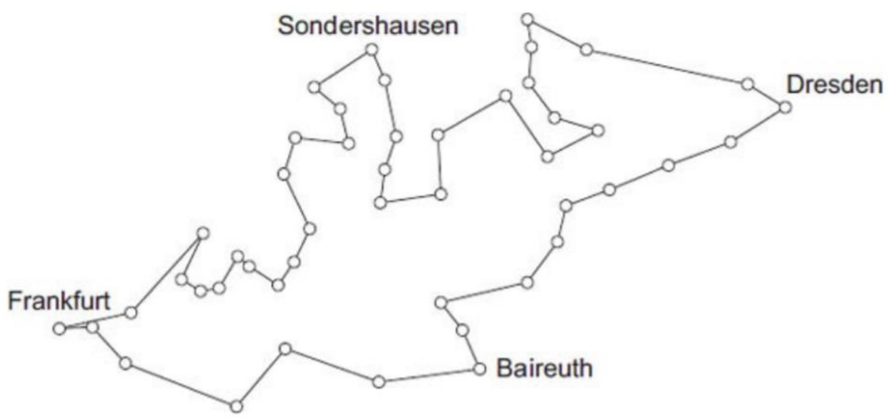


Fig.1 Rutas de Alemania, problema del TSP

Fuente: The travelling Salesman Problem, A computational Studio [4]

Sin embargo, el problema matemático surgió a través del “Hamilton Icosian Game”, un juego matemático desarrollado en 1857 por William Rowan Hamilton [10]. El objetivo del juego era dar con el recorrido óptimo de las aristas de un dodecaedro para visitar una sola vez cada vértice de la figura, y que a su vez el vértice de partida coincidiera con el de llegada. El desafío se planteaba como un tablero con agujeros en cada nodo del grafo del dodecaedro y su objetivo era la búsqueda de un ciclo de Hamilton, nombre que se puso en honor a su autor [6].

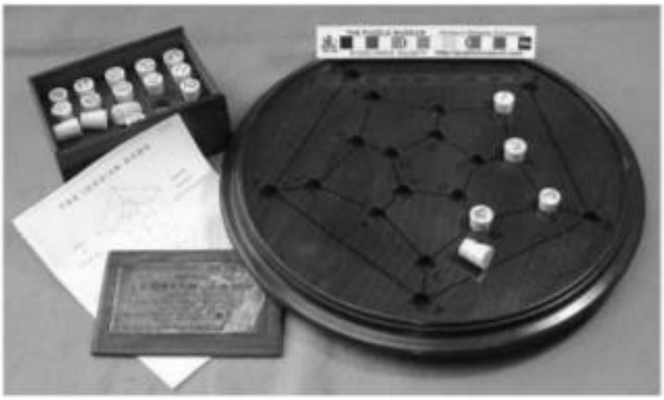


Fig.2 Hamilton Icosian Game

Fuente: Cooperación en los problemas del viajante y rutas de vehículos [6]

A raíz de esto, el problema del TSP ha ido evolucionando y obteniendo una gran variedad de estrategias y métodos de resolución, así como aumentando el número de nodos, y por lo tanto aumentando su complejidad.

En cuanto a sus estrategias de resolución, las que vamos a desarrollar en el trabajo serán métodos exactos usando modelos matemáticos, concretamente el primer modelo planteado será el MTZ, el cual debe su nombre a sus propios autores, Miller, Tucker y Zemlin (1960). Este modelo se formuló para introducir la eliminación de los subtours que se forman al ir de un nodo a otro, concepto que se explicará a lo largo del trabajo. Posteriormente, se hicieron varias versiones y mejoras del modelo MTZ, las cuales nosotros nos centraremos en el modelo DL, de Desrochers y Laporte, y otro modelo intermedio entre estos dos. Finalmente, analizaremos y experimentaremos con el modelo de Sarin.

3.2. Definición del TSP

El problema del viajante de comercio (TSP), debe su nombre a que la enunciación más común de este problema habla de un viajante de comercio que tiene que partir de su casa para visitar una serie de clientes ubicados cada uno en una ciudad diferente, y volver finalmente de nuevo a su casa. Sin embargo, se pueden plantear muchos otros enunciados similares que conlleven las mismas condiciones y se busque un mismo fin, reducir los costes, tanto de tiempo, de distancia o de precio.

Otros enunciados similares, podrían ser: un transportista que necesita hacer una ruta para entregar mercancías a varias tiendas, cada una situada en puntos diferentes de una localidad, o un avión que quiere recorrer todos los países de un mismo continente. Todos buscan el mismo fin, y tienen prácticamente que cumplirse las mismas condiciones:

- La entidad, que puede ser persona o transporte entre otros, debe salir de un punto inicial de origen, y visitar los N puntos restantes.
- Cada punto para visitar tiene que ser visitado sólo una vez.
- El coste, que supone ir de un punto a otro, debe ser conocido. Este coste podría ser distancia, tiempo o costes monetarios.
- Al terminar el recorrido, la entidad debe volver a su punto de origen cerrando así el grafo que forma.

3.3. Aplicaciones en la vida real

El problema del viajante de comercio es un problema planteado en muchos ámbitos diferentes, ya que el fin que busca se puede moldear a diversas situaciones similares. Entre ellos, destaca en el ámbito de la logística y distribución, en la cadena de producción, en la elaboración de microchips, en la agrupación de datos, en el turismo y agencias de viajes, en secuencias…

3.3.1. Logística y distribución

Una aplicación muy común del problema del TSP es en el movimiento en sí de personas, equipos y vehículos para proporcionar un servicio, ya sea repartir mercancías, visitar ciertos puntos concretos o con otros fines.

Cuando hablamos del "transporte" de estas entidades, nos referimos a la logística que lleva esa entidad en desarrollar su función.

Dentro del campo de la logística, caben destacar ciertos puntos donde se utiliza frecuentemente el problema del TSP para poder trabajar:

- Turismo:
  Muchas agencias de viaje hacen uso de un software para resolver problemas TSP, y así poder organizar sus viajes de una forma más económica tanto para ellos como para sus propios clientes, proporcionando un viaje más completo.

  No solo tienen que ser las agencias de viajes las que usen esto. También las propias personas a la hora de viajar por su cuenta usan métodos de resolución de los problemas TSP, sobre todo en viajes largos, donde se necesitan visitar muchos puntos o sitios diferentes para una mejor organización y un aprovechamiento máximo del viaje.

- <u>Vendedores:</u>
  En cuanto a vendedores, el problema TSP se implementa de tal forma que el viajante es un vendedor, y quiere recorrer el máximo número de puntos distintos sin pasar por los mismos, vendiendo así el mayor número posible de su producto, y con el objetivo a su vez de ahorrar lo máximo en costes, ya sea de tiempo, como de distancia o gasolina. Para ello, antes de partir, se crean rutas óptimas en el mapa de puntos interesados para cumplir con el objetivo.

  Dependiendo del número de ciudades, se obtendrá una gran cantidad de rutas diferentes que tomar, decidiéndose por la que mayor beneficio le proporcione.

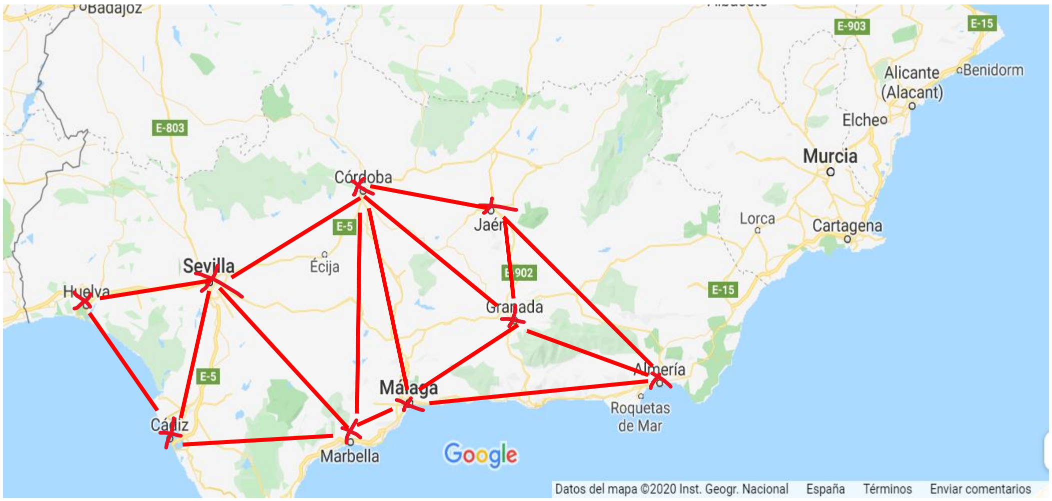


Fig.3 Recorridos de un vendedor

Fuente: Elaboración Propia

Un vendedor quiere vender su producto por las ciudades marcadas de Andalucía, partiendo de la ciudad origen, que es Sevilla. Traza todos los diferentes caminos posibles a realizar, y se da cuenta que no sabe bien qué ruta hacer para poder ahorrar lo máximo en tiempo, y en distancia, lo cual le ahorraría en combustible.

En este tipo de casos, en la vida real, las compañías realizan un estudio con el uso de herramientas apropiadas, para detallar qué recorrido sería el óptimo.

- <u>Envíos postales:</u>
  El problema TSP juega un papel importante en los problemas postales generales, donde las casas o calles están muy alejadas unas de otras, o donde solo es necesario visitar un subconjunto de casas como en la entrega de paquetes.
  Por ello, en este ámbito se ha adoptado recientemente el problema TSP en softwares para el uso en las aplicaciones postales. Un ejemplo es la empresa Rapidis [4], que emplea un módulo heurístico de Concorde para trazar rutas para sus clientes de Forbruger-Kontakt, un distribuidor de material publicitario y muestras que opera en Dinamarca y otros países.

- Secuencias:
  Cuando hablamos de secuencia, nos referimos al orden en el cual n trabajos tienen que ser procesados de tal forma que se minimice el costo total de producción.

- Rutas de transporte laborales o escolares:
  Muy similar a lo que sucede con el vendedor o el cartero, pero en este caso surge con la necesidad de crear la ruta óptima a la hora de recoger a trabajadores o alumnos en sus respectivas casas o paradas, para llevarlas al trabajo o colegio.

Fig.4 Rutas escolares en la Ciudad de Valencia
Fuente: Routing Maps, empresa de Rutas

3.3.2. Secuencia de Genomas

El problema TSP también juega un papel importante en la secuencia de genomas, ya que nos proporciona una herramienta para construir las secuencias a partir de datos experimentales sobre la proximidad de pares individuales de marcadores.

Cada cromosoma, situado en el mapa del genoma, tiene una secuencia de marcadores con cierta estimación de la distancia entre los marcadores adyacentes. Estos marcadores de los mapas son conocidos como segmentos de ADN, y pueden detectarse en un laboratorio de forma fiable. La capacidad de reconocer estos segmentos permite a los investigadores a utilizarlos para poder comparar y combinar mapas físicos creados en diferentes laboratorios [4].

El problema TSP entra en juego, a la hora de tener información sobre del orden en el que aparecen los marcadores en el genoma.

El mapeo de radiación híbrida (RH), desarrollada por Goss y Harris [4], es una de las técnicas principales para poder obtener datos sobre la posición relativa de los marcadores, creando los segmentos a partir de Rayos X.

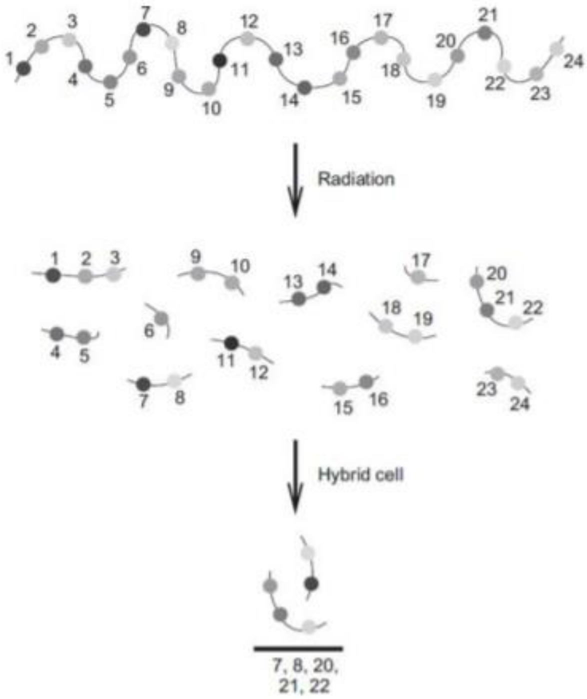


Fig.5 Mapeo de Radiación Híbrida

Fuente: The travelling Salesman Problem, A Computational Studio [4]

3.3.3. Placas de circuitos y Microchips

Esta es una de las utilidades más favorables que puede plantear el problema del viajante de comercio. La creación de placas de circuitos se enfoca en dos problemas distintos, los cuales pueden ser solucionados gracias al problema TSP.

- Problemas de perforado:
  El fin es buscar el orden óptimo de taladrar las placas.
  En este caso, tomaremos las ciudades como las posiciones a perforar y las distancias entre ellas como el tiempo que necesita la máquina en trasladarse de una a otra. La ciudad origen en este caso será un punto adicional, donde permanecerá la perforadora mientras no está trabajando.

- Problemas de conexión de chips:
  La idea principal es minimizar la cantidad de cable necesario para poder unir todos los puntos de una placa sin que haya interferencias, dando lugar así a errores.

Los chips son normalmente de tamaño bastante pequeño, es imposible poder poner más de dos cables en un único pin. Por lo tanto, en este caso, ahora tomamos las ciudades como los pins, y la cantidad de cable para unirlos como la distancia, intentando minimizarla al máximo [16].

3.3.4. Otras aplicaciones

- Inspecciones a sitios remotos:
  Ordenar los lugares que deberá visitar un inspector en el menor tiempo.
- Aplicaciones en Internet:
  Supongamos que el viajante de comercio es un bit de datos, y que las ciudades son servidores de Red distribuidos por todo el planeta.
  Esta variante del problema del viajante de comercio puede simularse en el uso óptimo de una plataforma masiva de distribución como es Internet.

- Problema de Scheduling:
  Este problema es más complejo de resolver, ya que se formula de tal manera que hay "T" tareas que realizar y "m" procesadores. El objetivo a la hora de utilizar el problema TSP es buscar una planificación en "m" procesadores para "T", intentando minimizar el tiempo.

4. MÉTODOS DE RESOLUCIÓN DEL TSP

Para la resolución del problema TSP, han ido evolucionando a lo largo de la historia diversas formas, estrategias y métodos. Algunos de estos métodos inicialmente eran realizados a mano, cuando el número de entidades o nodos era bastante pequeño, pero cuando empezaron a utilizarse en casos en los cuales la complejidad aumentaba exponencialmente debido al aumento de los nodos, tuvieron que poner en marcha distintos software y herramientas informáticas para su resolución.

En este apartado, hablaremos de distintos métodos que se han ido llevando a cabo, y finalmente con cual de todos estos métodos nos quedaremos para poder resolver los problemas planteados en el apartado anterior.

