Capítulo 1

El Problema del Agente Viajero

El Problema del Agente Viajero, o TSP por sus siglas en inglés (*Traveler Salesman Problem*), se define como sigue: dado un subconjunto de ciudades en un mapa, ¿cuál es el camino más corto que pasa por todas ellas (si acaso existe)?

1.1. La gráfica

Las ciudades en el mapa y sus conexiones estarán representadas por una gráfica ponderada $G(E, V)$, $E \subset V \times V$; con una función de peso para las aristas $w : E \longrightarrow \mathbb{R}^+$ que representará la distancia entre cada par de ciudades. La gráfica, aunque muy densa, no será completa.

Sea $S \subset V$; S será la instancia de TSP que se querrá resolver. Para hacer más sencilla la optimización de este problema, se utilizará la gráfica completa $G_S(V_S, E_S)$, donde $V_S = S$ y $E_S = \{(u, v) | u, v \in S \text{ y } u \neq v\}$, con la función de peso aumentada $w_S : E_S \longrightarrow \mathbb{R}^+$ que se define como sigue:

Definición 1.1.1 (Función de peso aumentada) *Sea $w_S : E_S \longrightarrow \mathbb{R}^+$ tal que:*

$$w_S(u, v) = \begin{cases} w(u, v) & \text{si } (u, v) \in E \\ d(u, v) \times \max_d(S) & \text{en otro caso} \end{cases}$$

donde $d(u, v)$ es la distancia natural *entre dos vértices u y v; y $\max_d(S)$ será la* distancia máxima *de S.*

La razón para definir w_S de esta manera se explicará en la sección 1.3.

1.1.1. Distancia máxima

La distancia máxima de S se define como:

Definición 1.1.2 (Distancia máxima de S) *La* distancia máxima *de S, denotada por* $\text{máx}_d(S)$, *es:*

$$\text{máx}_d(S) = \text{máx}\{w(u,v) | u,v \in S \text{ y } (u,v) \in E\}.$$

En otras palabras, $\text{máx}_d(S)$ es la distancia máxima que existe entre pares de elementos de S tales que estén conectados en G.

1.1.2. Distancia natural

Para calcular la distancia natural entre dos elementos de S, se utilizarán las coordenadas (en latitud y longitud) de las ciudades correspondientes y la fórmula que supone que el planeta Tierra es una esferea perfecta con un radio R de $6,373,000$ metros.

Las coordenadas estarán representadas en grados; la fórmula utiliza radianes, por lo que es necesario convertir grados a radianes con la ecuación 1.1.

$$rad(g) = \frac{g\pi}{180}. \tag{1.1}$$

Definición 1.1.3 (Distancia natural) *La* distancia natural *entre u y v, denotada por $d(u,v)$, está definida por:*

$$d(u,v) = R \times C,$$

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

$$C = 2 \times \arctan(\sqrt{A}, \sqrt{1-A}),$$

y A está definida por la ecuación 2:

$$A = \sin\left(\frac{lat(v) - lat(u)}{2}\right)^2 + \cos(lat(u)) \times \cos(lat(v)) \times \sin\left(\frac{lon(v) - lon(u)}{2}\right)^2. \tag{1.2}$$

La distancia natural no es exactamente la distancia que existe entre dos pares de coordenadas; como la Tierra no es una esfera perfecta, se puede dar un error de unos cuantos kilómetros. Esto no afectará al sistema.

1.2. Soluciones

Dada una instancia S de TSP, cualquier permutación de los elementos de S es una solución al problema en G_S, porque G_S es una gráfica completa. Si $S = \{v_1, v_2, v_3\}$, entonces todas las posibles soluciones de S son:

$$\begin{array}{ccc} v_1 & v_2 & v_3 \\ v_1 & v_3 & v_2 \\ v_2 & v_1 & v_3 \\ v_2 & v_3 & v_1 \\ v_3 & v_1 & v_2 \\ v_3 & v_2 & v_1 \end{array}$$

Por supuesto v_1, v_2, v_3 es una solución equivalente a v_3, v_2, v_1; esto sin embargo será irrelevante para el sistema.

