Algoritmos y Estructuras de Datos III
Primer Cuatrimestre 2020
Trabajo Práctico 2

Departamento de Computación
Facultad de Ciencias Exactas y Naturales
Universidad de Buenos Aires

Introducción

En optimización combinatoria se busca la mejor manera de resolver un problema entre un conjunto de posibles soluciones al mismo. Para hablar de la mejor manera, se debe asociar cada una de las posibles soluciones a una función objetivo que nos permita evaluarlas y compararlas entre sí.

Estos problemas son de mucho interés en la práctica porque se pueden modelar situaciones reales en donde tenemos que tomar una decisión sobre una tarea y la valuación puede representar una métrica que nos resulte de interés como una ganancia asociada o un costo a pagar.

Existen una gran variedad de problemas reales que se pueden resolver mediante optimización combinatoria. En nuestro caso apuntaremos al Problema del Viajante de Comercio (TSP, por sus siglas en inglés) que consiste en encontrar un circuito hamiltoniano de costo mínimo, es decir, un circuito que visite todos los vértices exactamente una vez. Hay muchas situaciones, generalmente relacionadas a problemas logísticos, en donde se debe transportar mercadería a un conjunto de clientes. Sin ir más lejos, en estos días muchas empresas están moviendo sus operaciones a una modalidad de entrega a domicilio. Supongamos el caso donde un vendedor de pizza debe entregar pizzas con su camioneta a 25 ubicaciones en la Ciudad de Buenos Aires. Uno de sus objetivos puede ser estar en la calle el menor tiempo posible, dado que el conductor del camión cobra un salario fijo por hora. En ese caso el peso de las aristas representa el tiempo de viaje entre ambos clientes. Por otra parte, si es un vehículo propio, entonces puede querer minimizar la cantidad de nafta consumida, para reducir los costos. En este caso, las aristas tienen un costo asociado indicativo de la cantidad de nafta consumida.

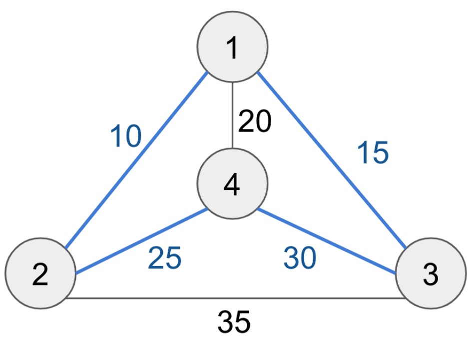

Figura 1: Ejemplo del TSP: Solución óptima en azul con costo 80.

Problema

El Problema del Viajante de Comercio (TSP) se puede definir formalmente de la siguiente manera. Sea $G = (V, E)$ un grafo completo donde cada arista $(i, j) \in E$ tiene asociado un costo c_{ij}. Se define el costo de un camino p como la suma de los costos de sus aristas $c_p = \sum_{(i,j)\in p} c_{ij}$.

El problema consiste en encontrar un circuito hamiltoniano p de costo mínimo. Sin pérdida de generalidad, vamos a pedir que el primer vértice del ciclo sea el 1. [1]

Dado que el TSP pertenece a la categoría de problemas $\mathcal{NP}$-hard, no buscaremos dar la solución óptima para cada instancia sino una de la mejor calidad posible. Sin embargo, existen algoritmos exactos que pueden resolver instancias de este problema con hasta 85900 vértices [1].

Parámetros y formato de entrada/salida

La entrada consistirá de una primera línea con dos enteros n, m indicando la cantidad de vértices y aristas del digrafo. Luego le sucederán m líneas con 3 enteros i, j, c, indicando que la arista (i, j) tiene costo c asociado.

Para la salida se debe imprimir una primera línea que contenga dos enteros n, c que indican la cantidad de vértices y el costo de la solución óptima. Luego continuará una línea con la secuencia de vértices óptima (sin incluir la repetición del primer vértice).

Entrada de ejemplo	**Posible salida esperada de ejemplo**

`4 6`
`1 2 10`
`1 3 15`
`1 4 20`
`2 3 35`
`2 4 25`
`3 4 30`
	`4 80`
	`1 2 4 3`

Enunciado

En el presente trabajo práctico se pide:

1. Describir el problema de TSP dando ejemplos y soluciones
2. Describir situaciones de la vida real que puedan modelarse utilizando TSP.
3. Diseñar e implementar para el TSP las siguientes soluciones:
   - Al menos dos heurísticas constructivas golosas.
   - Una heurística basada en árbol generador mínimo (AGM).
   - Metaheurística Tabú Search
     - Memoria basada en últimas soluciones exploradas.

[1] Notar que puede tener muchas soluciones óptimas.

- Memoria basada en estructura (aristas).

4. Para los métodos implementados, desarrollar los siguientes puntos:

   a) Explicar detalladamente el algoritmo implementado.

   b) Calcular el orden de complejidad temporal de peor caso del algoritmo.

   c) Describir (si es posible) instancias de TSP para las cuales el método no proporciona una solución óptima. Indicar (si es posible) qué tan mala puede ser la solución obtenida respecto de la solución óptima.

   d) Realizar una experimentación que permita observar la performance del algoritmo comparando la calidad de las soluciones obtenidas y los tiempos de ejecución en función de la entrada (y de otros parámetros de ser apropiado). Dentro de los casos de prueba se deben incluir también, como casos patológicos, aquellos descriptos en el ítem 4c. Para evaluar la calidad de las heurísticas con respecto a las soluciones óptimas conocidas, utilizar las instancias disponibles en http://elib.zib.de/pub/mp-testdata/tsp/tsplib/tsp/index.html cuyas soluciones óptimas se encuentran en http://elib.zib.de/pub/mp-testdata/tsp/tsplib/stsp-sol.html. En caso de que el algoritmo tenga algún parámetro configurable que determine su comportamiento (la metaheurística por ejemplo, aunque queda abierto a los demás también), se debe experimentar variando los valores de los parámetros y elegir, si es posible, la configuración que mejores resultados provea para el grupo de instancias utilizado. Presentar los resultados obtenidos mediante gráficos adecuados.

5. Una vez elegidos los mejores valores de configuración para cada heurística implementada, realizar una experimentación **sobre un conjunto nuevo de instancias** para observar la performance de los métodos comparando nuevamente la calidad de las soluciones obtenidas y los tiempos de ejecución en función del tamaño de entrada. Presentar todos los resultados obtenidos mediante gráficos adecuados y discutir al respecto de los mismos.

Fechas de entrega

- Primera entrega: Domingo 5 de Julio de 2020, hasta las 23:59 hs, enviando el trabajo por el campus virtual.

- Reentrega: Domingo 26 de Julio de 2020, hasta las 23:59 hs, enviando el trabajo por el campus virtual.

Importante: El horario es estricto. No se considerarán los correos recibidos después de la hora indicada.

Referencias

[1] David L. Applegate, Robert E. Bixby, Vašek Chvatál, and William J. Cook. The Traveling Salesman Problem: A Computational Study. Princeton University Press, 2006.