INTELIGENCIA ARTIFICIAL

Algoritmos de búsqueda

*Búsqueda A**

Encuentra la salida del laberito con el menor coste en entornos dinámicos

Belén Melián Batista
J. Marcos Moreno Vega

Curso 2025-2026

Objetivo:

Proponer, implementar y evaluar búsquedas A* para encontrar el camimo de menor coste desde el punto de entrada hasta el punto de salida de un laberinto en entornos dinámicos.

Tareas:

Además de las tareas descritas en el presente documento, los estudiantes tendrán que realizar las modificaciones que se planteen durante la corrección de la práctica.

Corrección:

Semana del 20 al 24 de octubre.

Evaluación:

Código fuente y memoria: hasta 3 puntos, si se realiza la modificación correctamente. Si el día de la corrección falta algún código o este es incorrecto, la práctica se calificará como No apta.

Modificación propuesta el día de la corrección y defensa oral de la misma: hasta 7 puntos.

Lenguaje de programación:

C++, Java o Python.

Problema del camino de mínimo coste entre los puntos de entrada y salida de un laberinto

Sea dado un laberinto como el que se muestra en la Figura 1. Los puntos de entrada y salida del laberinto están representados con las letras S y E, respectivamente. Se permite moverse a través del laberinto en dirección vertical, horizontal y diagonal, lo que incurre en un coste de 5, 5, y 7, respectivamente. El objetivo del problema a resolver es encontrar el camino de mínimo coste que va desde la casilla S hasta la casilla E. Para ello, se diseñará e implementará una Búsqueda A*. Dada una casilla n, se define la función $f(n)$ de la siguente manera:

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

donde $g(n)$ es el coste acumulado de movimiento desde el punto de entrada, S, hasta la casilla n, y $h(n)$ es una heurística para este problema. Para su definición se hará uso de la distancia de Manhattan. Por lo tanto, la función heurística para la celda n, con coordenadas en (x_n, y_n), donde (x_E, y_E) corresponde a la ubicación de la meta, E, y W es igual a 3, se define de la siguiente forma:

$$h(n) = (|x_E - x_n| + |y_E - y_n|) * W, W = 3.$$

La Figura 1 muestra la evaluación de las casillas vecinas al punto de entrada, S. Los valores representados en color verde muestran los resultados de la función de evaluación del movimiento, $g(n)$, mientras que los representados en color azul muestran los valores de la función heurística admisible, $h(n)$. Finalmente, los valores en color rojo representan la función de evaluación $f(n) = g(n) + h(n)$.

Algoritmo A*

Para realizar la implementación del algoritmo A*, se deben seguir los siguientes pasos:

1. Calcular $f(n)$, $g(n)$ y $h(n)$ para el punto de entrada al laberinto, S, que se inserta en la lista de nodos abiertos $\mathcal{A}$.

2. Repetir mientras $\mathcal{A}$ no esté vacía.

   (a) Seleccionar el nodo de menor coste $f(n)$, e insertarlo en la lista de nodos cerrados $\mathcal{C}$.

   (b) Para cada nodo vecino, realizar las siguientes acciones:

      i. Si el nodo no está ni en $\mathcal{A}$, ni en $\mathcal{C}$, su nodo padre será el nodo analizado y será insertado en $\mathcal{A}$. Realizar las acciones que corresponda.

