

Introducción a los Sistemas Informáticos Inteligentes

Convocatoria de Febrero 2010 – Boletín de ejercicios.

Fechas de entrega: a especificar en las sesiones de laboratorio.

Ejercicio 2.

Considérese el problema de encontrar un camino, en la situación representada en la figura, desde la posición “i” hasta la posición “e”. El NPC (*non player character*) puede moverse de forma horizontal y vertical, un sólo cuadrado en cada movimiento (cada movimiento tiene coste uno). Las zonas sombreadas impiden el paso del NPC a través de ellas.

		e		
e				
			i	

Aplicaremos los distintos algoritmos de búsqueda que se tratan en el laboratorio para resolver el problema, en el siguiente orden:

búsqueda en anchura
búsqueda en profundidad
búsqueda de coste uniforme
algoritmos A y A*

Consideraciones:

Para aquellos algoritmos en los que no es relevante el coste, el orden de los operadores (movimientos) es: arriba, izquierda, derecha, abajo. De producirse un ciclo en el uso de algún algoritmo que no lo controle, supondremos que existen mecanismos para eliminarlo.

El coste del movimiento es 1 si se produce en dirección vertical y 2 si se produce en dirección horizontal.

Para el caso del **algoritmo A** se utilizará la distancia Manhattan como heurística, es decir, la heurística $h(n)$ para el nodo n es la distancia Manhattan desde n al nodo e . La distancia Manhattan entre dos puntos es la distancia en horizontal más la distancia en vertical. En el ejemplo, la distancia Manhattan entre i y e es 4.

En el estudio de la **búsqueda en profundidad**, el límite de profundidad es 5. Describe qué es lo que sucede. Indica el tipo de fracaso (si lo hay). ¿Qué sucede si aumentamos el límite de profundidad a 6?

Representación de la solución:

Resuelve el problema con cada uno de los algoritmos de búsqueda propuestos.

Indica, para cada algoritmo de búsqueda, cuál es el algoritmo que se aplica para extraer los nodos de la frontera.

Escribe la evolución del conjunto de nodos frontera y el conjunto de nodos explorados durante el transcurso del desarrollo del algoritmo, así como la función coste y la heurística en los algoritmos que hagan uso de ellas.

En la figura, nombra (enumera) los nodos según el orden en el que son generados (incluidos en la frontera).

Indica cuándo la función test objetivo determina que el nodo chequeado es la solución.

Indica cuál es la profundidad en la que se ha encontrado la solución.

Razona y explica en cada algoritmo qué nodos y por qué conforman la solución.

Representa en la figura (con una flecha) el camino que conforma la solución.

Dibuja el árbol de búsquedas.

Ejercicio 3.

La heurística utilizada en el algoritmo A, ¿es admisible?. ¿Por qué? ¿Podemos decir que el algoritmo es A*?

En el artículo de Alseek "Intelligence for new worlds", el autor comenta en la página 5 respecto al movimiento de un NPC:

"Unlike today's software-based approaches (e.g., A*), the Intia's pathfinding uses no heuristics, thereby guaranteeing that the optimal path will always be found. This optimality also means that the Intia processor avoids the common pitfalls of A*, including failures to find a path when one exists, and the generation of "artifacts" (e.g., weird, unrealistic paths)."

"A diferencia de las aproximaciones/soluciones software de hoy en día (ejemplo algoritmo A*), el procesador Intia de búsqueda de caminos no usa ninguna heurística, garantizando por tanto que siempre se encontrará el camino óptimo. Esta optimización significa que Intia evita los fallos comunes de A*, incluyendo los fracasos en la búsqueda de un camino cuando existe, y la generación de "artefactos" (caminos no realistas o absurdos)".

¿Esta afirmación es consistente con tu respuesta a la cuestión 1 acerca de si A es A*?

En base a los resultados obtenidos, discute cuál de los algoritmos es más adecuado a este problema.

Ejercicio 4.

En el hipotético caso de que el servicio Mapas de Google empleara el algoritmo de **búsqueda bidireccional** para encontrar la ruta más corta (en Km.) entre dos localidades, calcula la solución que ofrecería para la ruta Ourense-Calatayud dadas las siguientes distancias kilométricas:

(Ourense, Ponferrada: 175)	(Ourense, Benavente: 236)
(Ponferrada, León: 113)	(Ponferrada, Benavente: 125)
(Benavente, León: 75)	(Benavente, Valladolid: 112)
(Benavente, Palencia: 112)	(Palencia, León: 131)
(Palencia, Valladolid: 48)	(Palencia, Osorno: 49)
(Palencia, Burgos: 92)	(León, Osorno: 121)
(Osorno, Burgos: 59)	(Valladolid, Aranda: 95)
(Burgos, Aranda: 84)	(Aranda, Osma: 58)
(Osma, Calatayud: 140)	(Osma, Soria: 58)
(Burgos, Soria: 143)	(Burgos, Logroño: 150)
(Logroño, Soria: 106)	(Soria, Calatayud: 91)