

Escala más Óptima.

Andrés Felipe Florián Quitián y Sara Gallego Rivera

Noviembre 2019

Universidad del Rosario

Matemáticas Aplicadas y Ciencias de la Computación

Repositorio: https://github.com/felipeflorian/Proyecto_Grafos

1 Introducción.

En este proyecto se realizó una propuesta para determinar las escalas favorables a la hora de realizar un vuelo. Se pretende encontrar una ruta alternativa que tenga la menor duración con intervalos de tiempo de vuelo menores al tiempo que representaría un vuelo directo. Para ello, se parte de unas nociones claves a partir del algoritmo de Dijkstra, grafo ponderado y etiquetas de un grafo. Fundamentalmente, el algoritmo de Dijkstra busca el camino mínimo entre un par de vértices en un grafo o digrafo ponderado. Con este algoritmo se encontraron las escalas más óptimas de un vuelo. Por otro lado, se creó un grafo ponderado con etiquetas, donde las etiquetas de cada vértice están dadas por el nombre de las ciudades y el peso de cada arista está determinado por el tiempo de vuelo entre dos ciudades.

Por último, se implementó el grafo ponderado de los vuelos entre ciudades con los atributos anteriormente mencionados. Para la implementación se usó

el lenguaje de programación Python y sus librerías networkx y matplotlib, con estas se realizó la representación gráfica del grafo y se usaron funciones ya implementadas en las librerías que ayudaron a la creación del grafo, de sus atributos y a la búsqueda del camino mínimo entre un par de vértices.

2 Objetivos.

2.1 General:

Encontrar la ruta óptima de vuelo en cuanto al tiempo entre un par de ciudades, con escalas máximo de siete horas.

2.2 Específicos:

- Creación de un grafo ponderado donde las etiquetas de los vértices representan las ciudades y los pesos de las aristas representan el tiempo de vuelo entre un par de ciudades.
- Implementación del algoritmo de Dijkstra para hallar el la ruta óptima o con el mínimo tiempo de vuelo entre un par de ciudades.
- Obtener una representación gráfica mediante un grafo de la ruta de vuelo óptima.

3 Caracterización de la problemática.

El programa proporciona rutas alternativas para desplazarse de una ciudad a otra, con escalas de tiempo menores o iguales a siete horas. Este proyecto

se desarrolló con el fin de dar una opción de ruta para aviones pequeños que tengan poca capacidad de combustible o personas que por sus condiciones físicas o psicológicas no puedan permanecer gran cantidad de tiempo en un avión.

4 Marco teórico.

- **Grafo ponderado:** Un grafo ponderado o grafo con peso es un grafo con etiquetas numéricas en las aristas denominadas pesos en las aristas.
- **Grafo con etiquetas:** Un grafo con etiquetas es un grafo cuyos vértices tienen asignadas etiquetas o nombres.
- **Algoritmo de Dijkstra:**
- **Input:** Un grafo o digrafo ponderado G con pesos no negativos $w(u,v)$ es el peso de la arista (u,v) . Sea $w(u,v) = \infty$ si $(u,v) \notin E(G)$.
- **Output:** $L(z)$ la distancia mínima de u a z .
- **Iteración:**
 1. $L(u) = 0$
 2. Para cada vertice $v \notin u$:
 3. $S = \emptyset$
 4. Mientras $z \notin S$
 - a. Seleccione un vértice $x \notin S$ con $L(x)$ mínimo.
 - b. $S = S \cup \{x\}$
 - c. Para todo $v \notin S$
 1. $L(v) = \min \{L(v), L(x) + w(x, v)\}$
- **Librerías:** A continuación se presentan las librerías que se utilizaran:
 1. **Networkx:** Librería de Python que sirve para crear grafos ponderados

2. Matplotlib: Librería de Python que sirve para graficar. En este proyecto se utilizará para graficar el grafo.

Por medio del lenguaje de programación Python se dio una solución para determinar las escalas favorables de un vuelo en cuanto al tiempo que este se demora en ir de un lugar a otro. Primero se creó el grafo ponderado por medio de la librería `networkx`, las etiquetas de los vértices están dadas por el nombre de la ciudad y el peso de las aristas por el tiempo de vuelo entre las ciudades.