4.1. Métodos exactos

Cuando mencionamos métodos exactos, nos referimos a métodos que nos proporcionan un resultado final óptimo, es decir, la mejor solución que nos proporciona es la correcta. Para explicar un poco estos métodos, es importante mencionar que encontrar el óptimo exacto es más complicado de lo que parece, por ello muchos de los métodos que nos proporcionan un resultado exacto, serán para casos con complejidad baja, es decir, con pocos nodos que recorrer. A pesar de ello, hay métodos que son exactos y además el tiempo de ejecución se encuentra en un rango bastante aceptable para un gran número de nodos.

Algunos de estos métodos podrían ser:

- Método de Branch and Bound:

  El método de Branch and Bound o también conocido como ramificación y poda, nos proporciona una solución óptima del problema del agente viajero. Para ello hay que ir calculando la solución del modelo mediante el algoritmo Simplex, es decir, un método numérico que busca un mínimo o un máximo local de una función cualquiera, examinando en cada paso los vértices de un Simplex.
  A medida que el tamaño de la red aumenta, es decir, al ir aumentando el número de nodos, el tiempo de resolución también aumenta a gran escala, por lo que este método solo sería útil para redes de pequeño tamaño [15].

- Método de la fuerza bruta:

  El método de la fuerza bruta tan solo consiste en la exploración de todos los recorridos posibles, sin la aplicación de ningún algoritmo sistemático. Por ello, es un método, que se puede implementar y utilizar en problemas que tengan un número mínimo de nodos, ya que al contrario sería interminable.
  A continuación, mostramos un ejemplo de problema resolviéndose con el método de la fuerza bruta, donde los costes son las distancias entre los nodos en kilómetros, y el nodo origen es el punto A. Como se puede observar, se trazan todos los trazos posibles que

hay de un nodo a otro, y calculando todos los recorridos completos, sumando los distintos costes que implican, podría deducirse el camino y coste óptimo.

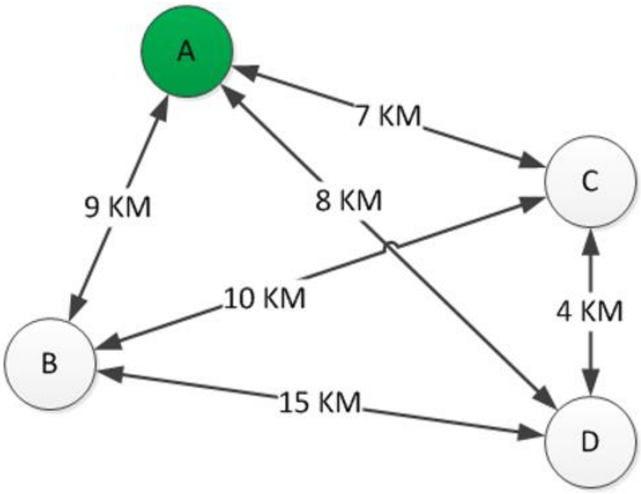


Fig.6 Ejemplo de ejercicio con la Fuerza Bruta

Fuente: Investigación de Operaciones, TSP. Ingeniería Industrial online [15]

- Modelos matemáticos:

Finalmente, en este apartado de métodos exactos, mencionaremos los modelos matemáticos. Con distintos modelos matemáticos, es decir, formulaciones de ciertas restricciones, funciones objeticos y variables, podemos conseguir un resultado del problema del TSP exacto.
Si estos modelos matemáticos los resolvemos sin la utilización de ninguna herramienta informática o software, nos encontraríamos en la misma situación que los dos métodos anteriores, es decir, solo nos servirían para casos con poco número de nodos. Por el contrario, si usamos programas o aplicaciones informáticas, con este método, podríamos encontrar un resultado exacto de un problema del TSP con un gran número de ciudades.
En nuestro trabajo, nos vamos a basar en estos métodos, usando cuatro modelos matemáticos diferentes, y ayudándonos de la herramienta Lingo (el cual usa un Branch and bound realmente para resolverlo), la cual nos permite que podamos hacer los problemas con una gran cantidad de ciudades, en un tiempo de ejecución bastante aceptable.
Los modelos que vamos a desarrollar, y los cuales explicaremos en el siguiente apartado serán:

- Modelo MTZ
- Modelo DL
- Modelo Intermedio
- Modelo de Sarin

4.2. Métodos aproximados

Cuando hablamos de métodos aproximados, nos referimos a aquellos en los que el resultado final que nos proporcionan no es exacto, si no próximo a él. Aunque no sean exactos, nos dan un óptimo bastante bueno y útil para resolver muchos problemas en la vida real. Además, estos métodos son mucho más sencillos de realizar y su tiempo de ejecución también se reduce bastante. Entre estos métodos podemos utilizar:

- <u>Métodos heurísticos:</u>

  Son algoritmos que con poco esfuerzo computacional proporcionan una solución aproximada, pero no necesariamente óptima del problema [13].
  Alguno de los métodos más utilizados para resolver el problema del TSP dentro de las heurísticas, son:

    - Métodos constructivos, son métodos deterministas en los que se suelen ir construyendo paso a paso una solución y se van eligiendo las mejores en cada iteración. Algunos ejemplos son el método del vecino más próximo o heurísticos de intersección.
    - Métodos de búsqueda local, o también llamados procedimientos de búsqueda. Estos procedimientos comienzan con una solución del problema y después van mejorando mientras se pueda la solución actual hasta que esta no pueda ser mejorada. La solución final se denominará óptimo local. Un ejemplo es el procedimiento de 2 intercambios.
    - Métodos combinados, que son los métodos en los que se combinan y mezclan los dos métodos anteriores, es decir, los de búsqueda y los constructivos.

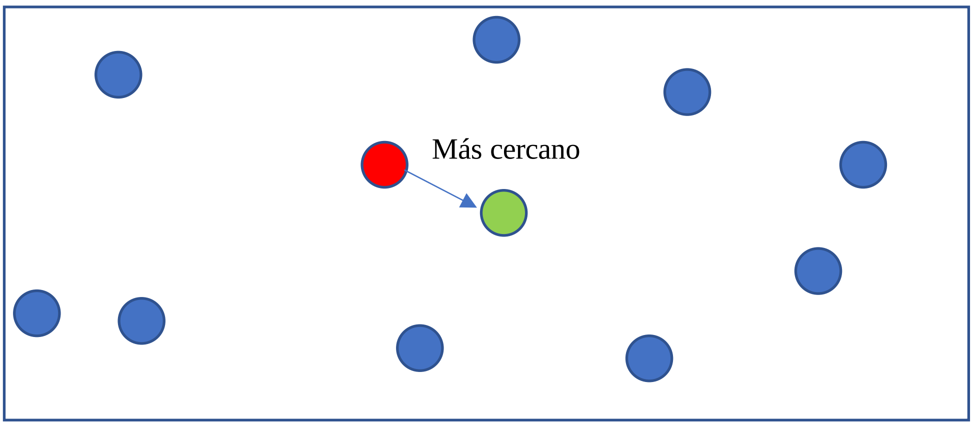


Fig.7 Ejemplo del método del vecino más próximo

Fuente: Elaboración propia

4.3. Formulación general TSP

Finalmente, como hemos explicado anteriormente, para la resolución de nuestros problemas TSP, vamos a utilizar dentro de los métodos exactos, unos modelos matemáticos con ayuda de la herramienta informática Lingo. Hablando ahora en términos del problema del viajante en sí, el objetivo es encontrar el orden en el cual se deben visitar cada una de las ciudades para poder minimizar la distancia recorrida. Todas las soluciones posibles a este problema, las llamaremos “tour”, sean las óptimas o no.

Para empezar a formular estos problemas, hay que tener en cuenta que existen dos condiciones obvias que deben cumplirse siempre:

- Condición 1: Exactamente una ciudad debe ser visitada inmediatamente después de la ciudad i [1].
  Esta condición podría ser representada matemáticamente como:

$$\sum_{j}^{n} x_{ij} = 1\,; \qquad \forall i = 1 \dots n;\;\; i \neq j$$

- Condición 2: Exactamente una ciudad debe ser visitada inmediatamente antes de la ciudad j [1].
  Esta condición podría ser expresada matemáticamente del siguiente modo:

$$\sum_{i}^{n} x_{ij} = 1\,; \qquad \forall j = 1 \dots n;\;\; i \neq j$$

El valor Xij valdrá 1 cuando después de la ciudad i, el viajante vaya directamente a la ciudad j, en caso contrario, valdrá cero. A pesar de estas dos restricciones, no serían suficientes para plantear el modelo del problema TSP, y esto se puede demostrar con el siguiente ejemplo:

Problema de ejemplo 1

Se nos plantea un problema en el que se quieren recorrer 7 ciudades diferentes, pasando por todas ellas, pero sin repetir y situadas de la siguiente manera:

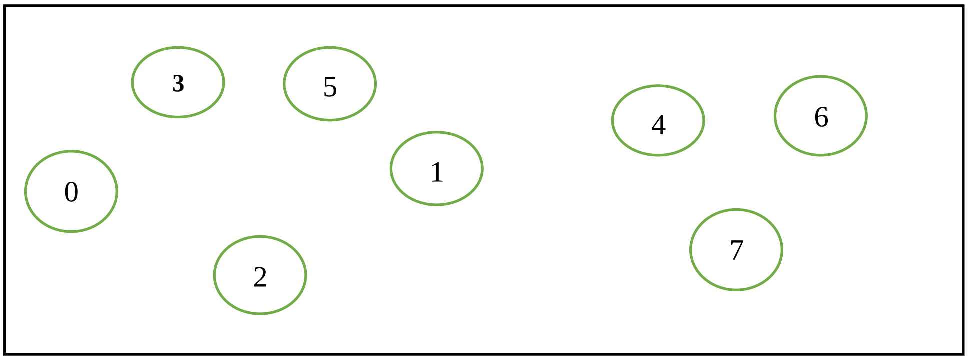


Fig.8 Problema TSP de ejemplo, datos
Fuente: Elaboración Propia

Una solución que cumpla con las dos condiciones previamente descritas podría ser la siguiente:

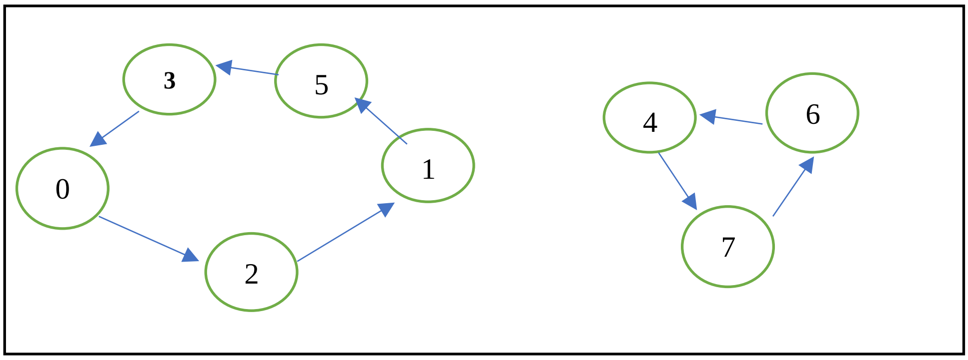


Fig.9 Problema TSP de ejemplo, solución 1
Fuente: Elaboración Propia

Se puede observar a simple vista, que la solución cumple con las restricciones explicadas anteriormente. Pero si nos fijamos bien, no es lo que buscamos, ya que aparecen "subtour".

Los subtour son las rutas o los tours, que o bien no incluyen a la ciudad de origen, es decir, a la ciudad 0, o bien no incluyen a todas las ciudades, aunque empiece y acabe en cero. Para que no suceda este tipo de problemas, habría que añadir al modelo otras restricciones, que serán las que hagan las diferencias entre las distintas formulaciones que vamos a implementar en nuestro trabajo.

Por otro lado, todos los modelos que vamos a estudiar tienen un mismo fin, y por lo tanto compartirán una misma función objetivo: reducir lo máximo posible los costes totales. Esta función podría ser:

$$Min \sum_{i}^{n} \sum_{j}^{n} c_{ij} x_{ij}$$

Donde, el elemento Cij representa el valor exacto en cuanto a costes, que hay en ir desde la ciudad i hasta la j, y el valor Xij, es lo explicado anteriormente, es decir, valdrá 1 si se cumple que directamente después de estar en la ciudad i, se irá a la j.

5. LOS MODELOS MATEMÁTICOS

En el apartado anterior mencionamos que los métodos que finalmente íbamos a utilizar para desarrollar y resolver nuestros problemas tanto simétricos como asimétricos serían dentro de los métodos exactos, los modelos matemáticos con ayuda de una herramienta informática llamada Lingo.

El verdadero fin de este trabajo, no es resolver solo estos problemas del TSP, si no, deducir cuál de los cuatro modelos matemáticos que vamos a utilizar es el mejor, tanto en la búsqueda del óptimo como del tiempo de ejecución que implementa.

En el último subapartado anterior, el apartado 4.3, en la formulación general del problema TSP, explicábamos que existen dos condiciones y una función objetivo común para cualquier método matemático, que son totalmente necesarias para que el problema del TSP pueda resolverse. Pero también añadíamos que con estas dos únicas condiciones se formaban unos "subtour", es decir, unos tours erróneos que no deberían de tenerse en cuenta, y que para hacer desaparecer estos subtour habría que añadir otras restricciones, que serían las que formarían las diferencias entre los cuatro modelos a desarrollar.

Por lo tanto, nuestros cuatro modelos, van a compartir varias variables, las dos condiciones fundamentales y la función objetivo, como hemos explicado antes y las cuales se pueden ver de forma resumida a continuación:

- Función objetivo:

$$Min \sum_{i}^{n} \sum_{j}^{n} c_{ij} x_{ij}$$

- Condición 1:

$$\sum_{j}^{n} x_{ij} = 1$$

- Condición 2:

$$\sum_{i}^{n} x_{ij} = 1$$

Y a raíz de esto, comenzaremos a explicar modelo a modelo, su introducción, sus variables y elementos que participan, sus restricciones y su función objetivo.

5.1. Modelo 1: Modelo MTZ

Este modelo matemático fue planteado por Miller-Tucker-Zemlin en torno al 1960. Lo que se pretendía con este modelo era añadir en la formulación del problema TSP la eliminación de los subtours, lo cual permitía encontrar el óptimo a los problemas.

5.1.1. Elementos y variables

A continuación, se explicarán los elementos que participan en el sistema. Son de forma generalizada y pueden ser cosas, personas, herramientas, lugares o tiempo, entre otros [7].

Estos elementos normalmente tienen asociados cierta información la cual depende de ellos y las llamaremos variables asociadas. Estas variables tienen que contener información, en este caso numérica ya que estamos hablando de un modelo matemático y son las que participarán en las acciones que se producen en el sistema y en las restricciones. Además, estos elementos tendrán asociados unos datos, también numéricos, para poder realizar el problema, Por lo tanto, los elementos, datos y variables asociadas que participan en este modelo son:

Elemento	Tipo	Variables asociadas	Datos
Ciudades/Nodos	Índice i,j=1…N	Ui(posición ciudad i), entero	-
Tramos/Arcos	(i,j)	-	Cij (coste de i a j), entero

Tabla 1. Elementos del modelo MTZ

Fuente: Elaboración propia

- **N:** Será el número total de ciudades que existen en el problema.
- **Tramos:** Son los recorridos que se formarán en ir de una ciudad i a una ciudad j.
- **Cij:** Son los costes que supone ir de una ciudad i una ciudad j, recogidos en la matriz de datos.
- **Ui:** Variable entera que implica el número de ciudades visitadas hasta la ciudad i. Básicamente esta variable indica cual es la posición de visita de la ciudad de partida i.

