

Inteligencia Artificial 2025

Segundo Proyecto

08.abril.2025

Segundo Proyecto: Algoritmos de Búsqueda.

En este proyecto haremos una comparación del desempeño de diferentes algoritmos de búsqueda para encontrar caminos a través de laberintos. La idea es aplicar un conjunto de técnicas de IA para resolver diferentes problemas.

1. Problema 1: Generación aleatoria de laberintos.

En un grid de tamaño $M \times N$ (M filas, N columnas), desarrollar una función que genere laberintos de forma aleatoria. Para ello, deberá implementar al menos 2 de los algoritmos para encontrar el árbol de expansión mínima comentados en clase:

- algoritmo de Kruskal,
- algoritmo de Prim,
- algoritmo de Boruvka,
- método *Reverse-delete*,
- *recursive Backtracking*.

Generar visualizaciones de la construcción de estos laberintos aleatorios, comparando la construcción de ambos algoritmos implementados. Por ejemplo, puede utilizar de referencia las siguientes visualizaciones:

- <https://www.youtube.com/watch?v=h8VWRDJBf1k>,
- <https://www.youtube.com/watch?v=Kyoep91w7NE>,
- <https://www.youtube.com/watch?v=VBNZJD4wCnE>,
- <https://www.youtube.com/watch?v=vmWSnkBVvQ0>.

2. Problema 2: Solución de un laberinto aleatorio.

Dado un laberinto en un grid de tamaño 60×80 (generado por los algoritmos del Problema 1), desarrollar una función que resuelva la ruta más corta entre la entrada del laberinto: celda $A = (1, 1)$ y la salida: celda $B = (60, 80)$.

Mostrar en la visualización, las celdas exploradas por su algoritmo, así como el camino encontrado. Al final de la simulación, muestre una estadística de la longitud del camino encontrado, y de la cantidad de nodos explorados en la búsqueda. Puede elegir implementar de entre los algoritmos: BFS, DFS, *Cost Uniform Search*, A^* .



3. Problema 3: Comparación de algoritmos de búsqueda.

En un grid de tamaño 45×55 , generar diferentes escenarios con laberintos aleatorios, y comparar el desempeño de los siguientes algoritmos de búsqueda en la solución de estos laberintos:

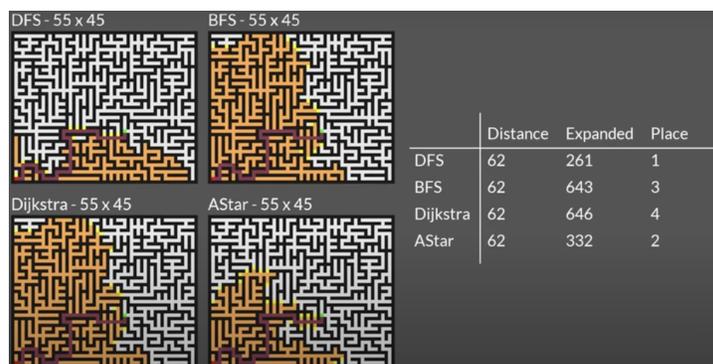
- BFS,
- DFS,
- *Cost Uniform Search* (Dijkstra),
- A^* .

Para ello, deberá generar $K = 25$ laberintos aleatorios distintos, y en cada uno de ellos deberá elegir una posición inicial A y una posición final B , (que estén al menos a una distancia Manhattan de 10 unidades). Para cada una de las K simulaciones, hacer lo siguiente:

Elaborar visualizaciones de la solución de una tabla con el conteo de nodos explorados por cada algoritmo, el tiempo de ejecución, así como de la longitud de la ruta óptima obtenida por cada uno. Ordenar los algoritmos del 1 al 4 para determinar en cada escenario cuál fue el mejor.

Finalmente, en su informe elaborar una tabla resumen que indique el ranking o desempeño promedio de los 4 algoritmos en todos los escenarios aleatorios. Puede tomar de referencia la siguiente comparación

- <https://www.youtube.com/watch?v=GC-nBgi9r0U> (ver a partir del minuto 3:44).



Evaluación: Preguntas 1 y 2: 10 puntos; Pregunta 3: 10 puntos; para un total de 20.

Fecha de Entrega: lunes 12 de mayo.