



República Bolivariana de Venezuela
Universidad Centroccidental Lisandro Alvarado
Decanato De Ciencias y Tecnología
Departamento de Ingeniería Telemática
Barquisimeto-Lara



Informe

Gabriel Rojas
28.454.911
Ingeniería Telemática

1. Planteamiento del Problema

El objetivo de este proyecto es optimizar las rutas de vuelos aéreos entre diferentes aeropuertos, considerando escalas en el trayecto. En particular, se busca encontrar la ruta más eficiente desde el Aeropuerto de Caracas (Venezuela) hasta el Aeropuerto de Pekín (China), haciendo escalas en otros aeropuertos como Madrid (España) y Nueva York (EE.UU.). Este problema se puede modelar como un grafo, donde los nodos representan los aeropuertos y las aristas representan los vuelos entre ellos, junto con sus respectivas distancias o tiempos de vuelo.

3. Configuración Inicial

La configuración inicial del grafo se define mediante un diccionario que contiene los aeropuertos y sus respectivas conexiones. Cada aeropuerto es una clave en el diccionario, y su valor es otro diccionario que contiene los aeropuertos vecinos y las distancias (o tiempos de vuelo) a esos aeropuertos.

Ejemplo de Configuración

Aeropuerto Caracas tiene conexiones a:

Aeropuerto Madrid (10 horas)

Aeropuerto Nueva York (8 horas)

Aeropuerto Madrid tiene conexiones a:

Aeropuerto Caracas (10 horas)

Aeropuerto Pekín (12 horas)

4. Función Heurística

La función heurística se utiliza para estimar el costo de llegar al destino desde un nodo dado. En esta implementación, la heurística utilizada es una función simple que calcula la diferencia en la longitud de los nombres de los aeropuertos. Aunque es una heurística básica, permite al algoritmo priorizar ciertos caminos sobre otros.

5. Algoritmo A-Epsilon

El algoritmo A-Epsilon es una variante del algoritmo A*, diseñado para encontrar la ruta más corta en un grafo. La implementación sigue estos pasos:

Inicialización: Se inicializan las estructuras de datos necesarias, incluyendo la cola de prioridad `open_set`, que mantiene los nodos a explorar, y los diccionarios `g_score` y `f_score`, que almacenan los costos desde el nodo inicial a los nodos vecinos.

Búsqueda: Mientras haya nodos en `open_set`, se extrae el nodo con el menor `f_score`. Si este nodo es el objetivo, se reconstruye la ruta y se devuelve.

Exploración de Vecinos: Para cada vecino del nodo actual, se calcula un `tentative_g_score`, que es el costo acumulado hasta ese vecino. Si este costo es menor que el costo previamente registrado para ese vecino, se actualizan los registros y se añade el vecino a `open_set`.

Reconstrucción de la Ruta: Si se encuentra el nodo objetivo, se reconstruye la ruta desde el nodo objetivo hasta el nodo de inicio utilizando el diccionario `came_from`, que almacena el camino recorrido.

Ejemplo de Ejecución

Al ejecutar el algoritmo desde el "Aeropuerto Caracas" hasta el "Aeropuerto Pekín", se obtiene la ruta optimizada considerando las escalas y costos de vuelo definidos en el grafo.

Conclusiones

Este informe ha presentado la implementación de un algoritmo A-Epsilon para optimizar rutas de vuelos aéreos entre diversos aeropuertos, teniendo en cuenta escalas y tiempos de vuelo. La estructura del grafo y la función heurística permiten al algoritmo buscar rutas de manera eficiente, mientras que el parámetro epsilon ajusta la exploración en función de los costos reales y estimados.

El enfoque puede ser extendido para incluir más aeropuertos, diferentes costos de vuelo o incluso restricciones adicionales, como horarios de vuelo o disponibilidad de asientos. Esta flexibilidad hace que el algoritmo sea útil en aplicaciones prácticas de planificación de viajes aéreos.