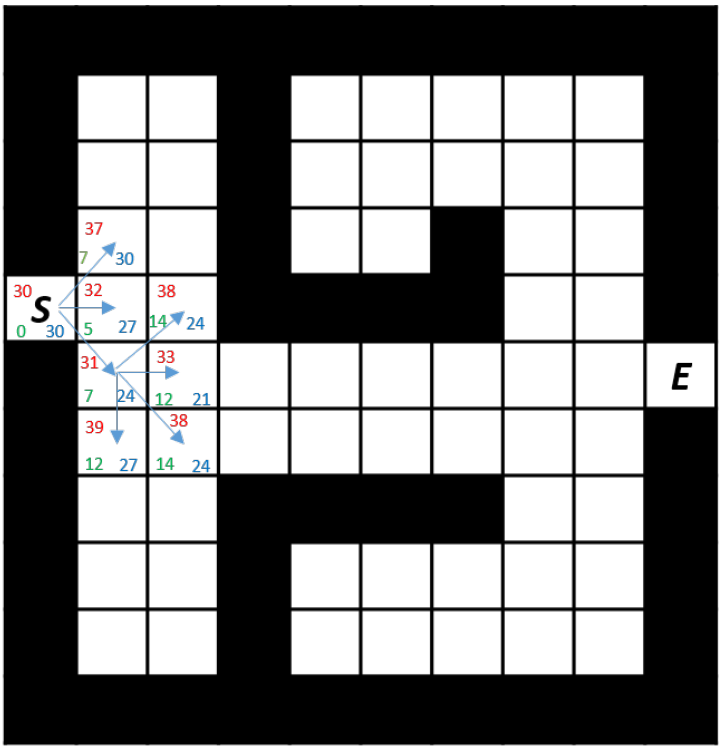

Figura 1: Laberinto y cálculo de costes hacia los vecinos

ii. Si el nodo está en $\mathcal{A}$, tener en cuenta que quizás sea necesario actualizar su coste $g(n)$ y, por lo tanto, su padre en el camino. Nótese que se deben recalcular los costes necesarios.

3. Si $\mathcal{A} = \emptyset$ y no se ha llegado a la salida del laberinto, E, no existe camino y se deberá mostrar en pantalla un mensaje que así lo indique.

Obstáculos dinámicos paso a paso

En esta práctica consideraremos que el entorno del laberinto es dinámico. Por lo tanto, los obstáculos del laberinto pueden cambiar de estado a lo largo de la ejecución del camino generado. El objetivo es simular un escenario más realista, donde el agente no se enfrenta a un mapa fijo, sino a un entorno que evoluciona en el tiempo.

El funcionamiento es el siguiente:

- Mientras A* está ejecutándose, no cambias el mapa. Es decir, durante una planificación concreta los obstáculos se mantienen fijos. De esta manera, la frontera es consistente con el estado del mapa en ese instante.

- Una vez que encuentras un camino, ahora toca avanzar paso a paso por él. Lo que sucede paso a paso es que el laberinto se actualiza, habrá obstáculos que nacen y

mueren. Entonces, el conjunto de nodos frontera deja de tener sentido dado que la configuración del laberinto ha cambiado. Por ello, **en cada iteración reinicias A* desde la nueva posición y con el nuevo estado del laberinto**.

Actualización de los obtáculos paso a paso

Tras cada paso del agente, se actualiza el estado del laberinto:

Cada casilla libre puede convertirse en obstáculo con probabilidad p_{in}.

Cada casilla que actualmente es obstáculo puede liberarse con probabilidad p_{out}

Este dinamismo es independiente para cada casilla y se modela mediante una variable aleatoria uniforme $U(0,1)$.

Para casillas libres:

$$\texttt{Si } U(0,1) < (1 - p_{in}), \texttt{ entonces libre; obstáculo, en otro caso.}$$

Para casillas con obstáculo:

$$\texttt{Si } U(0,1) < (1 - p_{out}), \texttt{ entonces obstáculo; libre, en otro caso.}$$

Consideraremos:

$$p_{in} = p_{out} = 0,5$$

La proporción máxima de casillas bloqueadas no podrá superar el 25% del total del laberinto. Si se excede este límite, se eliminarán aleatoriamente algunos obstáculos hasta respetar dicho límite.

Las casillas correspondientes a la posición inicial (S) y final (E) nunca se bloquean.

Caso de bloqueo temporal

Puede ocurrir que, en un paso dado, no exista ningún camino disponible hacia el destino. En ese caso, el sistema aplicará una nueva actualización del entorno y volverá a ejecutar A*.

Este proceso se repetirá hasta un máximo de 5 reintentos consecutivos.

Si después de estos 5 intentos no se genera un camino válido, el algoritmo concluye que el destino es inalcanzable en las condiciones actuales y finaliza la ejecución.

