



Universidad
de Cádiz

Escuela Superior
de Ingeniería

Trabajo de Fin de Grado

Grado en Ingeniería Informática

Estudio sobre TSP (título muy provisional)

AUTOR: Carlos González Parrado

Puerto Real,
?? de ?? de 2023



Universidad
de Cádiz

Escuela Superior
de Ingeniería

Trabajo de Fin de Grado

Grado en Ingeniería Informática

Estudio sobre TSP (título muy provisional)

DIRECTORA: Elisa Guerrero Vázquez
AUTOR: Carlos González Parrado

Puerto Real,
?? de ?? de 2023

Declaración Personal de Autoría

Carlos González Parrado con DNI 49566252X, estudiante de ingeniería informática en la Escuela Superior de Ingeniería de la Universidad de Cádiz, como autor de este documento académico titulado **NO SE AUN COMO** y presentado como Trabajo Final de Grado

DECLARO QUE

Es un trabajo original, que no copio ni utilizo parte alguna sin mencionar de forma clara y precisa su origen, tanto en el cuerpo del texto como en su bibliografía y que no empleo datos de terceros sin la debida autorización, de acuerdo con la legislación vigente. Asimismo, declaro que soy plenamente consciente de que no respetar esta obligación podrá implicar la aplicación de sanciones académicas, sin perjuicio de otras actuaciones que pudieran iniciarse.

En Puerto Real, a **FECHA QUE NO SE CUAL SERA**.

Fdo: Carlos González Parrado

Resumen

Palabras clave

Abstract

Keywords

Índice

1	Introducción	5
1.1	Motivación	5
1.2	Objetivos	5
1.3	Planificación y Estructura	5
1.4	Metodología	5
1.5	Siglas	5
2	Estado del Arte	6
2.1	Definición del problema TSP	6
2.2	Historia de TSP	6
2.3	Algoritmos más utilizados en TSP	7
2.3.1	Algoritmos exactos	7
2.3.2	Algoritmos aproximados	7
2.4	Definición del problema CSP	7
2.4.1	Algoritmos empleados	7
2.4.2	Estado actual del problema	7
2.5	Ejemplos de aplicaciones de TSP	7
2.6	Instancias más usadas para benchmarking	8
2.6.1	TSPLib	8
2.6.2	TNM	8
2.6.3	World TSP y National	8
2.6.4	VLSI	8
2.6.5	Procedural	8
3	Implementación y Experimentación	10
4	Caso práctico desarrollado	11
4.1	Objetivos del proyecto	11
4.2	Diseño del controlador	11
4.2.1	MavLink y PyMavLink	11
4.3	Desarrollo del controlador	11
4.4	Condiciones experimentales y reales	12
4.5	Adecuación de los datos	12
4.6	Uso del algoritmo desarrollado	12
5	Referencias	13
6	Anexo: Documentación de la librería	15

1 Introducción

1.1 Motivación

1.2 Objetivos

1.3 Planificación y Estructura

1.4 Metodología

1.5 Siglas

2 Estado del Arte

En esta sección se expondrán tanto la definición del problema abordado en el trabajo así como los problemas derivados del mismo, y los avances hasta la actualidad que se han realizado en los métodos para resolverlo.

2.1 Definición del problema TSP

El problema del viajante ('Traveling Salesman Problem', TSP) es uno de los problemas más conocidos sobre optimización combinatoria. En dicho problema, existe un conjunto de puntos $N = \{p_1, p_2, p_3, \dots, p_n\}$. El objetivo es calcular la ruta de distancia mínima que, empezando desde un punto i , pasa por todos los demás puntos del conjunto una única vez y regresa al punto de origen.