A las ciudades también las hemos llamado nodos, y a los tramos que realiza el viajante en ir de una ciudad a otra también podrían considerarse como arcos.

En estos problemas también hay que mencionar a las variables de decisión, es decir, son las acciones directas e indirectas que se producen en el sistema. A estas variables, en la ejecución del problema, se les va proporcionando un valor, el cual inicialmente no estará determinado. Estas variables son las que definen las variables principales del modelo y sus valores podrían ser cualquiera, es decir, un valor binario, un entero o un valor continuo. Las variables de decisión de este modelo serán [5]:

- **Xij:** Es una variable binaria (valor 1 o 0), la cual dependerá de si el TRAMOS(i,j) es seleccionado o no para encontrar el óptimo.

$$\boldsymbol{X_{ij}} = \begin{Bmatrix} 1, si\ se\ visita\ la\ ciudad\ j\ inmediatamente\ después\ de\ la\ ciudad\ i \\ 0, si\ no\ se\ visita\ la\ ciudad\ j\ inmediatamente\ después\ de\ la\ ciudad\ i \end{Bmatrix}$$

5.1.2. Restricciones

En este apartado, detallaremos y explicaremos cuales son las restricciones que se usan en el modelo MTZ. Las restricciones son condiciones y normas necesarias para poder cumplir con nuestro objetivo [8].

R1) $\sum_j^n x_{ij} = 1 \quad ; \quad \forall i = 1 \ldots n; \ \ i \neq j$

R2) $\sum_i^n x_{ij} = 1 \quad ; \quad \forall j = 1 \ldots n; \ \ i \neq j$

R3) $u_i - u_j + nx_{ij} \leq n - 1 \quad ; \quad \forall j, i = 1 \ldots n; \ \ i \neq j$

R4) $u_i \leq n \quad ; \quad \forall i = 1 \ldots n$

R5) $x_{ij} \in [0,1] \quad ; \quad \forall j, i = 1 \ldots n$

A continuación, explicamos el significado de cada una de ellas:

R1) Esta restricción es general para todos los modelos, significa que exactamente una ciudad debe ser visitada inmediatamente después de la ciudad i. Aquí va variando continuamente la ciudad j, mientras que la ciudad i está fijada, por lo tanto, como la variable Xij valdrá 1 en caso de que ese tramo sea recorrido, tiene que existir si o si un solo camino desde cualquier ciudad j a la ciudad i fijada.

R2) Esta restricción también es general para todos los modelos, y hace lo mismo que la anterior, refiriéndose a que exactamente una ciudad debe ser visitada inmediatamente antes de la ciudad

j. El procedimiento que utiliza es el mismo que el anterior, pero de forma contraria, es decir, ahora fijando la ciudad j.

R3) Esta restricción es la que elimina los subtours explicados anteriormente, y por lo tanto la que hace diferencia al modelo MTZ. Aquí se han añadido dos conjuntos de variables nuevas, que indican la posición de visita que tiene la ciudad de partida i, y la que tiene la ciudad de llegada j.

Como la ciudad de partida i, tiene que tener una posición anterior a la posición de llegada, primero se desarrolló esta ecuación [8]:

$$u_j \leq u_i + 1$$

Con esto se rompía el subtour entre dos ciudades. Además, en este conjunto de variables Ui, tendría que haber una variable Ux que indicara que era la primera ciudad visitada de todo el recorrido, y otra variable Uy que indicara que era la última ciudad visitada antes de volver a la ciudad origen (ciudad 0). Por lo tanto, habría que añadir algo a la ecuación que relacionara automáticamente a la última ciudad visitada con la ciudad origen. De esta manera, se reformuló la última restricción y se dejó tal y como está descrita arriba.

R4) Esta restricción obliga a que la variable contador de las ciudades visitadas hasta la i, es decir, la variable Ui, sea igual o inferior al número total de ciudades de nuestro problema. Esto es, ya que la variable Ui indica la posición de visita de la ciudad i, que máximo tendría la última posición, es decir, la posición n.

R5) La última restricción, es simplemente para indicar que nuestra variable de decisión Xij, será binaria, ya que como hemos explicado antes, valdrá 1 si se elige el tramo desde la ciudad i hasta la j, y 0 en caso contrario.

5.1.3. Función Objetivo

La función objetivo, como hemos explicado en la introducción de este apartado, será común para nuestros cuatro modelos a estudiar. Esto se debe a que el fin de la resolución de los problemas es el mismo, realizar un recorrido que cumpla con todas las condiciones anteriores, pero a su vez que ahorre lo máximo en costes, que en este caso los costes serán las distancias que hay entre unas ciudades y otras.

Estos costes son representados en las variables Cij, y se formula de la manera expuesta abajo, de tal forma que para todo el conjunto de ciudades i y j, el sumatorio de todos los camino elegidos (cuando Xij vale 1) multiplicado por el coste que suponen dichos caminos, sea lo mínimo posible:

$$Min \sum_{i}^{n} \sum_{j}^{n} c_{ij} x_{ij}$$

5.2. Modelo 2: Modelo DL

Este segundo modelo fue el desarrollado por Desrochers y Laporte [9], en el que ocurre lo mismo que con el modelo MTZ, a las dos condiciones generales y a la función objetivo común, se le ha añadido una nueva restricción para la restricción de los subtours, que es diferente a la del modelo MTZ, y por lo tanto en lo que únicamente se diferencian.

5.2.1. Elementos y Variables

Los elementos de este modelo son exactamente iguales que los del modelo MTZ, ya que la única diferencia que hay entre estos dos, es que la restricción 3, es decir, la de eliminación de subtour, que cambia muy levemente, manteniendo sus elementos, datos, variables asociadas y variables de decisión [9].

Elemento	Tipo	Variables asociadas	Datos
Ciudades/Nodos	Índice i,j=1…N	Ui(posición ciudad i), entero	-
Tramos/Arcos	(i,j)	-	Cij (coste de i a j), entero

Tabla 2. Elementos del modelo DL

Fuente: Elaboración propia

Elementos y variables asociadas:

- **N:** Será el número total de ciudades que existen en el problema.
- **Tramos:** Son los recorridos que se formarán en ir de una ciudad i a una ciudad j.
- **Cij:** Son los costes que supone ir de una ciudad i una ciudad j, recogidos en la matriz de datos.
- **Ui:** Variable entera que implica el número de ciudades visitadas hasta la ciudad i. Básicamente esta variable indica cual es la posición de visita de la ciudad de partida i.

Variables de decisión:

- **Xij:** Es una variable binaria (valor 1 o 0), la cual dependerá de si el TRAMOS(i,j) es seleccionado o no para encontrar el óptimo.

$$\boldsymbol{X_{ij}} = \begin{Bmatrix} 1, si\ se\ visita\ la\ ciudad\ j\ inmediatamente\ después\ de\ la\ ciudad\ i \\ 0, si\ no\ se\ visita\ la\ ciudad\ j\ inmediatamente\ después\ de\ la\ ciudad\ i \end{Bmatrix}$$

5.2.2. Restricciones

Las restricciones son iguales que las del modelo MTZ, excepto la restricción 3 que cambia:

R1) $\sum_j^n x_{ij} = 1$; $\forall i = 1 \dots n;\ \ i \neq j$

R2) $\sum_i^n x_{ij} = 1$; $\forall j = 1 \dots n;\ \ i \neq j$

R3) $u_i - u_j + (n-1)x_{ij} + (n-3)x_{ji} \leq n - 2$; $\forall j, i = 1 \dots n;\ \ i \neq j$

R4) $u_i \leq n$; $\forall i = 1 \dots n$

R5) $x_{ij} \in [0,1]$; $\forall j, i = 1 \dots n$

A continuación, pasamos a explicar las restricciones:

R1) Esta restricción es general para todos los modelos, por lo que significa lo mismo que en el modelo anterior, que exactamente una ciudad debe ser visitada inmediatamente después de la ciudad i.

R2) Esta restricción también es general para todos los modelos, y significa lo mismo que en el modelo anterior, que exactamente una ciudad debe ser visitada inmediatamente antes de la ciudad j.

R3) Esta es la restricción que diferencia al modelo, es decir, la de eliminación de los subtours. En esta restricción se hace una mezcla de la restricción 3 del modelo MTZ, y una nueva incorporación que produce la eliminación de los subtours entre dos ciudades:

Xij+Xji<=1

Esta nueva restricción, unida a la R3 anterior, forma en una sola restricción con estos dos conceptos, haciendo que el modelo pueda llegar a ser más eficiente.

R4) Esta restricción también se repite en el modelo anterior, y es para la variable que contiene la posición de las ciudades no supere al número de ciudades total.

R5) También se repite en el modelo anterior, y es simplemente para que la variable de decisión Xij sea binaria.

5.2.3. Función Objetivo

La función objetivo, vuelve a buscar la minimización de los costes totales en distancia:

$$Min \sum_{i}^{n} \sum_{j}^{n} c_{ij} x_{ij}$$

5.3. Modelo 3: Modelo intermedio

Las características de este modelo son prácticamente iguales a las del modelo MTZ y el modelo DL, donde se vuelve a repetir que la diferencia está en la ecuación de eliminación de los subtours. En este caso, se hace una variación entre el modelo MTZ y el DL, por eso lo hemos llamado Modelo Intermedio, donde utiliza la restricción 3 del modelo MTZ y añade una cuarta restricción que provoca el mismo efecto que el cambio ocurrido en el modelo DL.

5.3.1. Elementos y Variables

Los elementos de este modelo, al igual que ocurre en el modelo DL, son los mismos que los del modelo MTZ, ya que el cambio que hay es mínimo.

Elemento	Tipo	Variables asociadas	Datos
Ciudades/Nodos	Índice i,j=1…N	Ui(posición ciudad i), entero	-
Tramos/Arcos	(i,j)	-	Cij (coste de i a j), entero

Tabla 3. Elementos del modelo Intermedio

Fuente: Elaboración propia

Elementos y variables asociadas:

- **N:** Será el número total de ciudades que existen en el problema.
- **Tramos:** Son los recorridos que se formarán en ir de una ciudad i a una ciudad j.
- **Cij:** Son los costes que supone ir de una ciudad i una ciudad j, recogidos en la matriz de datos.
- **Ui:** Variable entera que implica el número de ciudades visitadas hasta la ciudad i. Básicamente esta variable indica cual es la posición de visita de la ciudad de partida i.

Variables de decisión:

- **Xij:** Es una variable binaria (valor 1 o 0), la cual dependerá de si el TRAMOS(i,j) es seleccionado o no para encontrar el óptimo.

$$\boldsymbol{X_{ij}} = \begin{Bmatrix} 1, si\ se\ visita\ la\ ciudad\ j\ inmediatamente\ después\ de\ la\ ciudad\ i \\ 0, si\ no\ se\ visita\ la\ ciudad\ j\ inmediatamente\ después\ de\ la\ ciudad\ i \end{Bmatrix}$$

5.3.2. Restricciones

En cuanto a las restricciones de este modelo, se vuelven a repetir las generales y cambia la del subtour, esta vez añadiendo simplemente una restricción nueva al modelo MTZ anterior:

R1) $\sum_j^n x_{ij} = 1 \quad ; \quad \forall i = 1 \dots n;\ \ i \neq j$

R2) $\sum_i^n x_{ij} = 1 \quad ; \quad \forall j = 1 \dots n;\ \ i \neq j$

R3) $u_i - u_j + n x_{ij} \leq n - 1 \quad ; \quad \forall j, i = 1 \dots n;\ \ i \neq j$

R4) $x_{ij} + x_{ji} \leq 1 \quad ; \quad \forall j, i = 1 \dots n;\ \ i \neq j$

R5) $u_i \leq n$; $\forall i = 1 \dots n$

R6) $x_{ij} \in [0,1]$; $\forall j, i = 1 \dots n$

En cuanto a la explicación de las restricciones anteriores:

R1) Misma restricción y explicación que en los dos otros modelos. Para indicar que se visitará una ciudad inmediatamente después de la ciudad i.

R2) Misma restricción y explicación que en los otros dos modelos. Para indicar que se ha visitado una ciudad previamente a la ciudad j.

R3) Esta restricción es la de eliminación de los subtours, y es exactamente igual que la que se usó en el modelo MTZ, con su misma explicación. Lo único que varía es en la restricción siguiente.

R4) Esta restricción es la que provoca la diferencia entre este modelo Intermedio y el modelo MTZ. También sirve para la eliminación de los subtours, y verdaderamente es la parte nueva que se añadió a la restricción 3 en el modelo DL, la que impedía los subtours entre dos ciudades.

Lo que ocurre, es que en este modelo en vez de usar esa única restricción ($u_i - u_j + (n-1)x_{ij} + (n-3)x_{ji} \leq n-2$) del modelo DL, utiliza la restricción 3 del modelo MTZ por un lado, y la otra por otro ($x_{ij} + x_{ji} \leq 1$).

R5) Esta restricción se repite en los otros dos modelos. Es para que la variable asociada Ui no supere al número de ciudades total.

R6) Finalmente, esta restricción también se repite en los otros dos modelos, y es para indicar que la variable de decisión Xij es binaria.

5.3.3. Función Objetivo

La función objetivo, vuelve a buscar la minimización de los costes totales en distancia:

$$Min \sum_{i}^{n} \sum_{j}^{n} c_{ij} x_{ij}$$

5.4. Modelo 4: Modelo de Sarin

5.4.1. Elementos y Variables

Los elementos y las variables asociadas de este modelo son las mismas que en los tres modelos anteriores, ya que son los que se usan en todos de manera generalizada. Sin embargo, las variables de decisión sí que cambian [2].

Elemento	Tipo	Variables asociadas	Datos
Ciudades/Nodos	Índice i,j=1…N	Ui(posición ciudad i), entero	-
Tramos/Arcos	(i,j)	-	Cij (coste de i a j), entero

Tabla 4. Elementos del modelo de Sarin

Fuente: Elaboración propia

Elementos y variables asociadas:

- **N:** Será el número total de ciudades que existen en el problema.
- **Tramos:** Son los recorridos que se formarán en ir de una ciudad i a una ciudad j.
- **Cij:** Son los costes que supone ir de una ciudad i una ciudad j, recogidos en la matriz de datos.

Variables de decisión:

- **Xij:** Es una variable binaria (valor 1 o 0), la cual dependerá de si el TRAMOS(i,j) es seleccionado o no para encontrar el óptimo.

$$\boldsymbol{X_{ij}} = \begin{Bmatrix} 1, si\ se\ visita\ la\ ciudad\ j\ inmediatamente\ después\ de\ la\ ciudad\ i \\ 0, si\ no\ se\ visita\ la\ ciudad\ j\ inmediatamente\ después\ de\ la\ ciudad\ i \end{Bmatrix}$$

Para que esta variable valga 1, la ciudad j tiene ser visitada inmediatamente después de la i.