Finalización

El proceso termina cuando el agente alcanza el destino o cuando se declara que no es posible llegar tras varios intentos fallidos.

Durante la ejecución, se contabilizarán las siguientes métricas de interés: número total de nodos generados e inspeccionados, número de pasos realizados, número de reintentos sin éxito y proporción media de obstáculos dinámicos presentes en el entorno.

Además de eso, en cada paso de ejecución se imprimirá el laberinto en pantalla para mostrar la configuración actual del mismo y el camino que ha sido generado. Con esta impresión paso a paso, se observará cómo ha evolucionado nuestro entorno dinámico y cómo se ha ido reconfigurando el camino para llegar a la salida.

Se deberá hacer uso de la estructura de datos adecuada para almacenar el camino óptimo y el coste del mismo. También deben ser impresos en pantalla.

Implementación

Las instancias del problema se suministrarán en un fichero de texto con el formato mostrado en la Figura 2. El punto 3 representa el punto de entrada al laberinto, el punto 4 representa la salida, el número 1 indica que hay un obstáculo y la casilla no es transitable y, por último, el número 0 representa las casillas de paso.

1	1	1	1	1	1	1	1	1	1
1	0	0	1	0	0	0	0	0	1
1	0	0	1	0	0	0	0	0	1
1	0	0	1	0	0	1	0	0	1
3	0	0	1	1	1	1	0	0	1
1	0	0	0	0	0	0	0	0	***4***
1	0	0	0	0	0	0	0	0	1
1	0	0	1	1	1	1	0	0	1
1	0	0	1	0	0	0	0	0	1
1	0	0	1	0	0	0	0	0	1
1	1	1	1	1	1	1	1	1	1

Figura 2: Ejemplo de instancia

Tareas

a) Diseñar e implementar una búsqueda A* para el problema del camino de mínimo coste entre las casillas de entrada y salida de un laberinto dado haciendo uso de la distancia de Manhattan en entornos dinámicos.

   Además, debe poder indicarse por teclado cuáles son las casillas de entrada y salida del laberinto.

b) Volcar en un **fichero la salida** indicada por la **Tabla 3**, así como **el camino obtenido marcado con un * sobre el laberinto de entrada con el formato indicado en la Figura 1**, iteración a iteración.

c) Analizar el comportamiento de la búsqueda A* considerando funciones heurísticas alternativas a la propuesta en este guión de prácticas.

Qué debe presentar el alumno

a) Código fuente, debidamente comentado, y fichero ejecutable.

b) Una memoria en formato pdf en la que se describan brevemente la búsqueda A* diseñada enumerando las estructuras de datos usadas y cualquier elemento necesario para comprender el diseño propuesto.

c) La memoria debe incluir también tablas o gráficas de resultados que muestren el comportamiento de la búsqueda sobre diferentes instancias del problema. En la Figura 3 se muestra el formato de estas tablas de resultados. Además, se debe **dibujar el camino obtenido marcado con un * sobre el laberinto de entrada con el formato indicado en la Figura 1**.

   Se han considerado tres laberintos (instancias M_1, M_2, M_3), de diferentes tamaños, con n el número de filas y m el número de columnas, con varias combinaciones de casillas S y E. En la tabla se mostrará el camino de coste mínimo mínimo para ir de S a E, su coste y los nodos generados e inspeccionados por la búsqueda A* que usa la función heurística $h(\cdot)$.

El fichero de salida debe contener la Tabla de la Figura 3, el laberinto con el camino mínimo pintado (marcado con $*$), y los nodos generados e inspeccionados en cada iteración, tal como se ha hecho en la Práctica 1. Todo en este orden.

Búsqueda A*. Función heurística $h(\cdot)$

Instancia	*n*	*m*	*S*	*E*	*Camino*	*Coste*	*Número de nodos generados*	*Número de nodos inspeccionados*
M_1								
M_1								
M_2								
M_2								
M_3								
M_3								

Figura 3: Búsqueda A*. Tabla de resultados