

Traveling Salesman Problem

Edgar Osvaldo López Zúñiga

Introducción

Desarrollo

Sea $G = (V, E)$ un grafo y \mathbb{F} la familia de todos los ciclos Hamiltonianos en G . Para cada arista $e \in E$ se puede asignar un costo c_e . Después, se puede definir el TSP como el problema de encontrar un recorrido o ciclo Hamiltoniano en G tal que la suma de los costos de todas las aristas en el recorrido se minimice.

Si el conjunto de los nodos del grafo es $V = \{1, 2, \dots, n\}$, se puede definir una matriz de costos o distancias $C = (c_{ij})_{n \times n}$ donde c_{ij} es el costo de la arista que une al nodo i con el j dentro de G . Sin perder la generalidad, se puede asumir que G es un grafo completo, y en caso de que no lo sea, se pueden reemplazar las aristas que no forman parte del grafo con aristas de un costo muy alto.

Dependiendo de la naturaleza de G y de su matriz de costos, se puede dividir al *TSP* en dos clases principales. Si G es un grafo no dirigido (C es simétrica), el TSP es llamado el problema del agente viajero simétrico (Symmetric Traveling Salesman Problem) o STSP. Si C es no necesariamente simétrica, entonces es llamado el problema del agente viajero asimétrico (o ATSP). Como cada grafo no dirigido puede verse con un grafo dirigido con aristas del mismo costo entre dos nodos, el STSP puede verse como un caso especial de ATSP. También es posible verificar que si se duplica el número de nodos, se puede representar al ATSP como un STSP.

Existen distintas formas de representar el TSP. Una de estas formas es tratar al TSP como un problema de permutaciones, en general existen dos variantes de la representación como permutación del TSP. La primera de ellas es la representación cuadrática del problema: Sea \mathbb{P}_n la colección de todas las permutaciones del conjunto $\{1, 2, \dots, n\}$. El TSP consiste en encontrar una permutación $\pi = \{\pi(1), \pi(2), \dots, \pi(n)\}$ de \mathbb{P}_n tal que

$$c_{\pi(n)\pi(1)} + \sum_{i=1}^{n-1} c_{\pi(i)\pi(i+1)}$$

se minimiza. Para esta representación $(\pi(1), \pi(2), \dots, \pi(n))$ representa el orden en el que las ciudades son visitadas. Cada desplazamiento cíclico de π corresponde al mismo recorrido, por lo que existen n diferentes permutaciones representándolo.

Otra representación del TSP como permutación, es nombrada representación lineal de permutaciones. En esta representación, solo se consideran factibles las permutaciones cíclicas. Sea \mathbb{C}_n la colección de todas las permutaciones cíclicas de $\{1, 2, \dots, n\}$. El TSP consiste en encontrar $\sigma = (\sigma(1), \sigma(2), \dots, \sigma(n)) \in \mathbb{C}_n$ tal que

$$\sum_{i=1}^n c_{i\sigma(i)}$$

sea minimizado. En esta representación $\sigma(i)$ nos dice el nodo que sigue al nodo i en la permutación.

Variantes del TSP

Se mencionan algunas variantes del TSP que pueden ser reformuladas como un TSP realizando distintas transformaciones.

- **MAX TSP:** El objetivo consiste en encontrar un recorrido en G donde el costo total de las aristas del ciclo se maximizan. Este problema se puede transformar en el TSP al reemplazar los costos de cada arista por su inverso aditivo.

- **Bottleneck TSP:** El objetivo es encontrar un recorrido en G tal que el mayor costo de las aristas en el recorrido sea minimizado. Este problema se puede formular como un TSP donde los costos crecen exponencialmente.
- **TSP con múltiples visitas:** Se quiere encontrar un recorrido que comienza en un nodo i de G , visita cada nodo al menos una vez y regresa al nodo i de tal forma que la distancia recorrida sea minimizada. Este problema se puede transformar al TSP reemplazando los costos de las aristas con las distancias más cortas entre cada par de nodos en G .
- **Clustered TSP:** El conjunto de nodos de G se particiona en distintos grupos. El problema consiste en encontrar un recorrido de menor costo en G en donde las ciudades dentro del mismo grupo deben ser visitadas consecutivamente. El problema puede ser transformado en un TSP añadiendo un costo muy alto entre nodos de distintos grupos.
- **TSP Generalizado:** Consiste en particionar el conjunto de nodos de G en distintos grupos. Se quiere encontrar el ciclo más corto de G tal que se visite exactamente un nodo de cada grupo.
- **TSP con m agentes:** Se tiene m agentes localizados en un nodo inicial de G . Comenzando por ese nodo, cada uno de los agentes visita un subconjunto V_i de los nodos de G una sola vez y regresa al nodo inicial. El problema consiste en encontrar una partición del conjunto de nodos $V - \{1\}$ y un recorrido para cada uno de los m agentes tal que $|V_i| \geq 1$ para cada i , la unión de todos los recorridos sea el conjunto de los nodos menos el nodo inicial, y su intersección sea nula.