- **Yij:** Es una variable binaria (valor 1 o 0), la cual dependerá del CAMINO(i,j). También valdrá 1 si la ciudad j se visita después de la ciudad i, pero no es necesario que sea inmediatamente.

$$Y_{ij} = \begin{Bmatrix} 1, si\ se\ visita\ la\ ciudad\ j\ después\ de\ la\ i\ en\ cualquier\ momento \\ 0, si\ no\ se\ visita\ la\ ciudad\ j\ después\ de\ la\ i\ en\ ningún\ momento \end{Bmatrix}$$

5.4.2. Restricciones

Las restricciones de este modelo [2] cambian más respecto a los otros tres modelos ya que se han introducido variables de decisión nueva. Las restricciones generales se mantienen, pero las relacionadas con la eliminación de los subtours cambia por completo, dejando de la siguiente manera la composición de las restricciones:

R1) $\sum_j^n x_{ij} = 1 \quad ; \quad \forall i = 1 \dots n;\ \ i \neq j$

R2) $\sum_i^n x_{ij} = 1 \quad ; \quad \forall j = 1 \dots n;\ \ i \neq j$

R3) $y_{ij} \geq x_{ij} \quad ; \quad \forall j, i = 2 \dots n;\ \ i \neq j$

R4) $y_{ij} + y_{ji} = 1 \quad ; \quad \forall j, i = 2 \dots n;\ \ i \neq j$

R5) $y_{ij} + y_{jk} + y_{ki} \leq 2 \quad ; \quad \forall i, j, k = 2 \dots n;\ \ i \neq j \neq k$

R6) $x_{ij} \in [0,1] \quad ; \quad \forall j, i = 1 \dots n$

La explicación de todas estas restricciones es la siguiente:

R1) La primera restricción es la general para todos, y se repite exactamente igual en los otros tres modelos, significando que se visitará una ciudad inmediatamente después de la ciudad i.

R2) Esta restricción también se repite en los otros tres modelos anteriores, significando que se ha visitado una ciudad previamente a la ciudad j.

R3) Esta restricción es la primera para la eliminación de los subtours. Como hemos explicado antes, la variable Yij mide si la ciudad j va después de la i en cualquier momento, y la variable Xij mide si va inmediatamente. Por lo tanto, si la ciudad j va después de la i, puede ser en cualquier momento por lo que Yij valdría 1, pero no tiene que ser inmediato, por lo que Xij podría valer 0. Por ello, es necesario que Yij sea siempre mayor o igual que Xij.

Las dos valdrían 1 en el caso de que sea inmediato, las dos valdrían cero si aún no se ha pasado por esas ciudades o si la ciudad i no precede a la j, y finalmente Yij valdría 1 y Xij valdría 0, si la ciudad i precede a la j pero no en caso inmediato.

R4) Esta es la segunda restricción para la eliminación de los subtours, donde se conoce que si Yij valiese 1, significaría que la ciudad j va después de la i. Si esto se cumple, sería imposible que la variable Yji valiese 1, es decir, que la ciudad i fuese después de la ciudad j. Por lo tanto, sólo una de estas dos variables podría valer 1, quedando reflejado que la suma de estas dos variables tiene que ser obligatoriamente 1.

R5) La quinta restricción es la última para la eliminación de los subtours, pero esta vez utiliza tres índices diferentes: i, j y k. Estos tres índices se utilizan para imponer una condición entre tres ciudades, siendo cada índice una de ellas.

Significa que teniendo tres ciudades, si la ciudad i precede a la j, es decir, Yij vale 1, ya inmediatamente sabemos que la ciudad j va después de la i. Por otro lado, si la ciudad j precede a la ciudad k, es decir, Yjk vale 1, conocemos que la ciudad i es la primera, después la ciudad j y finalmente la k. Por ello, la variable Yki tendría que ser cero, provocando así que la suma máxima de estas tres variables sea 2.

R6) Finalmente esta restricción es la misma que en los otros tres modelos, y es para indicar que la variable Xij es binaria.

5.4.3. Función Objetivo

La función objetivo es la misma que en los tres modelos anteriores, es decir, buscar el camino total que requiera la menor distancia posible:

$$Min \sum_{i}^{n} \sum_{j}^{n} c_{ij} x_{ij}$$

5.5. Comentarios sobre los modelos

Este apartado es simplemente para añadir que una de las restricciones que se repite en los tres primeros modelos, es decir, en el modelo MTZ, el modelo DL y el modelo Intermedio, podría cambiarse por una restricción que posiblemente disminuiría el tiempo de ejecución del problema, ya que trabaja con menos variables.

Esta restricción es:

$$u_i \leq n \qquad ; \quad \forall i = 1 \dots n$$

La cual se podría haber sustituido por:

$$u_i \leq n - 1 \qquad ; \quad \forall i = 1 \dots n$$

Es un simple cambio, pero al final provoca una exploración menos, por lo que podría disminuir el tiempo de ejecución y por lo tanto ser un modelo más favorable. Nosotros hemos querido desarrollar el modelo con la primera restricción ya que es más general, pero hemos experimentado con esta segunda forma solo con uno de los problemas, el br17 en concreto, para ver si el tiempo de ejecución variaba mucho, y es prácticamente el mismo.

6. LOS PROBLEMAS

Antes de explicar y desarrollar los cuatro modelos que vamos a estudiar y con los cuales vamos a realizar todos los experimentos, vamos a detallar qué problemas son también los que vamos a usar, y ver las principales diferencias que hay entre ellos. A su vez explicaremos el formato en el que se encuentran, y qué datos nos están proporcionando.

6.1. Introducción de los problemas

Estos enunciados a nuestros problemas están en un formato de archivo de texto (".txt"), y todos simulan a localizaciones con ciertas ciudades dentro de ellas, las cuales querrían ser visitadas ahorrando el mayor coste posible. Dentro de ellos se encuentran los siguientes datos relevantes:

- NAME: Nombre del problema, que nos orienta de su localización y dimensión.
- TYPE: puede ser TSP o ATSP, diferencia que se explicará posteriormente.
- DIMENSION: Dimensión del problema. Aquí se detalla la dimensión exacta de cada problema, en este caso la dimensión dependerá del número de ciudades que tenga cada problema.
- DATOS COSTES: Todos los costes que existen en ir de una ciudad a otra. Esta parte del enunciado, nos la dan o bien en forma de matrices completas, o bien en matrices diagonales inferiores.

```
NAME:  br17
TYPE: ATSP
COMMENT: 17 city problem (Repetto)
DIMENSION:  17
EDGE_WEIGHT_TYPE: EXPLICIT
EDGE_WEIGHT_FORMAT: FULL_MATRIX
EDGE_WEIGHT_SECTION
 9999    3    5   48   48    8    8    5    5    3    3    0    3    5    8    8    5

    3 9999    3   48   48    8    8    5    5    0    0    3    0    3    8    8    5

 5    3 9999   72   72   48   48   24   24    3    3    5    3    0   48   48   24

48   48   74 9999    0    6    6   12   12   48   48   48   48   74    6    6   12
48   48   74    0 9999    6    6   12   12   48   48   48   48   74    6    6   12
    8    8   50    6    6 9999    0    8    8    8    8    8    8   50    0    0
    8
    8    8   50    6    6    0 9999    8    8    8    8    8    8   50    0    0
    8
    5    5   26   12   12    8    8 9999    0    5    5    5    5   26    8    8
    0
    5    5   26   12   12    8    8    0 9999    5    5    5    5   26    8    8
    0
    3    0    3   48   48    8    8    5    5 9999    0    3    0    3    8    8
    5
    3    0    3   48   48    8    8    5    5    0 9999    3    0    3    8    8
    5
    0    3    5   48   48    8    8    5    5    3    3 9999    3    5    8    8
    5
    3    0    3   48   48    8    8    5    5    0    0    3 9999    3    8    8
    5
    5    3    0   72   72   48   48   24   24    3    3    5    3 9999   48   48
   24
    8    8   50    6    6    0    0    8    8    8    8    8    8   50 9999    0
    8
    8    8   50    6    6    0    0    8    8    8    8    8    8   50    0 9999
    8
    5    5   26   12   12    8    8    0    0    5    5    5    5   26    8    8
 9999
EOF
```

Fig.10 Ejemplo de problema en fichero de texto

Fuente: Elaboración propia

Los costes se encuentran en matrices, porque es la forma más sencilla de recopilar todos los datos. Cada fila de la matriz representa a una ciudad, y a su vez cada columna de la matriz representa también a cada ciudad. Para explicar mejor cómo se disponen los costes, usamos una matriz de ejemplo:

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix}$$

En esta matriz que sería de una dimensión 3x3, es decir, que tendríamos 3 ciudades, cada elemento de la matriz contiene los costes que hay entre la ciudad de su fila con la ciudad de su columna. De esta manera, por ejemplo, el elemento a12, contendría el coste que hay de ir de la ciudad 1 a la ciudad 2, o bien el elemento a32, es el coste que hay en ir de la ciudad 3 a la ciudad 2.

A su vez hay que explicar, que los elementos diagonales, es decir, a11, a22 y a33, tendrían un valor igual a cero, ya que estos costes serían los que hay de ir desde una ciudad a ella misma, por lo tanto, esos costes son nulos.

Por otro lado, hay una sola diferencia entre unos problemas u otros, sin tener en cuenta las diferencias de dimensión y costes propios, y esa diferencia es la que crea los dos grupos distintos de problemas con los que vamos a trabajar: Problemas simétricos y problemas asimétricos.

6.2. Problemas simétricos

Cuando hablamos de problemas simétricos o también llamados los de tipo TSP, nos estamos refiriendo a aquellos en los que el coste de ir desde una ciudad i a una ciudad j es el mismo que el coste de ir desde la ciudad j a la ciudad i.

Esto se ve reflejado en nuestra matriz de costes, en que los elementos opuestos tienen el mismo valor, es decir, es nuestra matriz de ejemplo anterior el valor del coste a12 será el mismo que el valor del coste a21. Por ello, a la hora del enunciado de los problemas, solo nos dan la matriz diagonal inferior para no repetir lo mismo en la matriz diagonal superior, ahorrando así un gran espacio.

```
NAME: gr17
TYPE: TSP
COMMENT: 17-city problem (Groetschel)
DIMENSION: 17
EDGE_WEIGHT_TYPE: EXPLICIT
EDGE_WEIGHT_FORMAT: LOWER_DIAG_ROW
EDGE_WEIGHT_SECTION
 0
 633 0
 257 390 0
  91 661 228 0
 412 227 169 383 0
 150 488 112 120 267 0
  80 572 196 77  351 63 0
 134 530 154 105 309 34 29 0
 259 555 372 175 338 264 232 249 0
 505 289 262 476 196 360 444 402 495 0
 353 282 110 324 61 208 292 250 352 154 0
 324 638 437 240 421 329 297 314 95 578 435 0
 70 567 191 27 346 83 47 68 189 439 287 254 0
 211 466 74 182 243 105 150 108 326 336 184 391 145 0
 268 420 53 239 199 123 207 165 383 240 140 448 202 57 0
 246 745 472 237 528 364 332 349 202 685 542 157 289 426 483 0
 121 518 142 84 297 35 29 36 236 390 238 301 55 96 153 336 0
EOF
```

Fig.11 Fichero de texto con enunciado de un TSP simétrico

Fuente: Elaboración Propia

Se puede comprobar con este ejemplo de problema simétrico, que únicamente nos dan la matriz diagonal inferior, ya que la otra parte de la matriz es la misma, pero de forma traspuesta, por lo que así ahorramos espacio. A su vez vemos que la diagonal está compuesta de ceros como hemos explicado antes, debido a que el coste de una ciudad a sí misma es nulo.

A continuación, expondremos una tabla con todos los problemas simétricos que vamos a resolver, con su nombre y dimensión:

Nombre del problema	Dimensión (número de ciudades)
Bays29	29
Brazil58	58
Gr17	17
Gr21	21
Gr24	24
Gr48	48
Hk48	48

Tabla 5. Listado de problemas simétricos

Fuente: Elaboración propia

6.3. Problemas asimétricos

Los problemas asimétricos o también conocidos como los de tipo ATSP, son aquellos en los que el coste de ir desde una ciudad i hasta la ciudad j, cambia respecto al ir desde la j hacia la i. Por lo tanto, en nuestra matriz de ejemplo, el elemento a13, que es el coste que requiere ir de la ciudad 1 a la 3, es distinto que el valor del elemento a31, que es el coste de ir de la ciudad 3 a la ciudad 1.

Por este motivo, a la hora del enunciado es necesario que la matriz de coste esté completa, es decir, nos den todos los elementos de ella ya que cada posición de la matriz contiene un valor único y necesario.

Sin embargo, vuelva a ocurrir que la diagonal es cero, como hemos explicado anteriormente. Un ejemplo de este tipo de enunciado es la figura 8, definido en la introducción a modo de ejemplo. El listado de problemas asimétricos con los que vamos a trabajar se muestran en la tabla inferior:

Nombre del problema	Dimensión (número de ciudades)
Br17	17
Ft53	53
Ft70	70
Ftv33	34
Ftv35	36
Ftv38	39
Ftv44	45
Ftv47	48
Ftv55	56
Ftv64	65
Ftv70	71
P43	43
Ry48p	48

Tabla 6. Listado de problemas asimétricos

Fuente: Elaboración propia

7. LINGO

Lingo será la herramienta con la que haremos la resolución de todos nuestros problemas definidos anteriormente, en la que implementaremos los distintos modelos descritos en el apartado anterior, pero con su propio lenguaje.

Antes de describir cómo quedan nuestros cuatro modelos en Lingo y cómo ha sido la experimentación, es importante definir esta herramienta para entender mejor cómo funciona y cómo va a complementarse con el problema TSP.

7.1. Introducción a Lingo

Lingo es una herramienta informática utilizada para resolver modelos de optimización tanto lineales como no lineales automáticamente y de forma rápida, ofreciendo unos resultados muy detallados de manera que nos permitirá interpretarlos de forma cómoda.

Para programar en Lingo es necesario utilizar su propio lenguaje, el cual describiremos a continuación, así como el formato necesario para escribir el modelo [14].

El formato principal que debe tener cualquier modelo en Lingo consta de 5 partes:

- **Encabezado:**

Sirve para indicarle a Lingo que vamos a escribir un modelo.

"MODEL: "

- **Función Objetivo:**

Aquí se especifica si lo que buscamos es maximizar o minimizar los valores de los costes de todas las variables. Además, hay que añadir para evitar errores, poner punto y coma (;) al final de cada línea, de modo que así indicaremos al programa que en la siguiente línea se va a escribir otra indicación diferente a la anterior. Por ejemplo:

" MIN = 5*X4 + 2*X3 + X2 + 6*X1"

- **Restricciones del problema:**

Al igual que todos los modelos de optimización, existen una serie de restricciones que se deben cumplir para conseguir el objetivo y hay que reflejarlas manteniendo la siguiente forma:

"X1 + 2*X2 = 2 ;"

"2*X3 + X4 <= 1 ;"

- **Restricciones asociadas al tipo de variable:**
  - En Lingo, por defecto, las variables son continuas (≥ 0)
  - Las variables enteras se escriben como @GIN(X1)

- Las variables binarias se escriben como @BIN(X2);
- Las variables libres se escriben como @FREE(X3);

- **Fin:**

Para indicarle a Lingo que hemos finalizado la escritura del modelo.

"END"

- **Detalles:**

Además, hay que tener en cuenta varios detalles:

- Para insertar un comentario en la propia programación del modelo hay que preceder la línea a comentar con el símbolo "¡" y el final con un punto y coma.

- Lingo ignora las líneas en blanco.

-La longitud máxima admitida en una misma línea es de 512 caracteres

Ejemplo de modelo en Lingo

A continuación, ilustramos un ejemplo de modelo matemático sencillo, para pasarlo posteriormente a Lingo utilizando su lenguaje propio y su estructura previamente explicada.

Max X1 + Y1 – X2

s.a.

