

Implementación del Algoritmo A* sobre el Mapa de Carreteras de Rumania

Introducción

El algoritmo A* es una técnica reconocida en el ámbito de la inteligencia artificial y la informática, utilizada para identificar caminos óptimos en grafos ponderados. Este algoritmo se destaca por su eficacia al determinar la ruta más corta desde un nodo inicial hasta uno final, gracias a la aplicación de una función heurística que orienta la búsqueda de forma informada. En este proyecto, implementamos el algoritmo A* para resolver el problema de encontrar el camino más corto entre Arad y Bucarest en el mapa de carreteras de Rumania.

Descripción del Problema

El propósito de este proyecto es calcular la ruta más eficiente, en términos de distancia, entre Arad y Bucarest mediante el uso del algoritmo A*. El mapa de Rumania se modela como un grafo, donde:

- Cada ciudad se representa como un nodo.
- Cada carretera que une dos ciudades se considera una arista con un peso que indica la distancia entre ellas.

Estructura del Grafo

El grafo correspondiente al mapa de carreteras de Rumania se define utilizando un diccionario en Python. En esta estructura:

- Cada clave representa el nombre de una ciudad.
- Cada valor es una lista de tuplas que contienen las ciudades vecinas y las distancias a ellas.

Por ejemplo, la representación de Arad sería:

```
'Arad': [('Zerind', 75), ('Sibiu', 140), ('Timisoara', 118)]
```

Esto indica que Arad está conectado con:

- Zerind (75 km)
- Sibiu (140 km)
- Timisoara (118 km)

Heurística

Para guiar el algoritmo A*, implementamos una función heurística que estima la distancia directa (en línea recta) desde cada ciudad hasta Bucarest. Estas estimaciones son aproximadas, pero admisibles, ya que nunca exceden la distancia real. Esto facilita al algoritmo explorar de manera efectiva, dirigiéndose al destino sin sacrificar la precisión.

Por ejemplo:

- La heurística para Arad hacia Bucarest es de 366 km.
- La heurística para Sibiu es de 253 km.

Estas aproximaciones ayudan al algoritmo a tomar decisiones informadas en la búsqueda del camino óptimo.

Implementación del Algoritmo A*

El algoritmo A* se basa en dos funciones principales:

- **g(n)**: Representa el costo acumulado para llegar desde el nodo inicial (Arad) hasta el nodo actual.
- **h(n)**: Indica la estimación heurística de la distancia desde el nodo actual hasta el nodo objetivo (Bucarest).

La función total que el algoritmo A* utiliza para evaluar cada nodo es:

$$f(n) = g(n) + h(n)$$

Esta fórmula permite que el algoritmo elija el nodo con el menor costo total estimado, combinando el costo real hasta el momento y la estimación de lo que queda por recorrer.

Proceso del Algoritmo

A continuación se presenta el proceso paso a paso del algoritmo A* en esta implementación:

1. Inicialización:

- Se crea una lista denominada `open_list` que contiene los nodos por explorar, comenzando con el nodo inicial (Arad).
- Cada nodo se evalúa utilizando la función $f(n)$, y se selecciona el nodo con el menor valor de $f(n)$ en cada iteración.

2. Expansión de Nodos:

- En cada paso, se extrae de la lista el nodo con el menor coste $f(n)$.
- Si el nodo extraído es el destino (Bucarest), el algoritmo finaliza y devuelve la ruta óptima.

3. Actualización de Costos:

- Para cada vecino del nodo actual, se calcula el costo tentativo $g(n)$ para llegar a él.
- Si el nuevo costo tentativo es menor que el costo actualmente registrado para ese vecino, se actualiza el valor de $g(n)$ y se registra el nodo actual como su predecesor.

4. Repetición:

- El proceso se repite hasta que se localice el destino o se agoten las opciones en la lista.

Resultados

Al ejecutar el algoritmo A*, se obtienen los siguientes resultados:

- Ruta óptima de Arad a Bucarest:
 - Arad → Sibiu → Rimnicu Vilcea → Pitesti → Bucarest
- Costo total del recorrido en kilómetros:
 - La distancia total para el camino óptimo es de 418 km.

Este resultado demuestra que el algoritmo ha encontrado el camino más corto, cumpliendo con el objetivo del proyecto.

Visualización del Grafo

Para visualizar el grafo y destacar la ruta óptima hallada, empleamos las bibliotecas **NetworkX** y **Matplotlib**. Esta visualización permite observar las conexiones entre las ciudades, con las distancias entre ellas y el camino óptimo resaltado en un color distintivo.

Descripción de la Visualización

- **Nodos (Ciudades):** Cada ciudad en el mapa de Rumania se representa como un nodo en el grafo.
- **Aristas (Carreteras):** Cada carretera se ilustra como una arista ponderada, indicando la distancia entre las ciudades conectadas.
- **Ruta Óptima:** El camino más corto de Arad a Bucarest se resalta en color rojo, facilitando su identificación en el mapa visual.

Instalación de NetworkX

Trabajar con NetworkX en otro proyecto o si decides utilizarlo para construir o visualizar grafos, puedes instalarlo utilizando el siguiente comando:

pip install networkx

```
PROBLEMAS  SALIDA  CONSOLA DE DEPURACIÓN  TERMINAL  PUERTOS
PS C:\Users\corre\Desktop\Algoritmo A - Rumania Franklin Camacho> pip install networkx
Requirement already satisfied: networkx in c:\users\corre\appdata\local\programs\python\python311\lib\site-packages (3.4.2)

[notice] A new release of pip is available: 23.1.2 -> 24.2
[notice] To update, run: python.exe -m pip install --upgrade pip
PS C:\Users\corre\Desktop\Algoritmo A - Rumania Franklin Camacho> 
```