Aunque todas las permutaciones de S son soluciones válidas en la gráfica G_S, esto no es cierto para G, que es donde realmente se desea resolver el problema. Clasificaremos entonces a todas las permutaciones de S en dos tipos distintos: factibles y no factibles.

Definición 1.2.1 (Solución factible) *Sea $S\{v_1, \ldots, v_k\}$ una instancia de TSP y $P = v_{\rho(1)}, v_{\rho(2)}, \ldots, v_{\rho(k)}$ una permutación de los elementos de S. P será una solución* factible *si y sólo si $(v_{\rho(i-1)}, v_{\rho(i)}) \in E$, para $i = 2, \ldots, k$.*

En otras palabras, una permutación será factible si y sólo si las aristas entre dos elementos consecutivos de la permutación existen en E (siempre existen en E_S, porque G_S es completa).

Si al menos una arista entre dos elementos consecutivos de la permutación no existe en E, la solución será no factible.

1.3. Función de costo

La optimización de TSP necesitará una función de costo que se usará para evaluar qué tan buenas (o no) son las soluciones que el sistema encuentre. Como queremos minimizar la distancia de la trayectoria que pase por todos los vértices, definimos w_s en la sección 1.1 de tal forma que las aristas que estén en E_S, pero no en E, tengan un peso *mucho mayor* que las aristas en E. Esto alentará al sistema a tratar de descartar aristas que no estén en G para entonces encontrar soluciones factibles.

Para poder comparar, aunque sea burdamente, las evaluaciones a soluciones factibles de instancias de TSP diferentes, vamos a normalizar la función de costo. Para esto necesitaremos un normalizador.

Definición 1.3.1 (Normalizador) *Para cada par no ordenado $u, v \in S$, si $(u, v) \in E$, agregamos la distancia $w(u, v)$ a una lista L y ordenamos L de mayor a menor. Sea $L' = L$ si la longitud de L es menor que $|S| - 1$; o la sublista de L con sus primeros $|S| - 1$ elementos en otro caso.*

El normalizador *de S (denotado por $\mathcal{N}(S)$) está definido como:*

$$\mathcal{N}(S) = \sum_{d \in L'} d.$$

En otras palabras, y suponiendo que $|S| = k$, el normalizador de S es la suma de las $k-1$ aristas más pesadas en E formadas de elementos de S.

Toda permutación de ciudades en S tal que sea una solución factible de TSP tendrá, por definición, una longitud menor o igual a $\mathcal{N}(S)$. El normalizador nos permitirá, como su nombre indica, normalizar nuestra función de costo de tal manera que todas las soluciones factibles de S se evalúen entre 0 y 1; y que todas las soluciones no factibles se evalúen con un valor mayor a 1.

Con el normalizador y la función de peso aumentada podemos definir la función de costo:

Definición 1.3.2 (Función de costo) *Sea $S \subset V$ una instancia de TSP; la función de costo f de una permutación $P = v_{\rho(1)}, \ldots, v_{\rho(k)}$ de los elementos de S se define como:*

$$f(P) = \frac{\sum_{i=2}^{k} w_S(v_{\rho(i-1)}, v_{\rho(i)})}{\mathcal{N}(S)}.$$

Ésta es la función que se buscará optimizar.

1.4. Vecinos

Además de la función de costo, las optimizaciones que se cubrirán en este libro suelen utilizar la estrategia de *búsqueda local* (*local searching*), lo cual implica conocer el *vecindario* de una solución. Para esto se necesita definir el *vecino* de una solución de TSP.

Definición 1.4.1 (Vecino de una solución) *Sea S una instancia de TSP y $P = v_{\rho(1)}, \ldots, v_{\rho(k)}$ y $P' = v_{\sigma(1)}, \ldots, v_{\sigma(k)}$ dos permutaciones de los elementos de S.*

Diremos que P y P' son vecinos si y sólo si existen dos índices $1 \leq s, t \leq k$, $s \neq t$, tales que:

- $v_{\rho(i)} = v_{\sigma(i)}$ *si* $i \notin \{s, t\}$, *y*
- $v_{\rho(s)} = v_{\sigma(t)}$ *y* $v_{\rho(t)} = v_{\sigma(s)}$.

En otras palabras, dos soluciones serán vecinas si intercambiando dos elementos de la primera se obtiene la segunda.