X1 + Y1 >= 2

X2 + Y1 <= 4

X1 + X2 – Y1 <=3

X1, X2 >= 0 , entera

Y1 binaria

Fig. 12 Ejemplo de Modelo matemático

Fuente: Elaboración propia

El modelo consta de una función objetivo, tres restricciones y tres variables, de las cuales dos son variables enteras y una es una variable binaria, es decir, que puede valer 1 o 0. En Lingo se pondrá el tipo de las variables como restricciones, quedando así:

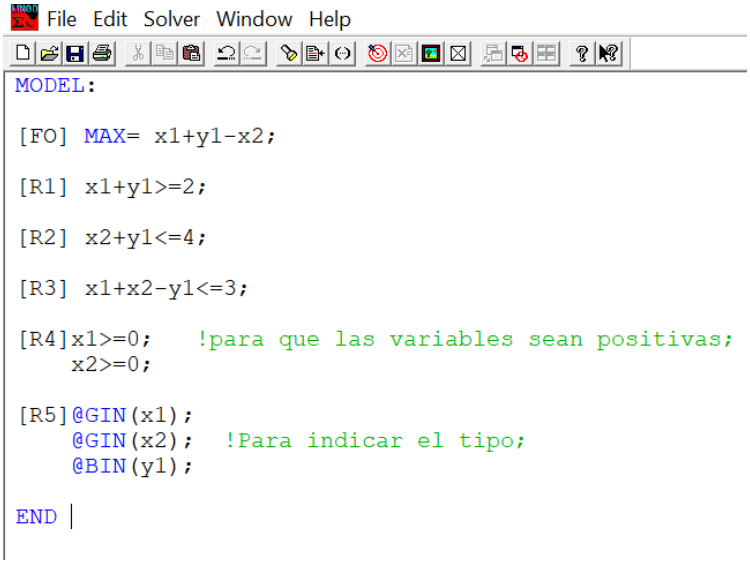


Fig.13 Ejemplo de modelo matemático simple en Lingo

Fuente: Elaboración propia

7.2. Ampliación de Lingo

En este subapartado vamos a detallar un poco más en la herramienta Lingo, así como explicar funciones de la librería, explicar los conjuntos y lectura de datos numéricos o detallar en cómo nos devuelve Lingo la solución óptima [14].

7.2.1. Los conjuntos

Cuando queremos expresar un modelo de forma simplificada, es necesario el uso de conjuntos. Los conjuntos están relacionados directamente con los elementos del problema, y podrían distinguirse dos tipos:

- **Conjuntos primitivos:** Recogen los elementos conjunto del problema y sus variables asociadas independientes o relacionadas con elementos no conjuntos.
- **Conjuntos derivados:** Recogen las variables asociadas relacionadas entre elementos conjunto. Están formados, por tanto, por los elementos primitivos ya definidos.

Los elementos concepto o unitarios simples no son necesario definirlos y sus variables asociadas independientes se definen directamente en la sección de valores (DATA). LINGO posee dos secciones para la definición de los conjuntos:

- **Sección SET:** Se definen los nombres y número de elementos de los conjuntos.
- **Sección DATA:** Se definen los valores de las variables asociadas. Es posible también definir aquí el número de elementos del conjunto.

Por ejemplo, si tuviéramos un problema con 4 ciudades o nodos a visitar, habría que especificar esa dimensión en el apartado SETS. A su vez, habría que indicar cuáles son las variables asociadas al elemento ciudades. En el modelo MTZ, por ejemplo, una variable asociada al elemento "ciudades" podría ser la variable Ui, la que indica la posición de cada ciudad, y quedaría así:

Ciudades/1…4/: Ui;

Arcos (ciudades ,ciudades): Cij, Xij;

ENDSETS

Aquí también se observa cómo se ha creado el elemento "arcos", que serán los tramos entre una ciudad y otra, por ello, se escribe "(ciudad,ciudad)", porque los datos de esos tramos estarán definidos en una matriz de dimensión del número de ciudades a visitar. Además, se observa que se definen las dos variables relacionadas con los arcos, es decir, la variable Cij, que verdaderamente serán datos, y la variable de decisión Xij, que será la binaria.

Ahora para dar valores numéricos a esos datos relacionados con las ciudades, entraremos en la sección DATA. Verdaderamente aquí solo es necesario proporcionarles datos a los costes Cij, los cuales estarán definidos en nuestro fichero de texto en nuestro caso. De forma general se podrían definir así:

DATA:

C= 1, 2, 3, 4, 5,6,7,8,9,10,11,12,13,14,15,16;

ENDDATA

7.2.2. Restricciones

- Funciones:

A la hora de representar restricciones más complejas en Lingo usamos funciones como @FOR o @SUM:

Función	**Uso**
@FOR	*@FOR* es usado par generar restricciones sobre los miembros de un conjunto.
@SUM	*@SUM* calcula la suma de una expresión sobre los miembros de un conjunto.

Tabla. 7 Funciones para restricciones en Lingo

Fuente: Resolución de Problemas, Librerías de optimización

Donde la función que desempeñan se podría simular a:

$$\forall i: \quad \sum_{j=1}^{12} x_{ij} \le A_i$$

@FOR @SUM

Fig. 14 Aplicación de las funciones Lingo

Fuente: Resolución de problemas, Librerías de optimización

Y para implantarlo en Lingo, se debería seguir esta forma:

@funcion (nombre_conjunto [(lista índices) [| calificador condicional]]: expresión);

La lista de índices y el calificador condicional son opcionales. La expresión es la restricción que se impone.

- <u>Signos:</u>

El formato utilizado en Lingo en las restricciones para sus signos es el siguiente:

#EQ#	=	*#And#*	y
#NE#	≠	*#Or#*	o
#GE#	≥		
#GT#	>		
#LT#	<		
#LE#	≤		

Fig. 15 Aplicación de los signos en Lingo

Fuente: Resolución de problemas, Librerías de optimización

<u>Ejemplo de Restricción compleja en Lingo:</u>

Suponiendo que v(i) es un dato del elemento Objetos(i), la restricción sería:

$$s.a. \quad \sum_{i=1}^{10} v_i \alpha_i \le V \longrightarrow$$

@SUM(Objetos(i): v(i)*Alpha(i)) <= VM;

7.2.3. Interfaz con el usuario

Una vez que ejecutamos la aplicación Lingo, lo primero que nos aparece es la siguiente ventana:

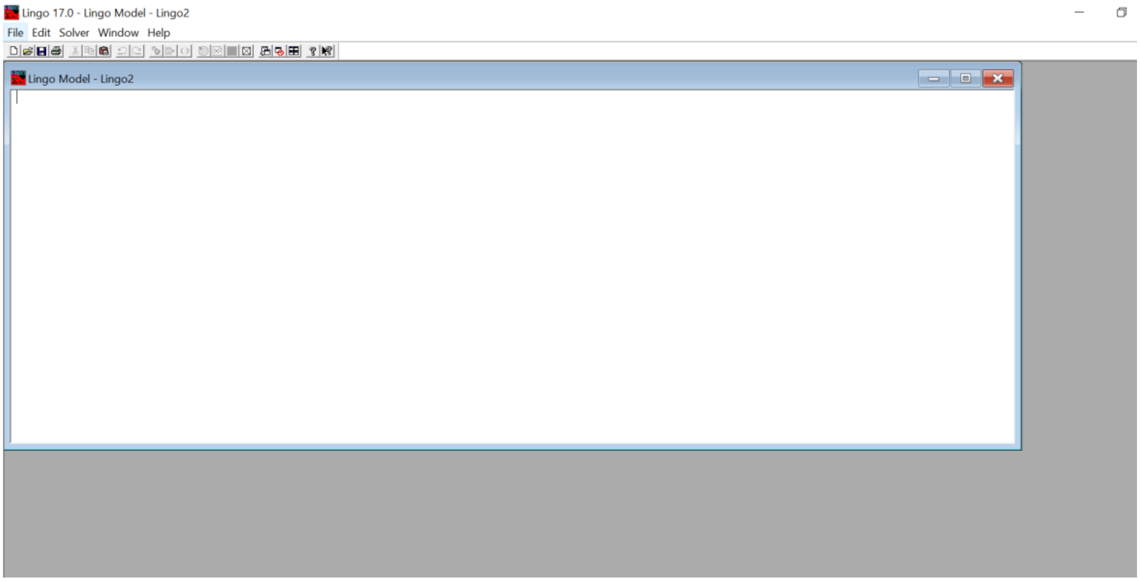


Fig.16 Inicio de Lingo, interfaz con el usuario

Fuente: Elaboración propia

En esta página en blanco es donde comenzaremos a escribir el modelo en lenguaje Lingo que queremos solucionar, utilizando todo lo explicado anteriormente. Si queremos escribir un nuevo código solo hay que pulsar en la pestaña de la hoja en blanco, situada en la barra superior a la izquierda.

En esa barra superior se encuentran varias herramientas que nos pueden servir de ayuda, pero la que vamos a explicar ya que es la más importante, es la que se necesita pulsar para que el modelo de Lingo se pueda solucionar:

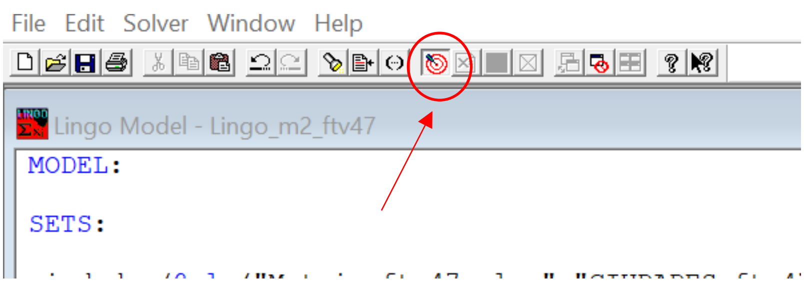


Fig.17 Ejecución en Lingo, interfaz con el usuario

Fuente: Elaboración propia

Una vez definido el modelo con todos sus encabezados necesarios, sus elementos y variables definidas, sus restricciones y su función objetivo, solo habría que pulsar la pestaña de arriba, y Lingo nos daría en un tiempo de ejecución desconocido que dependerá de la complejidad del problema, la solución de este. Esto ocurrirá en caso de que todo esté perfectamente descrito y de que Lingo haya podido encontrar un óptimo, en caso contrario, saldrá un mensaje de error.

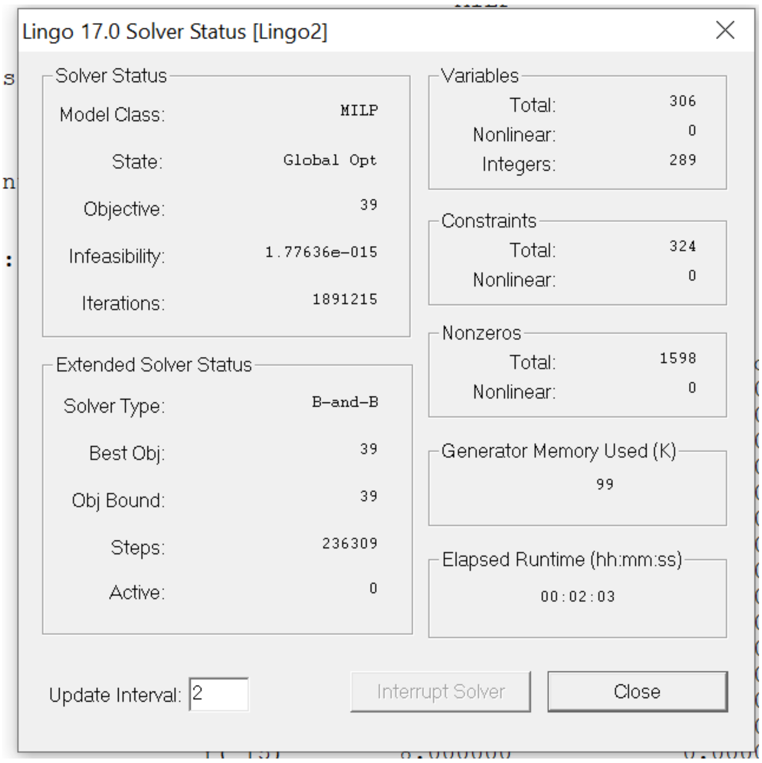


Fig.18 Solución de Lingo, interfaz con el usuario

Fuente: Elaboración propia

Si todo es correcto, como hemos explicado, y Lingo encuentra para el modelo un óptimo, nos aparecerá esta pestaña a modo de resumen donde contendrá los datos principales de la resolución. Por ejemplo:

- El tiempo de ejecución (Elapsed Runtime)
- El óptimo encontrado (Objetive)
- Número total de variables utilizadas (Variable- total)
- El tipo de modelo, en nuestro caso, MILP (Mixer Integer Linear Problems) (Solver Status- Model Class)
- El tipo de método de resolución (Extended Solver Status- Solver Type) que usa, en nuestro caso un Branch and Bound.

Seguidamente, después de esta pestaña, se nos abrirá otra que contiene el desarrollo de todo el problema. Aquí se detallarán todas las variables obtenidas en el modelo y sus valores, así como diversos datos que nos pueden ayudar a hacer un buen estudio del modelo.

7.3. Conversión de los ficheros (".txt") a Lingo

Los enunciados de los diferentes problemas que vamos a desarrollar en esta experimentación se encuentran en ficheros de texto, es decir, en documentos de tipo ".txt". Por lo tanto, necesitaremos alguna aplicación o método para realizar la conversión de dichos enunciados a nuestra herramienta de resolución Lingo.

Gracias a la forma en la que nos enunciados vienen desarrollados en los ficheros, con los datos necesarios para la resolución de los problemas en forma de matriz, decidimos que la forma más sencilla para hacer el traspaso de datos sería a través de Excel. Además, esta forma es la más utilizada y estudiada en el Grado de Ingeniería de las Tecnologías Industriales, específicamente en una asignatura llamada Técnicas de Optimización que está ligeramente relacionada con este proyecto, y de la cual he obtenido muchos conocimientos para poder desarrollarlo.

Primero explicaremos, el traspaso de los datos desde el fichero de texto hasta la aplicación Excel, y después explicaremos cómo se utiliza Lingo para poder extraer datos de Excel de una forma muy sencilla.