Si el problema se prefiere definir en términos de teoría de grafos, su definición equivalente será "Dado un grafo ponderado completo, calcule el ciclo con el menor peso", donde los elementos del grafo representan los diferentes elementos del problema (puntos como vértices, aristas como caminos y los pesos como las distancias). Desde esta perspectiva, se deduce que TSP es un problema derivado del problema del ciclo Hamiltoniano y por tanto, es de tipo NP-Hard. En otras palabras, no existe ningún algoritmo en la actualidad que encuentre siempre la solución óptima con un coste computacional de orden polinomial.

2.2 Historia de TSP

Sobre el 1800 empezaron a surgir problemas matemáticos relacionados con el problema del viajante (TSP), los cuales fueron tratados por matemáticos como William Rowan Hamilton y Thomas Penyngton Kirkman. Posteriormente, en 1954, George Dantzig, Ray Fulkerson y Selmer Johnson (miembros de RAND Corporation) introdujeron el nombre de dicho problema dado por Hassler Whitney: 'Traveling Salesman Problem'. Además presentaron con ello uno de los primeros algoritmos basados en programación lineal, el algoritmo de 'planos cortantes'.

Por ello, RAND Corporation se consideró la principal organización investigadora, anunciando recompensas económicas por la resolución de algunas instancias del problema conocidas para dar mayor difusión al problema.

Dos décadas más tarde, los matemáticos Cook, Karp y Lewis desarrollaron investigaciones sobre la dificultad de resolución de problemas clasificados como NP-Hard dado que los algoritmos que los calculan no son de orden polinómico. Consecuentemente, se pudo demostrar que el TSP euclídeo pertenece al conjunto NP-Hard.

Por ello, además del diseño de algoritmos exactos para el cálculo de la solución óptima (basados en programación dinámica y branch and bound), aparecen otros

cuya prioridad no es sólo la optimización total de la solución, sino el ofrecer un resultado eficiente en un tiempo razonable. Ejemplos de este tipo de algoritmos son los basados en heurísticas, que dada una o un conjunto de soluciones iniciales aleatorias, realizan sucesivas mejoras en la misma hasta que tras cierta condición de parada, ofrecen la solución final.

En la actualidad, el algoritmo sobresaliente en cuanto a precisión es Concorde (creado por David Applegate, William Cook, Vasek Chvátal y Robert Bixby, 2006), el cual ha obtenido para instancias de hasta 85900 ciudades soluciones de gran precisión.

2.3 Algoritmos más utilizados en TSP

Concorde, CLK, LKH

2.3.1 Algoritmos exactos

2.3.2 Algoritmos aproximados

2.4 Definición del problema CSP

2.4.1 Algoritmos empleados

2.4.2 Estado actual del problema

2.5 Ejemplos de aplicaciones de TSP

Una de las aplicaciones más directas del TSP es aquella sobre cálculo de rutas entre ciudades repartidas en un mapa (para control y enrutamiento del tráfico, por ejemplo), donde se cumple la condición de que un conjunto de localizaciones deben ser visitadas una única vez. Ejemplos concretos de esta aplicación son las rutas seguidas por repartidores de correos, tours de agencias de viaje y planificación urbana.

Otras aplicaciones conocidas del problema son en el ámbito del cableado de electrodomésticos y ordenadores, ya que algunos equipos pueden describirse en términos de módulos que se unen entre sí por medio de cables. Se desea unir todos los módulos (ciudades) por medio de cables (aristas) de mínima longitud, de manera que cada módulo quede unido a otros dos. También se aplica a problemas relacionados con logística. Se supone que una máquina debe realizar n trabajos de forma secuencial y que c_{ij} es el tiempo que la máquina invertiría en pasar del trabajo i al trabajo j . Entonces de nuevo, para minimizar el tiempo invertido en cambiar de un trabajo a otro (o change-over time) es posible formular el problema en términos del TSP introduciendo un trabajo artificial de coste 0, de forma análoga al problema anterior. Otra aplicación interesante se da en cristalografía, donde

algunos experimentos consisten en tomar un gran número de medidas sobre cristales a través de un detector. Cada muestra debe montarse en un aparato que debe ser posicionado para poder hacer la medición, invirtiendo tiempo en cambiar la posición del detector desde la que disponía para realizar la última medición hasta la requerida para realizar la actual. Como en los anteriores, añadiendo una ciudad artificial de coste 0 que una la primera muestra con la última obtenemos una instancia del TSP donde las ciudades son las muestras a examinar y el coste de las aristas que las unen es el tiempo invertido por el detector para cambiar su posición.