Problema tratado

Para este trabajo, se trataron distintas instancias del TSP simétrico. Las instancias utilizadas se obtuvieron de la TSPLIB, una librería con problemas específicos para distintas variantes del TSP y problemas relacionados.

Para cada instancia se cuenta con un archivo con una sección de especificación del problema y una sección de datos. En la sección de especificaciones se cuenta con información como el nombre de la instancia, el tipo de problema, la dimensión del problema, el tipo de aristas y su formato, el tipo de coordenadas, etc. En cuanto a la sección de los datos, el formato puede variar dependiendo de las especificaciones anteriores.

Para este trabajo, se trató con problemas específicos para el STSP donde los pesos de las aristas están dados por la distancia euclidiana entre pares de nodos. Los datos en las instancias euclidianas están dados por las coordenadas dentro del plano de cada uno de los nodos. También se trató con instancias en donde los costos son dados de forma explícita, ya sea como una matriz triangular superior, o como una matriz triangular inferior. En todos los casos considerados, se trata al problema de la misma manera, la única diferencia consiste en el preprocesamiento de los datos de entrada para cada instancia.

Instancia	Dimensión	Tipo	Óptimo
fri26	26	Matriz TI	937
kroA100	100	Euclidiana	21282
rd100	100	Euclidiana	7910
kroA200	200	Euclidiana	29368
rd400	400	Euclidiana	15281
pcb442	442	Euclidiana	50778
si535	535	Matriz TS	48450
rat575	575	Euclidiana	6773
pr1002	1002	Euclidiana	259045
si1032	1032	Matriz TS	92650

Tabla 1: Instancias utilizadas. TI es triangular inferior, TS es triangular superior.

Algoritmos Evolutivos

Existen distintas variantes de algoritmos evolutivos. La idea común detrás de ellos es la misma: dada una población de individuos y un ambiente con recursos limitados, la competencia por estos recursos causará la selección natural. Esto a su vez, causa un aumento en la aptitud de la población.

Dada una función de calidad a maximizar, se puede crear aleatoriamente un conjunto de soluciones candidatas. Después se puede aplicar esta función a ellas para elegir a los mejores individuos que formarán parte de la siguiente generación. Esta transferencia de individuos a la siguiente generación se hace mediante operadores de recombinación o de mutación. La recombinación consiste en seleccionar dos o más candidatos (llamados padres) para producir una o más nuevas soluciones (llamada hijas). La mutación se aplica a un solo candidato y resulta en uno nuevo. Así utilizando estos operadores se generan nuevos candidatos (descendencia). La aptitud de estos candidatos es evaluada y después compiten con los anteriores para tomar lugar en la siguiente generación. Este proceso se repite hasta encontrar una solución lo suficientemente buena o hasta que se alcanzan las condiciones de terminación impuestas al algoritmo.

Varias componentes de un algoritmo evolutivo son estocásticas. Durante la selección, los mejores individuos no serán elegidos de forma determinista, y los individuos peor adaptados tienen una oportunidad de formar parte del conjunto de padres. Durante la recombinación, las partes que serán heredadas de cada padre son elegidas aleatoriamente. De la misma manera con la mutación, las piezas de una solución que serán modificadas, y las piezas que las sustituirán, son elegidas aleatoriamente.

Una iteración de un algoritmo evolutivo consiste en lo siguiente:

- Seleccionar a los padres
- Recombinar parejas de padres
- Mutar a la descendencia resultante
- Seleccionar individuos para la siguiente generación

En general, todos los algoritmos evolutivos siguen un esquema similar al descrito, difiriendo principalmente en detalles técnicos. Particularmente, algunas corrientes se caracterizan por su representación. Los algoritmos genéticos se caracterizan por representar las soluciones como cadenas o arreglos evaluados en un alfabeto finito.

Las partes que conforman un algoritmo evolutivo son:

- Representación

- Función de aptitud
- Población
- Mecanismo de selección de padres
- Operadores de variedad, recombinación y mutación.
- Selección de supervivientes.

Resultados

Conclusiones

Referencias