7.3.1. <u>De Ficheros de Texto a Excel</u>

Para explicar este proceso, simplemente pondremos a modo de ejemplo el cambio de fichero a Excel de uno de los problemas. El ejemplo lo haremos con un problema asimétrico, ya que la matriz viene desarrollada al completo:


br17: Bloc de notas

Archivo Edición Formato Ver Ayuda

```
NAME:  br17
TYPE: ATSP
COMMENT: 17 city problem (Repetto)
DIMENSION:  17
EDGE_WEIGHT_TYPE: EXPLICIT
EDGE_WEIGHT_FORMAT: FULL_MATRIX
EDGE_WEIGHT_SECTION
 9999    3    5   48   48    8    8    5    5    3    3    0    3    5    8    8    5
    3 9999    3   48   48    8    8    5    5    0    0    3    0    3    8    8    5
    5    3 9999   72   72   48   48   24   24    3    3    5    3    0   48   48   24
   48   48   74 9999    0    6    6   12   12   48   48   48   48   74    6    6   12
   48   48   74    0 9999    6    6   12   12   48   48   48   48   74    6    6   12
    8    8   50    6    6 9999    0    8    8    8    8    8    8   50    0    0    8
    8    8   50    6    6    0 9999    8    8    8    8    8    8   50    0    0    8
    5    5   26   12   12    8    8 9999    0    5    5    5    5   26    8    8    0
    5    5   26   12   12    8    8    0 9999    5    5    5    5   26    8    8    0
    3    0    3   48   48    8    8    5    5 9999    0    3    0    3    8    8    5
    3    0    3   48   48    8    8    5    5    0 9999    3    0    3    8    8    5
    0    3    5   48   48    8    8    5    5    3    3 9999    3    5    8    8    5
    3    0    3   48   48    8    8    5    5    0    0    3 9999    3    8    8    5
    5    3    0   72   72   48   48   24   24    3    3    5    3 9999   48   48   24
    8    8   50    6    6    0    0    8    8    8    8    8    8   50 9999    0    8
    8    8   50    6    6    0    0    8    8    8    8    8    8   50    0 9999    8
    5    5   26   12   12    8    8    0    0    5    5    5    5   26    8    8 9999
EOF
```

Fig.19 Paso 1 para la conversión de enunciados

Fuente: Elaboración propia

Se abre el fichero de texto del problema a resolver. Los datos que vamos a necesitar de aquí serán tanto la dimensión del problema como todos los costes que hay en la matriz. Una vez que el fichero de texto esté medianamente ordenado (en algunos ficheros venía la matriz peor definida), está listo para traspasarlo a Excel:

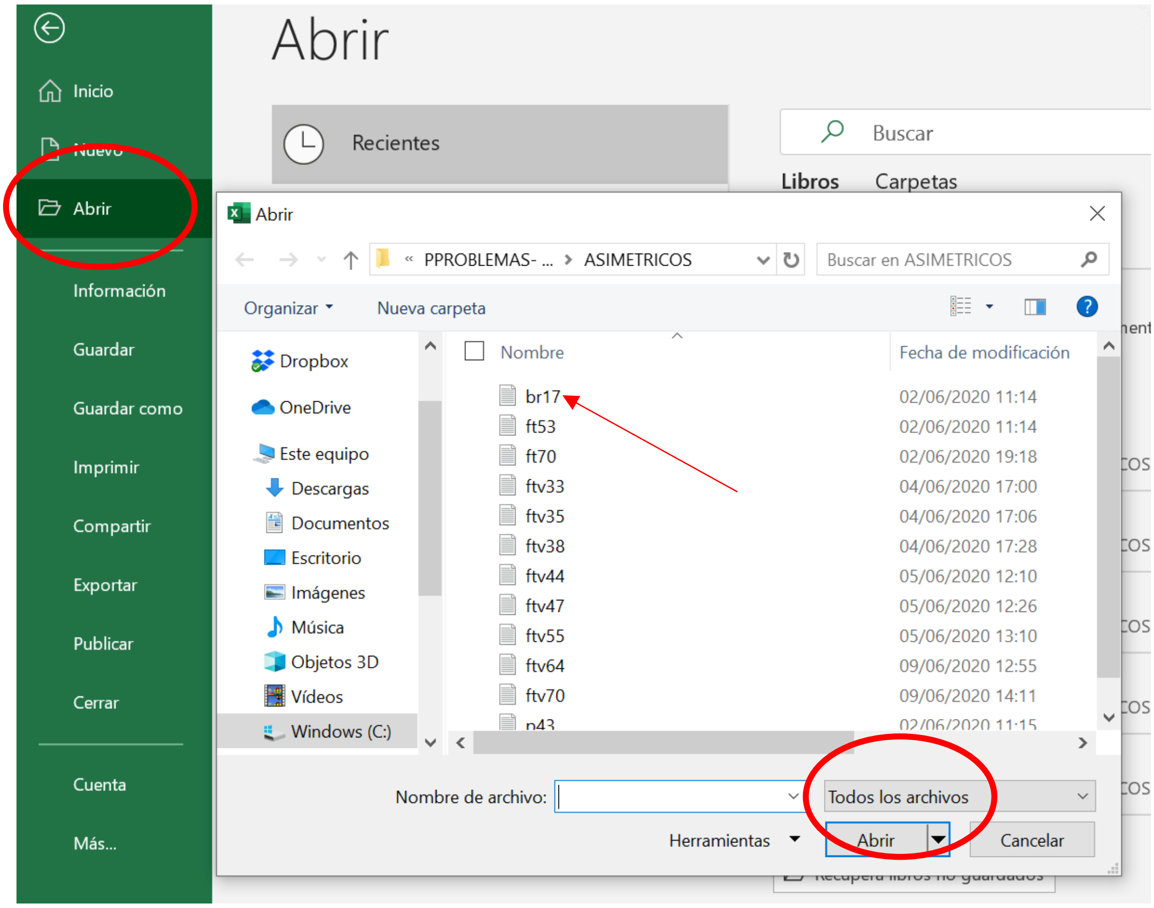


Fig.20 Paso 2 para la conversión de enunciados

Fuente: Elaboración propia

Abrimos Excel, y lo primero que hacemos es pulsar en la barra superior a la izquierda, en la pestaña Archivo-Abrir (1). Después le daremos a examinar para poder buscar más fácilmente nuestro fichero de texto, siempre y cuando hayamos modificado previamente que se puedan abrir "Todos los archivos" (2). Si no se hace esto, solo nos dejará abrir documentos tipo Excel.

Abrimos nuestro fichero de texto, y nos aparecerá lo siguiente:

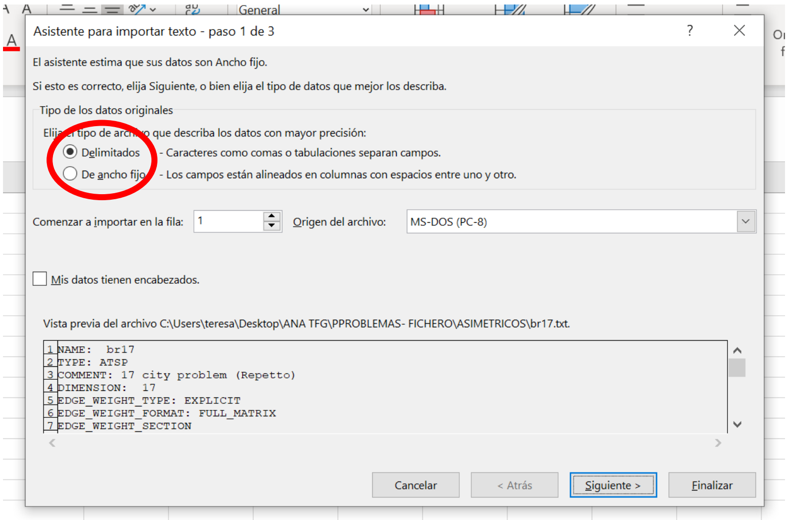


Fig.21 Paso 3 para la conversión de enunciados

Fuente: Elaboración propia

Al abrir el fichero en Excel, lo primero que nos aparece es esta pestaña. En ella tendremos que pulsar la opción de “Delimitados”, y a continuación pulsar “Siguiente”. Lo segundo que nos aparecerá será la próxima imagen adjunta, en la que debemos de indicar cual será nuestro delimitador. Esto se refiere a qué será lo que provoque la separación en las distintas celdas de Excel. En nuestro caso, cada dato en el fichero viene separado simplemente por un espacio, por lo que nuestro delimitador será el “Espacio”:

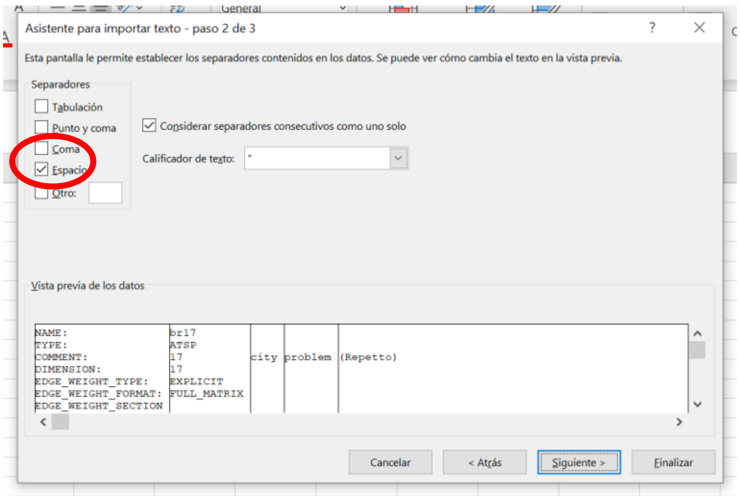


Fig.22 Paso 4 para la conversión de enunciados

Fuente: Elaboración propia

Posteriormente, habría que finalizar en esa misma pestaña, y automáticamente el fichero de texto se colocará en formato Excel. Cada dato del problema será una casilla del Excel, lo que nos permitirá trabajar con él.

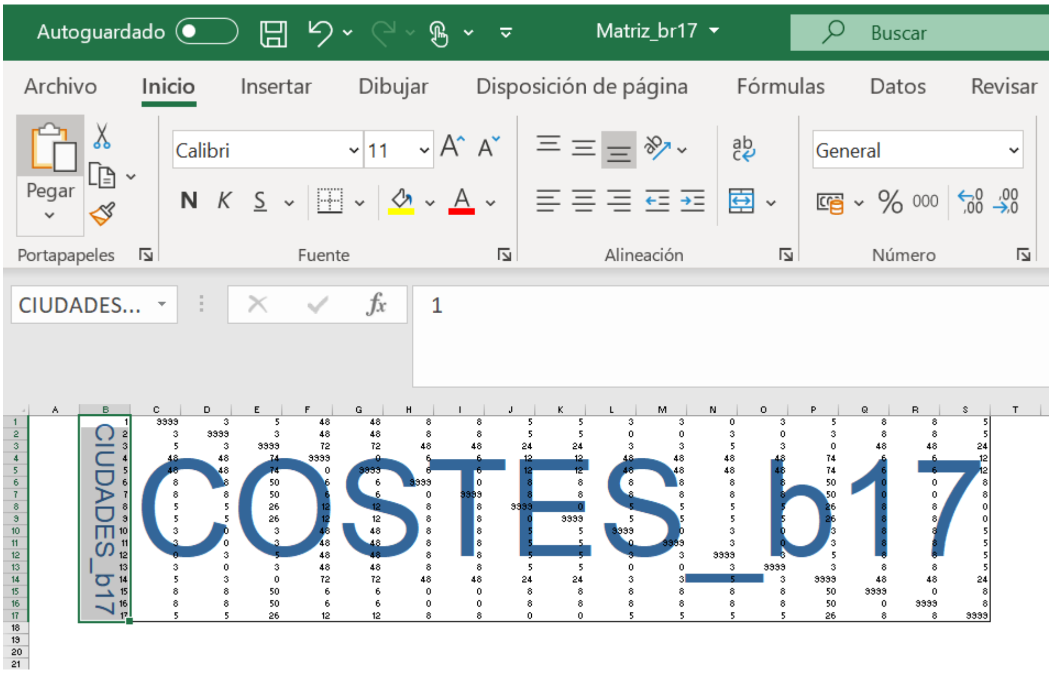


Fig.23 Paso 5 para la conversión de enunciados

Fuente: Elaboración propia

Para finalizar el traspaso de los datos del Fichero de texto a Excel, únicamente sería necesario proporcionarles nombres dentro de Excel a esas ciudades y esos costes que se necesitan en Lingo. Esto es, dar nombre a la matriz completa de costes la cual la hemos llamado a todas "COSTES_NombreProblema" (en nuestro ejemplo será COSTES_br17), y por otro lado, crear un vector "CIUDADES_NombrePorblema", que contendrá a todas las ciudades del problema (en nuestro ejemplo de llamará "CIUDADES_br17"). Todo esto se puede comprobar en la imagen colocada en la parte superior.

7.3.2. De Excel a Lingo

Ahora, necesitamos que Lingo sea capaz de leer todos esos datos de Excel para que poder resolver el problema. Para ello simplemente debemos conocer una función que usa Lingo para leer datos de Excel.

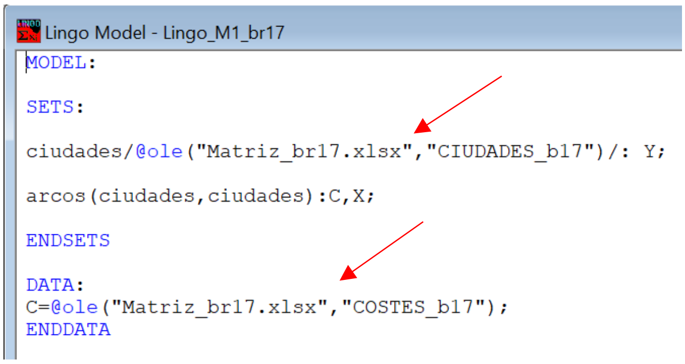
Esa función es @OLE, utilizándola junto con los nombres definidos anteriormente en el propio Excel, Lingo sería capaz de recoger todos esos datos que necesita. Para utilizarla correctamente, a la hora de describir los elementos y los conjuntos en Lingo en la sección SETS y DATA, debemos de escribir esta función seguida de los nombres de los datos de Excel. En el ejemplo de Lingo que vimos en el subapartado 7.2, los valores de los costes lo dábamos de manera:

```
DATA:

C= 1, 2, 3, 4, 5,6,7,8,9,10,11,12,13,14,15,16;

ENDDATA
```

Sin embargo a través de Excel, lo escribiremos de la siguiente manera:

```
DATA:
C=@ole("Matriz_NombreProblema.xlsx","COSTES_NombreProblema");
ENDDATA
```

Esto es, porque a todos los documentos de Excel que contienen las matrices de datos los hemos llamado Matriz_NombreProblema, y a los datos de costes los hemos llamado COSTES_NombreProblema. De igual manera se utilizará el vector de las ciudades, quedando finalmente en Lingo el siguiente código:


Lingo Model - Lingo_M1_br17

```
MODEL:

SETS:

ciudades/@ole("Matriz_br17.xlsx","CIUDADES_b17")/: Y;

arcos(ciudades,ciudades):C,X;

ENDSETS

DATA:
C=@ole("Matriz_br17.xlsx","COSTES_b17");
ENDDATA
```

Fig.24 Paso 6 para la conversión de enunciados

Fuente: Elaboración propia

7.4. Los Modelos finales en Lingo

A continuación, mostraremos cómo han quedado finalmente nuestros cuatro modelos en Lingo.

Los modelos que se van a desarrollar están escritos para el primer problema desarrollado, es decir, el problema br17. De esta manera es más fácil entender la estructura, y simplemente habría que cambiar el nombre del problema por otro que se quiera resolver.

- Donde pone “Matriz_br17.xlxs”, iría el nombre del Excel que contiene el enunciado del problema.
- Donde pone “CIUDADES1_br17”, iría el vector ciudades explicado anteriormente, que tiene la dimensión del problema.
- Donde pone “COSTES_br17” irían los datos de los costes entre las distintas ciudades del problema.
- Y donde pone n, sería la dimensión del problema, que en este caso sería un 17.

Lo único a cambiar en el resto de problemas es poner el nombre del problema a resolver, en lugar de br17.