2.6 Instancias más usadas para benchmarking

2.6.1 TSPLib

TSPLib es el principal conjunto de test usado en la mayoría de estudios debido a que, a pesar de poseer tan solo 88 instancias, éstas provienen de una diversa gama de fuentes.

2.6.2 TNM

Este conjunto de testing proviene de un tipo de generador en específico, basado en instancias de tetraedros.

2.6.3 World TSP y National

La instancia más grande abordada es aquella que contiene todas las ciudades del mundo, conocida como World TSP. Dado que cuenta con 1.9 millones de puntos cuando lo usual son 30-30000 puntos, no resulta práctico a la hora de probar algoritmos. Sin embargo, se extrajeron varios subconjuntos de tamaños más usuales de esta instancia, conocidos como 'National'.

2.6.4 VLSI

Las siglas hacen referencia a 'Very Large Scale Integration'. Este conjunto de testing proviene de circuitos integrados creados por la Bonn University (Alemania). Estas 100 instancias son de interés dado que son buenas representantes de la aplicación de TSP al diseño de microprocesadores.

2.6.5 Procedural

Además de los conjuntos mencionados anteriormente, existen generadores de instancias. Los más conocidos son RUE (Random Uniform Euclidean), NETGEN (Net-

work Generated), NETGENM (Unión de RUE y NETGEN) y TSPGEN.

3 Implementación y Experimentación

Metodología en los experimentos y el razonamiento, implementación (lenguaje, técnicas utilizadas...), casos probados y resultados.

4 Caso práctico desarrollado

4.1 Objetivos del proyecto

Dada la naturaleza del problema desarrollado en este trabajo, un tipo de caso práctico muy conveniente sería aquel que necesite de cálculo de circuitos entre varios puntos en el espacio. Gracias al profesor Andrés Yáñez Escolano, quién me puso en contacto con Ana Isabel Vázquez Mejías, investigadora del grupo de 'Señales, Sistemas y Comunicaciones Navales', obtuve un buen caso práctico para el proyecto. El proyecto trata, esencialmente, sobre la construcción de un ROV (vehículo submarino) de bajo coste. Una de las partes fundamentales del proyecto consiste en métodos de posicionamiento dinámico y su enfoque a vehículos submarinos OSH (Open-Source Hardware) y OSS (Open-Source Software).

Dado que el propósito del proyecto es disminuir el coste que supone el uso de vehículos de este tipo y los costes de producción (hardware y software) están altamente optimizados, un aporte significativo sería el de proveer un 'sistema guardián' que sea capaz de crear rutas optimizadas, seguras y con una comunicación eficiente. De esta forma, se optimiza el uso de la batería y, en caso de emergencia, permitiría actuar efectivamente para evitar daños o deterioros significativos en el vehículo.

Por ello, el objetivo de la colaboración se centra en crear una rutina que permita, a partir de una serie de puntos dados, obtener la ruta óptima y establecer una comunicación entre el controlador y el vehículo de forma que, de haber cualquier situación anómala (daños en algún componente, batería baja, por ejemplo), seguir un protocolo de acción inmediata (detención inmediata de la navegación o retorno inmediato al punto de origen). Pese a que pueda parecer una aplicación muy directa del algoritmo desarrollado, hay un aspecto crucial y es que en esta ocasión no nos situamos en un espacio euclídeo simple, sino que tendremos una fuerza afectando nuestro desplazamiento (la corriente marina), que se deberá tener en cuenta para seguir minimizando el desplazamiento total entre puntos.

