

Titulo del paper

Carlos Cruz
Carné: 10-10168

Gabriel Gedler
Carné: 10-10

Abstract

El Problema del Agente Viajero o TSP es uno de los problemas más estudiados en la ciencia de la computación. En este trabajo se estudian las mejoras de tiempo que puede ofrecer el uso de metaheurísticas para obtener soluciones óptimas (o subóptimas).

1. Introducción

Actualmente, el Problema del Agente Viajero o TSP (abreviado del inglés *Travelling Salesman Problem*), es uno de los problemas NP-duro más estudiados en la ciencia de la computación; y, debido al gran número de aplicaciones que tiene, se busca poder encontrar soluciones óptimas lo más rápido posible. En este trabajo se estudia el uso de metaheurísticas que permitan obtener soluciones óptimas (o subóptimas bastante cercanas a la óptima) en un tiempo considerablemente menor al que utilizaría un algoritmo voraz (*greedy*).

Citan así [1].

2. Descripción del problema

El Problema del Agente Viajero consiste en, dado un conjunto de ciudades, encontrar el camino más corto posible que pase exactamente una vez por cada ciudad y regrese a la ciudad de partida. Este problema es de complejidad NP-duro (lo que implica que es de complejidad NP) lo que quiere decir que su orden de crecimiento no es polinomial si no, en este caso, el orden es de $O(n!)$.

Este es uno de los problemas de la ciencia de la computación más estudiados actualmente, por el gran número de aplicaciones que tiene, tanto dentro como fuera de la computación, como: planificación de viajes por ciudades, diseño de microchips, estudio de problemas genéticos como secuencias de ADN, entre muchos otros.

3. Trabajos previos

Trabajos previos.

4. Representación de la solución

Para modelar este problema se utiliza un grafo implícito en el que cada vértice representa una ciudad,

cada arco representa la existencia de un camino entre cada par de ciudades y el peso de cada arco representa el costo de viajar entre cada par de ciudades

Para representarlo computacionalmente se utiliza:

- numNodos: almacena el número de nodos que tiene el grafo.
- caminos: es un arreglo de enteros que almacena el número de identificación de la ciudad a la que se viajará a continuación. Esto permite conocer el sucesor de cada nodo en orden de complejidad lineal, lo cual ayuda enormemente para optimización 2-opt. También otorga la idea de un ciclo dirigido, que tiene más sentido que con una matriz de adyacencias.
- costos: es una matriz de números reales que almacena el costo de viajar entre cualquier par de ciudades (nodos). Esto permite realizar búsquedas de costos en orden constante.

5. Implementación de búsqueda local

Búsqueda local.

Algorithm 1: Nombre

Input: Descripción

Output: Descripción

```

1 Primer paso
2 Segundo
3 foreach  $i = 1 \dots n$  do
4   if Alguna condición then
5     └ Algo aquí
6 return Valor
```

Conclusiones

Aquí concluyen.

Referencias

- [1] C. So and H. So. A groundbreaking result. *Journal of Everything*, 59(2):23–37, 2005.

[2] <https://www.cs.ubc.ca/~hutter/previous-earg/EmpAlgReadingGroup/TSP-JohMcg97.pdf>

Apéndice

Bla.