7.4.1. Modelo MTZ

File Edit Solver Window Help

```
MODEL:

SETS:

ciudades/@ole("Matriz_br17.xlsx","CIUDADES_br17")/: Y;

arcos(ciudades,ciudades):C,X;

ENDSETS

DATA:
C=@ole("Matriz_br17.xlsx","COSTES_br17");
ENDDATA

MIN=@SUM(arcos(i,j):C(i,j)*X(i,j));

@FOR(arcos(i,j):@BIN(X));

@FOR(ciudades(j):@SUM(ciudades(i)|i#ne#j:X(i,j))=1);

@FOR(ciudades(i):@SUM(ciudades(j)|j#ne#i:X(i,j))=1 );

@FOR(arcos(i,j)|j#ne#1:Y(i)-Y(j) + n*X(i,j)<=n-1);

@FOR(ciudades(i): y(i)<=17);


END
```

Fig.25 Modelo MTZ en Lingo

Fuente: Elaboración propia

7.4.2. Modelo DL

File Edit Solver Window Help

```
MODEL:

SETS:

ciudades/@ole("Matriz_br17.xlsx","CIUDADES_br17")/: Y;

arcos(ciudades,ciudades):C,X;

ENDSETS

DATA:
C=@ole("Matriz_br17.xlsx","COSTES_br17");
ENDDATA

MIN=@SUM(arcos(i,j):C(i,j)*X(i,j));

@FOR(arcos(i,j):@BIN(X));

@FOR(ciudades(j):@SUM(ciudades(i)|i#ne#j:X(i,j))=1);

@FOR(ciudades(i):@SUM(ciudades(j)|j#ne#i:X(i,j))=1 );

@FOR(arcos(i,j)|j#ne#1:Y(i)-Y(j) + (17-1)*X(i,j)+X(j,i)*(17-3)<=17-2);

@FOR(ciudades(i): y(i)<=17);


END
```

Fig.26 Modelo DL en Lingo

Fuente: Elaboración propia

7.4.3. Modelo Intermedio

File Edit Solver Window Help

```
MODEL:

SETS:

ciudades/@ole("Matriz_br17.xlsx","CIUDADES_b17")/: Y;

arcos(ciudades,ciudades):C,X;

ENDSETS

DATA:
C=@ole("Matriz_br17.xlsx","COSTES_b17");
ENDDATA

MIN=@SUM(arcos(i,j):C(i,j)*X(i,j));

@FOR(arcos(i,j):@BIN(X));

@FOR(ciudades(j):@SUM(ciudades(i)|i#ne#j:X(i,j))=1);

@FOR(ciudades(i):@SUM(ciudades(j)|j#ne#i:X(i,j))=1 );

@FOR(arcos(i,j)|j#ne#1:Y(i)-Y(j) + 17*X(i,j)<=16);

@FOR(arcos(i,j)|i#LT#j:X(i,j)+X(j,i) <=1);

@FOR(ciudades(i): y(i)<=17);


END
```

Fig.27 Modelo Intermedio en Lingo

Fuente: Elaboración propia

7.4.4. Modelo de Sarin

File Edit Solver Window Help

```
MODEL:

SETS:

ciudades/@ole("Matriz_br17.xlsx","CIUDADES_b17")/;

arcos(ciudades,ciudades):C,X,Y;

ENDSETS

DATA:
C=@ole("Matriz_br17.xlsx","COSTES_b17");
ENDDATA

MIN=@SUM(arcos(i,j):C(i,j)*X(i,j));

@FOR(arcos(i,j):@BIN(X)); !R6;

@FOR(ciudades(j):@SUM(ciudades(i)|i#ne#j:X(i,j))=1); !R1;

@FOR(ciudades(i):@SUM(ciudades(j)|j#ne#i:X(i,j))=1 ); !R2;

@FOR(arcos(i,j)|j#ne#i#AND#i#GT#1#AND#j#GT#1:Y(i,j)>=X(i,j));  !R3;

@FOR(arcos(i,j)|j#ne#i#AND#i#GT#1#AND#j#GT#1:Y(i,j)+Y(j,i)=1);  !R4;

@FOR(ciudades(i):@for(ciudades(j):@for(ciudades(k)|i#ne#j#AND#i#GT#1#AND#j#GT#1#AND#k#GT#1#AND#i#ne#k#AND#j#ne#k:Y(i,j)+Y(j,k)+Y(k,i)<=2))); !R5;



END
```

Fig.28 Modelo de Sarin en Lingo

Fuente: Elaboración propia

8. EXPERIMENTACIÓN

En este apartado desarrollaremos la experimentación llevada a cabo, así como la exposición de todos nuestros resultados con los diferentes modelos y en los diferentes problemas, para poder compararlos de una forma sencilla.

Primero mostraremos la batería de problemas que hemos ejecutado, y seguidamente comenzaremos a comparar los resultados finales. Es importante mencionar que la ejecución de los problemas se ha parado al llevar una hora ya que, en caso de no hacerlo, podríamos pasarnos días esperando a que un problema se resuelva. Esta experimentación se podría hacer en cualquier periodo de tiempo determinado, siempre y cuando todos tengan el mismo, para poder hacer una comparación coherente. Debido a esto, no podríamos decir que estamos obteniendo datos totalmente exactos en todos los problemas, ya que muchos de ellos, al interrumpirlos no llegarían al óptimo real, pero si a una aproximación.

Para la comparación de los diferentes modelos vamos a separar primero los problemas en simétricos y asimétricos, para ver cómo se comportan los modelos con ambos tipos de problemas. Una vez especificado en el tipo de problema en el que nos encontramos, comenzaremos a comparar por tablas, basándonos en cuatro criterios por separado:

- El error:
  Esto lo calcularemos con la siguiente fórmula del error, para poder ver qué margen de error tenemos con la solución obtenida por nosotros y el verdadero óptimo del problema.

$$error(\%) = \frac{óptimo\ modelo - óptimo\ real}{óptimo\ real} x100$$

- Tiempo de ejecución:
  Para esta comparación, simplemente miraremos el tiempo de ejecución de cada problema. En caso de que el problema haya sido interrumpido debido a la superación de la hora, pondremos una “I” de interrumpido (lo cual supondrá que el tiempo de ejecución ha sido una hora), en caso contrario, pondremos los minutos y segundos exactos de ejecución.

- Número de óptimos conseguidos:
  En esta tabla escribiremos el número de óptimos totales, es decir, dentro del periodo de la hora, en cuántos problemas distintos han podido cada modelo conseguir su óptimo. Si el modelo MTZ tiene un 4, significa que ha sido capaz de obtener el óptimo real de 4 de los problemas que ha ejecutado.

- Lower Bound:
  Esta parte de la comparación la haremos solo para dos problemas de cada tipo, ya que supone una nueva experimentación con Lingo cambiando la restricción de la variable Xij binaria, por una que indique ser menor o igual a 1 (Xij<=1). Esto nos proporcionará el Lower Bound o límite inferior. Cuanto más próximo al óptimo, mejor.

Una vez realizadas las comparativas por cada punto, se hará una comparativa general para decidir cuál de los cuatro modelos podría ser el mejor para cada tipo de problema.

Antes de comenzar las comparaciones y exposición de datos, explicaremos los pasos que hemos llevado a cabo para la realización de estos experimentos:

- PASO 1:
  El primer paso fue el desarrollo de los 4 modelos matemáticos en Lingo, haciéndolos de una forma generalizada, es decir, para ningún problema en concreto.

- PASO 2:
  Seguidamente, se realizó la conversión de los enunciados de los ficheros de texto a Lingo siguiendo los pasos explicados en el subapartado 7.3. Para ello traspasamos los datos del fichero a Excel, y después, en la propia experimentación de cada problema en los modelos genéricos, se leerían desde Lingo los datos necesarios de los documentos de Excel creados.

- PASO 3:
  Aquí comienza la experimentación de todos los problemas. Para ello, hemos ido resolviendo toda la batería de problemas en un mismo modelo, y después hacerlo de la misma manera con el resto de modelos. Este proceso se tiene que hacer con el enunciado en Excel abierto del problema a resolver, y modificando el modelo en Lingo correctamente con los datos del problema para que pueda proceder a su lectura, como se explica en el subapartado 7.3.

- PASO 4:
  Finalmente, una vez modificado el modelo de Lingo de acuerdo con el problema a resolver y su Excel de datos abierto simultáneamente, se procede a la ejecución del problema. Como hemos dicho antes, el tiempo máximo de ejecución será de 1 hora, es decir, si el problema lleva ejecutándose más de una hora, se interrumpiría para tener todos los datos en un mismo periodo de tiempo.

- PASO 4:
  Este último paso es la interpretación de los datos una vez resuelto o interrumpido el problema con su modelo. Los datos que debemos recoger de cada problema terminado, será el óptimo/mejor solución encontrada y su tiempo de ejecución. Pueden ocurrir dos casos:

  - Que se haya interrumpido la ejecución a la hora, por lo que en la hoja de solución nos aparecerá un mensaje que pondrá "`Feasible solution found`", es decir, el óptimo encontrado no tiene por qué ser el óptimo del problema, es solo la mejor solución encontrada en ese rango de tiempo. Ese valor encontrado podría ser el óptimo, pero como Lingo no lo sabe, determina que es la mejor solución encontrada, ya que verdaderamente necesitaría más tiempo para seguir inspeccionando otros resultados.

- Que el problema haya terminado de resolverse en menos de 1 hora. Entonces, en la hoja de solución de Lingo aparecerá el mensaje "`Global optimal solution found`", es decir, verifica que se ha encontrado el óptimo del problema ejecutado.

8.1. Experimentación Problemas Asimétricos

Para comenzar a detallar en la comparación de este tipo de problemas, pondremos una tabla con todos los distintos problemas que hemos llevado a cabo, así como su dimensión:

Nombre del problema	Dimensión (número de ciudades)
Br17	17
Ft53	53
Ft70	70
Ftv33	34
Ftv35	36
Ftv38	39
Ftv44	45
Ftv47	48
Ftv55	56
Ftv64	65
Ftv70	71
P43	43
Ry48p	48

Tabla 8. Listado de problemas asimétricos

Fuente: Elaboración propia

Tenemos en total una batería de 13 problemas asimétricos a resolver. Como explicamos en un subapartado anterior, los problemas simétricos serán los que tengan una matriz de costes completa, donde los elementos Cij serán distintos a los elementos Cji.

A continuación, desarrollaremos las cuatro comparativas de los cuatro modelos distintos que tenemos. Para ello mostraremos una tabla con la recogida de todos los datos significantes de la experimentación, y en amarillo pondremos los datos relevantes finales, es decir, con los que se podría resumir esta tabla en cuanto a mejora de los modelos.

- <u>Comparación según el error:</u>

Nombre	Dim.	Óptimo	MTZ		DL		Intermedio		Sarin	
-	-	-	Solución	% e	Solución	% e	Solución	% e	Solución	% e
br17	17	39	39	0	39	0	39	0	39	0
ft53	53	6905	6905	0	6905	0	6905	0	7916	14,6416
ft70	70	38673	38874	0,51974	38732	0,1526	38673	0	-	-
ftv33	34	1286	1286	0	1286	0	1286	0	1286	0
ftv35	36	1473	1473	0	1473	0	1473	0	1473	0
fvt38	39	1530	1530	0	1530	0	1530	0	1542	0,78431
ftv44	45	1613	1613	0	1613	0	1613	0	1625	0,74396
ftv47	48	1776	1776	0	1776	0	1776	0	1822	2,59009
ftv55	56	1608	1608	0	1608	0	1608	0	-	-
ftv64	65	1839	1839	0	1839	0	1839	0	-	-
ftv70	71	1950	1950	0	1950	0	1950	0	-	-
p43	43	5620	5650	0,53381	5637	0,3025	5621	0,01779	5933	5,5694
ry48p	48	14422	14429	0,04854	14422	0	14422	0	14446	0,16641
%error total			1,102087243		0,455052322		0,017793594		24,49573071	

Tabla 9. Comparación por error Problemas Asimétricos

Fuente: Elaboración propia

En esta tabla hemos definido los problemas a resolver, con su dimensión (número de ciudades del problema) y el óptimo real que tienen. Seguidamente, hemos colocado las diferentes soluciones obtenidas en los cuatro modelos, así como el error que tienen estos modelos en cuanto a cercanía a su óptimo real de la forma explicada en el apartado anterior.

El porcentaje de error total es el sumatorio de todos los porcentajes de error que han surgido al utilizar ese método en todos los problemas. Como se puede observar en la tabla, los tres primeros modelos tienen un porcentaje de error mínimo frente al 24% de error que tiene el modelo de Sarin.

Sin embargo, dentro de esos tres modelos, el que parece mejor en cuanto a porcentaje de error del óptimo es el modelo Intermedio, con un 0,018% de error, lo cual es mínimo y verifica que mediante él podríamos obtener datos prácticamente exactos.

- Comparación según el tiempo de ejecución:

Nombre	Dim.	MTZ	DL	Intermedio	Sarin	Modelo más rápido
-	-	Tiempo de Ejecución	Tiempo de Ejecución	Tiempo de Ejecución	Tiempo de Ejecución	
br17	17	02:03	17:11	01:14	I	Intermedio
ft53	53	I	06:01	20:22	I	DL
ft70	70	I	I	00:28	I	Intermedio
ftv33	34	00:02	00:03	00:02	08:38	MTZ-Interm.
ftv35	36	00:06	00:07	00:05	38:25	Intermedio
fvt38	39	00:06	00:09	00:07	I	MTZ
ftv44	45	00:09	00:10	00:04	I	Intermedio
ftv47	48	00:18	00:36	00:18	I	MTZ-Interm.
ftv55	56	02:32	03:03	03:08	I	MTZ
ftv64	65	31:19	24:04	I	I	DL
ftv70	71	I	32:08	I	I	DL
p43	43	I	I	I	I	-
ry48p	48	I	59:15	I	I	DL

Tabla 10. Comparación por tiempo Problemas Asimétricos
Fuente: Elaboración propia

En esta comparación estamos teniendo en cuenta el tiempo de ejecución de cada problema en cada tipo de modelo. Como dijimos antes, si aparece una “I”, significa que el problema se ha tenido que interrumpir a la hora, por lo que verdaderamente no sabríamos su tiempo de ejecución final. Este periodo de tiempo podría ampliarse o reducirse, lo importante es que todos los problemas y modelos se sometan al mismo tiempo de ejecución para poder compararlos.

Hemos definido el problema, y después todos sus tiempos de ejecución en los cuatro modelos distintos. Al final, hemos indicado para cada problema cuál ha sido el modelo que más rápido ha proporcionado la mejor solución encontrada. Se puede descartar como válido en tiempo de ejecución el modelo de Sarin, ya que en ningún problema se ha alcanzado el mejor tiempo. Por otro lado, los tres primeros modelos han trabajado bastante similar, proporcionando unos tiempos de ejecución bastante rápidos y completando la mayor parte de los problemas dentro del periodo de la hora.

Sin embargo, se puede verificar que el modelo más exitoso en cuanto a tiempo de ejecución es el modelo Intermedio, ya que 6 de los 13 problemas que hay se resuelven más rápido a través del modelo Intermedio.