4.2 Diseño del controlador

4.2.1 MavLink y PyMavLink

** Explicación del protocolo y el wrapper ¿? (No es mi parte y está con un pie en el wrapper y con el otro en la API base...)

4.3 Desarrollo del controlador

** ¿? (Ha durado 2 horas y media, contando ambas partes)

4.4 Condiciones experimentales y reales

** Datos que podemos disponer y limitaciones

4.5 Adecuación de los datos

** Transformación en los datos anteriores para que representen mejor la realidad (deformación del mapa original con la vel estimada del ROV y la corriente)

4.6 Uso del algoritmo desarrollado

** Mismo código que pruebas teóricas, describir la capa de comunicación entre Python y Matlab

5 Referencias

- Skorobohatyj Georg (Junio de 1995). Instancias de TSPLib. <http://elib.zib.de/pub/mp-testdata/tsp/tsplib/tsp/index.html>
- Cook William (Septiembre de 2021). Página dedicada a la historia, aplicaciones y líneas de investigación sobre TSP. <http://www.math.uwaterloo.ca/tsp/index.html>
- Universität Heidelberg (Marzo de 2022). Página dedicada a la historia, aplicaciones y líneas de investigación sobre TSP. <http://comopt.ifi.uni-heidelberg.de/>
- Davendra Donald y Bialic-Davendra Magdalena (Agosto de 2020). Blog dedicado a historia y métodos aplicados al problema de TSP <https://www.intechopen.com/chapters/74003>
- Definición de problema P, NP y NP-Hard [https://hmong.es/wiki/NP_\(complexity\)](https://hmong.es/wiki/NP_(complexity))
- Haider A Abdulkarim, Ibrahim F Alshammari (Agosto de 2015). Comparison of Algorithms for Solving Traveling Salesman Problem. International Journal of Engineering and Advanced Technology (IJEAT)
- Vidal Morales Miquel (Septiembre de 2012). Integración de Google Maps y TSP Solvers para la denición y resolución del TSP. Universitat de Lleida
- Brocki, L. (2010). Kohonen self-organizing map for the traveling salesperson. In Traveling Salesperson Problem, Recent Advances in Mechatronics.
- Hougardy Stefan (Septiembre de 2018). Instancias TNM. <http://www.or.uni-bonn.de/~hougardy/HardTSPInstances.html>
- Cook William (Septiembre de 2021). Instancias de National. <https://www.math.uwaterloo.ca/tsp/world/countries.html>
- Cook William (Septiembre de 2021). Instancias de VLSI. <https://www.math.uwaterloo.ca/tsp/vlsi/index.html>
- González Velarde José Luis y Ríos Mercado Roger Z.. 1Investigación de Operaciones en Acción:Aplicación del TSP en Problemas de Manufactura y Logística. Texas AM University y Universidad de Colorado.
- Luiz Usberti, Fábio (Febrero de 2022) Instancias y mediciones sobre CSP. <https://www.ic.unicamp.br/~fusberti/problems/csp/>

- Matsuura Takafumi y Kimura Takayuki (Enero de 2021). Covering Salesman Problem with Nodes and Segments. <https://www.scirp.org/journal/paperinformation.aspx?paperid=77781>
- Porto Maziero Lucas, Luiz Usberti Fábio y Cavellucci Celso (Abril de 2021). Branch-and-cut algorithms for the covering salesman problem. <https://arxiv.org/pdf/2104.01173.pdf>
- Golden Bruce, Naji-Azimi Zahra, Raghavan S., Salari Majid y Toth Paolo (Octubre de 2011). The Generalized Covering Salesman Problem. Articles in Advance, pp.1-20
- Mukherjee Anupam, Goutam Panigrahi Samir Maity y Maiti Manoranjan. Covering Solid Travelling Salesman Problem - An Algorithmic Study. BIM-TECH Business Perspectives.

6 Anexo: Documentación de la librería