4

Para hallar las escalas óptimas de un vuelo se usó el algoritmo de Dijkstra (networkx), para ello se da una ciudad de salida u y una ciudad de destino v . Al realizar el algoritmo se obtiene un u, v -camino mínimo, este es el que determina cuál es la escala más óptima. Por último se muestra la representación gráfica del grafo resultante después de realizar el algoritmo.

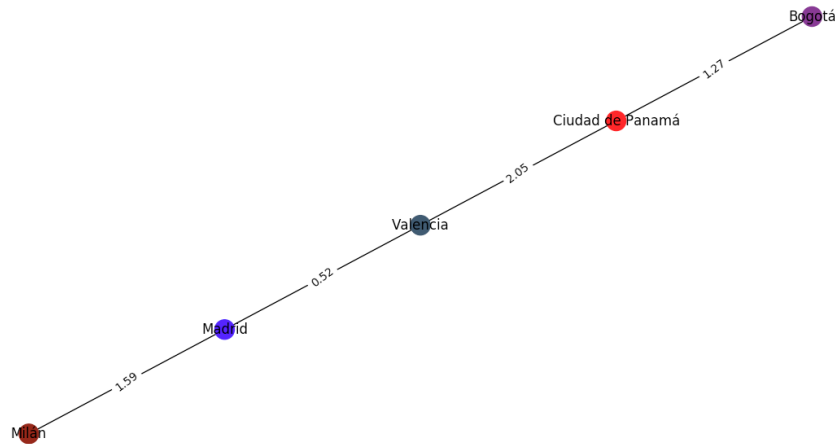


Figure 2: Grafo ejemplo de la ruta óptima entre un par de ciudades, Milán y Bogotá.

6 Solución propuesta.

Creación de un grafo ponderado con pesos no negativos, ya que los pesos representan tiempo, donde la etiqueta de cada vértice está dada por el nombre de una ciudad y el peso de cada arista está dado por el tiempo que toma un vuelo desde una ciudad hasta la otra.

Se busca encontrar el camino mínimo entre un par de vértices, lo que significa encontrar la escala de vuelo más óptima. No se consideraron vuelos directos entre dos ciudades a menos que el tiempo estimado de dicho vuelo sea de menos de siete horas.

7 Implementación de la solución y resultados obtenidos.

Se implementó un programa que pide a la persona que digite en que continente se encuentra y hacia qué continente desea ir. Al seleccionar ambos continentes se muestra una lista de países que puede seleccionar como país de salida, pertenecientes al continente seleccionado inicialmente, y una lista de países que puede visitar, pertenecientes al continente seleccionado posteriormente. De cada país hay aproximadamente tres ciudades que la persona puede escoger como destino.

Por medio del algoritmo Dijkstra se determina cuál es la ruta más corta en cuestión de tiempo desde la ciudad donde se encuentra la persona hasta la ciudad que desea ir. Al tener el resultado el programa grafica un grafo mostrando cual es la ruta óptima entre los dos destinos.

8 Conclusiones.

El programa se llevo a cabo en python por medio de su librería networkx con la cual se creo el grafo con sus respectivos atributos, y por medio de la funcion "dijkstra path" (networkx) se halló la escala o escalas más óptimas de un vuelo. Sin embargo, hacer la estructura y la implementación fue un trabajo dispendioso.

Para poder implementar los grafos ponderados primero se creó cada ciudad con su respectiva posición, fueron alrededor de 300 ciudades. Posteriormente se crearón todas las aristas con pesos menores a siete, cosa que también resultó un poco tediosa ya que eran muchas ciudades y muchas más aristas.

Inicialmente tuvimos la idea de que las escalas fueran máximo de 3 horas, sin embargo fue algo muy complicado debido a que para pasar de algunos continentes a otros hay que atravesar los océanos, y conseguir un vuelo que atraviese un océano en menos de tres horas no es algo muy viable.

El proyecto es un prototipo muy útil que proporciona rutas alternativas de vuelo, esto es muy beneficioso para personas que por sus condiciones de salud tengan limitaciones en cuanto al tiempo que puedan permanecer en un avión o para aviones pequeños que no cuenten con gran capacidad de combustible. Para futuras mejoras del proyecto se podrían agregar mas países y ciudades en consideración, así los resultados podrían llegar a ser más precisos.

9 Bibliografía.

- West, D. B. (2002). Introduction to Graph Theory (Second). Singapore: Pearson Education.
- google.com (Nov 2019). Google flights, recuperado de <https://www.google.com/flights>