- Comparación por número de óptimos conseguidos:

Nombre	Dim.	Óptimo	MTZ		DL		Intermedio		Sarin	
-	-	-	Solución	conseguido	Solución	conseguido	Solución	conseguido	Solución	conseguido
br17	17	39	39	SI	39	SI	39	SI	39	SI
ft53	53	6905	6905	SI	6905	SI	6905	SI	7916	NO
ft70	70	38673	38874	NO	38732	NO	38673	SI		NO
ftv33	34	1286	1286	SI	1286	SI	1286	SI	1286	SI
ftv35	36	1473	1473	SI	1473	SI	1473	SI	1473	SI
fvt38	39	1530	1530	SI	1530	SI	1530	SI	1542	NO
ftv44	45	1613	1613	SI	1613	SI	1613	SI	1625	NO
ftv47	48	1776	1776	SI	1776	SI	1776	SI	1822	NO
ftv55	56	1608	1608	SI	1608	SI	1608	SI	-	NO
ftv64	65	1839	1839	SI	1839	SI	1839	SI	-	NO
ftv70	71	1950	1950	SI	1950	SI	1950	SI	-	NO
p43	43	5620	5650	NO	5637	NO	5621	NO	5933	NO
ry48p	48	14422	14429	NO	14422	SI	14422	SI	14446	NO
Nº óptimos conseguidos			10		11		12		3	

Tabla 11. Comparación por óptimos, Problemas Asimétricos

Fuente: Elaboración propia

Esta tercera comparación calcula cuantos óptimos ha sido capaz de proporcionar el modelo en el periodo de tiempo establecido, es decir, en una hora. Por ello, hemos definido primero el óptimo de cada problema, después la solución encontrada por cada uno de los modelos, y finalmente, si estos dos últimos valores coinciden o no. Si hay un "SI", es que el modelo ha logrado el óptimo, y en caso contrario, un "NO".

Se puede observar una vez más, que el modelo de Sarin consigue pocos óptimos, por lo que es descartado de nuestra elección como mejor modelo de resolución. También se observa que los tres primeros modelos dan muy buenos resultados, resolviéndose 10 o más problemas correctamente. Entre estos tres métodos, destaca el modelo Intermedio una vez más, con una resolución óptima de 12 de los 13 problemas que hay, casi un 93% de acierto.

- Comparación según el Lower Bound:

Problemas	br17		ftv38	
Óptimo	Óptimo=39		Óptimo =1530	
Solución/Tiempo	Solución	Tiempo	Solución	Tiempo
MTZ	2,11	0:01	1440	0:01
DL	18	0:01	1476	0:01
Intermedio	22	0:01	1476	0:01
Sarin	18	0:01	1481	0:05
Mejor Modelo	Intermedio		Sarin	

Tabla 12. Comparación por Lower Bound, Problemas Asimétricos
Fuente: Elaboración propia

Finalmente, la última comparación ha sido con el Lower Bound o límite inferior. Este dato es mejor cuanto mayor y más próximo sea al óptimo real de cada problema. Esto se debe a que cuando la cota inferior está muy cerca al óptimo, el rango de resultados que dará este modelo estará muy reducido.

Estos datos se han obtenido cambiando la restricción explicada anteriormente, y se ha resuelto solo para dos problemas asimétricos, para ver como trabajaban con esta nueva restricción. Se puede observar según la tabla adjuntada, que para el problema "br17" el que mejor lower bound que proporciona es el modelo Intermedio, pero para el problema "ftv38" el mejor modelo según el lower bound es el modelo de Sarin, proporcionando una cota de 1881, bastante cercano a su óptimo.

- Comparación final:

Para acabar con la comparación de los problemas asimétricos, después de haber recopilado todos los datos sobre error, tiempo de ejecución, óptimos y lower bound de todos los problemas con todos los modelos, se puede verificar que el modelo de Sarin no es muy eficaz a la hora de la resolución de este tipo de problemas, proporcionando datos muy negativos en comparación con los otros tres modelos.

Finalmente, comparando los tres primeros modelos, es decir, el modelo MTZ, el DL y el Intermedio, se observa que todos dan unos resultados muy semejantes y eficaces para estos problemas. Sin embargo, nos decantaríamos por el modelo Intermedio, debido a que en las cuatro comparaciones anteriores, ha sido el modelo más destacado y con mejores resultados obtenidos.

8.2. Experimentación Problemas Simétricos

Para este tipo de problemas realizaremos exactamente lo mismo que en el subapartado anterior, es decir, mostraremos la batería de problemas a resolver, y posteriormente haremos las cuatro comparativas distintas, mostrando los diferentes resultados y analizando cuál de los cuatro modelos es más eficaz para estos problemas.

Nombre del problema	Dimensión (número de ciudades)
Bays29	29
Brazil58	58
Gr17	17
Gr21	21
Gr24	24
Gr48	48
Hk48	48

Tabla 13. Listado de problemas simétricos

Fuente: Elaboración propia

- <u>Comparación según el error:</u>

Nombre	Dim.	Óptimo	MTZ		DL		Intermedio		Sarin	
-	-	-	Solución	% e	Solución	% e	Solución	% e	Solución	% e
Bays29	29	2020	2020	0	2020	0	2020	0	2020	0
Gr17	17	2085	2085	0	2085	0	2085	0	2085	0
Gr21	21	2707	2707	0	2707	0	2707	0	2707	0
Gr24	24	1272	1272	0	1272	0	1272	0	1272	0
Gr48	48	5046	5046	0	5046	0	5046	0	6555	29,9049
Hk48	48	11461	11461	0	11461	0	11626	1,43966	17515	52,8226
%error total			0		0		1,439664951		82,72749098	

Tabla 14. Comparación por error Problemas Simétricos

Fuente: Elaboración propia

Para comenzar la comparación en los problemas simétricos, comprobaremos el porcentaje de error total que tienen cada uno de los modelos para todos sus problemas.

Es el mismo procedimiento que en los problemas asimétricos, y se puede comprobar que el modelo de Sarin tiene un alto porcentaje de error. Le sigue el modelo Intermedio, donde se reduce muchísimo ese porcentaje, pero que al compararlo con los modelos MTZ y DL, con un 0% de error, también sería descartado.

Por lo tanto, según esta comparación, nos quedaríamos como modelos más eficaces tanto el modelo MTZ como con el modelo DL.

- Comparación según el tiempo de ejecución:

Nombre	Dim.	MTZ	DL	Intermedio	Sarin	Modelo más rápido
-	-	Tiempo de Ejecución	Tiempo de Ejecución	Tiempo de Ejecución	Tiempo de Ejecución	
Bays29	29	00:25	00:17	00:20	25:36	DL
Gr17	17	00:10	00:05	00:17	00:13	DL
Gr21	21	00:00	00:18	00:01	00:01	MTZ
Gr24	24	00:01	00:01	00:02	00:48	MTZ-DL
Gr48	48	I	I	I	I	-
Hk48	48	53:19	01:26	I	I	MTZ

Tabla 15. Comparación por tiempo Problemas Simétricos

Fuente: Elaboración propia

Aquí compararemos los distintos tiempos de ejecución, de la misma manera explicada en los problemas asimétricos. También se han interrumpido al transcurrir una hora de ejecución y se ha señalado con la "I" de interrumpido.

Todos los modelos han ido medianamente rápido, pero al igual que antes, destacan los modelos MTZ y DL, siendo los más rápidos en cuanto a tiempo de ejecución. Esto se debe a que tres de ellos se han resuelto más rápido con el modelo MTZ, y los otros tres se han resuelto más rápido con el modelo Dl.

El problema "gr48" ha sido el único en no poder resolverse en un tiempo igual o inferior a la hora por uno de estos dos modelos, el resto se han resuelto sin problemas.

- <u>Comparación por número de óptimos conseguidos:</u>

Nombre	Dim.	Óptimo	MTZ		DL		Intermedio		Sarin	
-	-	-	Solución	conseguido	Solución	conseguido	Solución	conseguido	Solución	conseguido
Bays29	29	2020	2020	SI	2020	SI	2020	SI	2020	SI
Gr17	17	2085	2085	SI	2085	SI	2085	SI	2085	SI
Gr21	21	2707	2707	SI	2707	SI	2707	SI	2707	SI
Gr24	24	1272	1272	SI	1272	SI	1272	SI	1272	SI
Gr48	48	5046	5046	SI	5046	SI	5046	SI	6555	NO
Hk48	48	11461	11461	SI	11461	SI	11626	NO	17515	NO
Nº óptimos conseguidos			6		6		5		4	

Tabla 16. Comparación por óptimos Problemas Simétricos

Fuente: Elaboración propia

Esta tercera comparación se ha realizado según el número de óptimos conseguidos en el periodo de ejecución establecido. Es el mismo proceso que en los problemas asimétricos, es decir, contabilizar los óptimos de los problemas conseguidos por cada uno de los modelos.

Como se puede observar en la tabla superior, se vuelve a descartar el modelo de Sarin, que a pesar de que consigue resolver con óptimos más de la mitad de los problemas, los otros modelos le han superado. Le sigue el modelo Intermedio, que logra casi un 80% de óptimos, pero los modelos MTZ y DL vuelven a ser los más eficientes, con un 100% de óptimos resueltos en el periodo de tiempo.

- Comparación según el Lower Bound:

Problemas	Bays29		gr17	
Óptimo	Óptimo=2020		Óptimo =2085	
Solución/Tiempo	Solución	Tiempo	Solución	Tiempo
MTZ	1774	0:01	1655	0:01
DL	1924	0:01	1684	0:01
Intermedio	1944	0:01	1684	0:01
Sarin	1924	0:01	1684	0:01
Mejor Modelo	Intermedio		DL-Intermedio-Sarin	

Tabla 17. Comparación por tiempo Problemas Simétricos

Fuente: Elaboración propia

Esta última comparación se hace con el Lower Bound, exactamente igual que como se ha realizado con los problemas asimétricos.

Este Lower bound es el límite inferior, por lo que cuanto más próximo al óptimo del problema, mejor. Todos los modelos se han acercado más o menos igual, pero ha destacado el modelo Intermedio, siendo tanto en el problema “bays29” como en el problema “gr17” el que ha proporcionado una solución más próxima a sus respectivos óptimos.

- Comparación final:

Para la comparación final de los cuatro modelos según la resolución de seis problemas simétricos, se ha podido comprobar que los modelos que más han destacado por su inmejorable resolución, con una aproximación casi exacta a los óptimos y un tiempo de ejecución bastante bueno, han sido los modelos MTZ y DL.

Estos dos modelos han sido capaces de proporcionar un óptimo exacto del total de los problemas, con un error del 0%, además de ejecutarlos en un periodo bastante rápido.

Por otro lado, el modelo de Sarin una vez más ha sido el primer descartado, pero es importante mencionar que este modelo funciona mejor con los simétricos que con los asimétricos. En cuanto al modelo Intermedio, ha proporcionado muy buenos resultados, pero no ha destacado como ha ocurrido con los problemas asimétricos.

9. CONCLUSIÓN

Gracias a la existencia de librerías de optimización como LINGO y utilizando cuatro modelos matemáticos de optimización diferentes (MTZ, DL, Intermedio y Sarin), en este proyecto hemos analizado la resolución de una batería de problemas TSP, de diferentes tamaños y características. La conclusión más importante es que para los problemas asimétricos sería más oportuno utilizar el modelo Intermedio, y para los problemas simétricos, el modelo MTZ o el modelo DL. Estos modelos a la hora de nuestra experimentación nos han proporcionado una optimización y un tiempo de ejecución aceptables.

Este trabajo podría utilizarse en otros muchos estudios y en la resolución de diversos problemas, que como indicábamos al comienzo del trabajo, tienen muchas aplicaciones en la vida real. Por ello, por su gran demanda y aplicación, la resolución y el desarrollo de estos problemas del TSP resultan muy interesantes y de gran provecho, sobre todo mirando un horizonte en los que la tecnología permita la resolución exacta de problemas de gran tamaño. Actualmente, muchos de los rompecabezas que se pueden plantear en el día a día están relacionados con estos problemas, así como la organización de actividades o viajes, la recogida de personas o cosas, o incluso las secuencias de trabajos, por lo que supone de gran utilidad ponerlo en marcha, utilizando en caso necesario estas herramientas y modelos desarrollados.

Además, se utiliza una herramienta informática muy eficaz, Lingo, con la que puede resolverse cualquier problema de optimización. Gracias a este trabajo, hemos podido exprimir al máximo esta herramienta, profundizar en ella y ver las distintas capacidades y usos que nos puede proporcionar, observando que, en problemas de bastante complejidad y tamaño es capaz de proporcionar soluciones óptimas en cuestión de segundos o minutos.

Finalmente, me gustaría resaltar el esfuerzo de acometer un estudio experimental dentro del área de la optimización. La implementación, el chequeo de errores y obtener conclusiones de un análisis de este tipo han supuesto todo un reto para mí.

10. BIBLIOGRAFÍA

[1] London School of Economics

"Model of Solving the Travelling Salesman Problem"

Autor: H.P. Wiliams

[2] Operation Research Letters 33

"New tighter polynomial length formulations for the asymmetric traveling salesman problem with and without precedence constraints"

Autor: Subhash C. Sarin (2003)

[3] Lecture Notes in Computer Science book series (LNCS)

"The Traveling Salesman, Computational Solutions for TSP applications"

Autor: Gerhard Reinelt (1994)

[4] ProQuest Ebook Central

"The Traveling Salesman Problem: A Computational Study"

Priceton University Press (2007)

[5] Scientia et Technica Año XXII, Vol. 22, No. 4

"An Application of the MTZ method to the solution of the Travelman Salesman Problem"

Autores: Jairo Alberto Villegas Flórez, Carlos Julio Zapata Grisales¸ Gustavo Gatica

Ingeniería Industrial, Universidad Tecnológica de Pereira, Pereira, Colombia. Ingeniería Industrial, Universidad Andrés Bello, Chile

[6] Universidad de Santiago de Compostela, Máster Interuniversitario en Técnicas Estadísticas

"Cooperación en los problemas del viajante (TSP) y de rutas de vehículos (VRP): una panorámica"

Autor: Aida Calviño Martínez (2011)

[7] Tesis USM, Universidad Técnica Federico Santa María

"Formulaciones basadas en Flujos en redes para un problema de ruteo en Last Mile"

Autor: Álvaro Javier Correa Maldonado

[8] Modelos y Optimización I

"El problema del Viajante: Conceptos, Variaciones y Soluciones Alternativas"

Autor: Lic. Silvia A.Ramos

[9] European Journal of Operational Research 59

"The Traveling Salesman Problem: An overview of exact and approximate algorithms"

Autor: Gilbert Laporte (1992)

[10] Programación matemática del grado de matemáticas en la Universidad de Cádiz

"El problema del viajante de Comercio o TSP"

Autores: Azahara Carpintero, Juan de Dios Días Ramírez, Victoria Franco Otero.

[11] Universidad Autónoma del Estado de Hidalgo

URL: https://www.uaeh.edu.mx/scige/boletin/tlahuelilpan/n3/e5.html

[12] Asociación Nacional de Estudiantes de Matemáticas

URL: http://www.anem.es/web/2018/09/el-problema-del-viajante/

[13] Programación matemática del grado de matemáticas en la Universidad de Cádiz

URL: http://knuth.uca.es/moodle/mod/page/view.php?id=3417

[14] Ingeniería de Organización: Métodos Cuantitativos de Gestión

"Resolución de problemas, Librerías de Optimización Lingo"

Autor: José Manuel García Sánchez

[15] Ingeniería Industrial Online

URL: https://www.ingenieriaindustrialonline.com/investigacion-de-operaciones/problema-del-agente-viajero-tsp/

[16] Universidad Autónoma del Estado de Hidalgo

URL: https://www.uaeh.edu.mx/scige/boletin/tlahuelilpan/n3/e5.html

Escuela Técnica Superior de
INGENIERÍA DE SEVILLA