Facultad de Ciencias, UNAM
Resolviendo el problema del agente viajero usando recocido simulado

Marisol Amézcua Lopez

1. Introducción

El problema del agente viajero ha sido muy conocido en la historia de las Matemáticas y Ciencias de la Computación, se sabe que no es posible resolverlo usando un algoritmo determinista en un tiempo acotado por un polinomio, por lo que se procede a intentar resolverlo o aproximarse a una solución usando otros métodos. En este escrito se describirá una forma de realizar esto usando recocido simulado usando aceptación por umbrales.

2. El problema

El problema puede ser resumido como: Si un agente viajero quiere visitar exactamente cada ciudad de una lista de m ciudades (donde el costo de visitar desde la ciudad i a la ciudad j es de c_{ij}) y después de esto regresar a la ciudad de origen, la pregunta entonces es: ¿Cuál es la ruta que el viajero puede tomar para que el costo sea el mínimo? [1].

La importancia de este problema es que es un representante de una clase de problemas conocidos como problemas de optimización combinatoria, la cual a su vez se encuentra en una clase que en Complejidad Computacional se conoce como problemas NP-Duro. A la fecha no se ha encontrado un algoritmo que en tiempo polinomial pueda resolver el problema del agente viajero.

2.1. Características

El primer paso es modelar lo anterior de una forma matemática, para esto se modelará el conjunto de ciudades con una gráfica no dirigida, donde cada nodo representará una ciudad y la arista que las conecta contendrá el costo de viajar de la ciudad i a la j.

En este trabajo no se contemplara el viaje redondo, es decir, nos abstendremos te tomar una última arista que va desde la ciudad final a la ciudad inicial, por lo que nuestra ruta se puede ver como un camino Hamiltoniano. El costo final será la suma de los costos de cada arista usada para la ruta.

Para la implementación de el sistema en este artículo se usará una base de datos de ciudades la cual cuenta con diversas ciudades, sus coordenadas geográficas y conexiones entre las mismas, como aclaración no se cuenta con todas las posibles conexiones para cada ciudad, es decir, la base de datos no nos modela una gráfica completa. Para corregir esto en la implementación se procede a agregar nuevas aristas con pesos mucho mas grandes que cualquier arista. Para esto hacemos lo siguiente:

Sea $G(V, E)$ una gráfica no dirigida y $S \in V$ una instancia de TSP con la que trabajaremos (ciudades que entraran en el problema).

1. Primero obtenemos la distancia máxima de S, denotada por $max_d(S)$ como:

$$max_d(S) = max(\{w(u, v)|u, v \in S \wedge (u, v) \in E\}) \tag{1}$$

2. Obtenemos la distancia natural para cada par de vértices tales que su arista no esta en la base de datos. Para esto definimos la distancia natural $d(u, v)$ como:

$$d(u, v) = R * C \tag{2}$$

   Donde R es el radio del planeta Tierra en metros (aproximadamente 6,373,000) y C está dada por:

$$C = 2 * arctan(\sqrt{A}, \sqrt{1 - A}) \tag{3}$$

   y A a su vez es:

$$A = sin\left(\frac{lat(v) - lat(u)}{2}\right)^2 + \left[cos(lat(u)) * cos(lat(v)) * sin\left(\frac{lon(v) - lon(u)}{2}\right)^2\right] \tag{4}$$

   donde lat(u) es la latitud de la ciudad u y long(u) es la longitud de u.

3. Al final el peso aumentado será:

   $w(u, v) = max_d(S) * d(u, v)$

Para esta implementación utilizaremos un normalizador, el cual será el costo de la ruta mas pesada, para esto ordenamos las aristas que estan en la base de datos de mayor a menor costo y tomamos las $|S| - 1$ aristas mas pesadas, a este conjunto le llamaremos L. El normalizador de la instancia será entonces:

$$\mathcal{N}(S) = \sum_{a \in L} costo(a) \tag{5}$$

Nuestro costo para cierta ruta será:

$$f(S) = \frac{\sum_{i=2}^{k} w_S(v_{i-1}, v_i)}{\mathcal{N}(S)} \tag{6}$$

Esto nos da costos de rutas factibles (rutas que no tienen aristas artificiales) entre 0 y 1.

3. La heurística

Como se dijo en la introducción se utilizará aceptación por umbrales, la cual es una heurística que usa como base el recocido simulado de propuesto por Kirkpatrick.

4. Implementación

Definimos un normalizador el cual será el costo de la ruta mas pesada, para esto ordenamos las aristas que estan en la base de datos de mayor a menor costo y tomamos las $|S| - 1$ aristas mas pesadas, a este conjunto le llamaremos L. El normalizador de la instancia será entonces:

$$\mathcal{N}(S) = \sum_{a \in L} costo(a) \tag{7}$$

Referencias

[1] Hoffman, K. L. ,Padberg, M. & Rinaldi, G.. (2013). Traveling Salesman Problem. En Encyclopedia of Operations Research and Management Science(pp. 1537-1579). Boston, MA: Springer US.