Técnicas de los Sistemas Inteligentes.

Práctica 1. Experimentación con técnicas de búsqueda.

UNIVERSIDAD DE GRANADA

Ignacio Yuste López.

54141533Q

TSI. A1.

Tabla de resultados

Algoritmo	Mapa	Runtime(ms)	Tamaño de la ruta	Número de nodos expandidos	Máximo número de nodos en memoria
BFS	Muy pequeño	0	15	96	96
	Pequeño	1	34	130	130
	Mediano	2	110	845	845
	Grande	5	209	4632	4632
DFS	Muy pequeño	0	27	70	70
	Pequeño	0	64	80	80
	Mediano	1	238	401	401
	Grande	4	935	1212	1212
A*	Muy pequeño	2	15	54	55
	Pequeño	3	40	99	100
	Mediano	12	261	602	603
	Grande	38	261	1564	1565
IDA*	Muy pequeño	11	15	4790	11
	Pequeño	1	34	38	24
	Mediano	TO	155	90699	59
	Grande	-	-	-	-

Cuestiones

1. Entre BFS y DFS, ¿qué algoritmo puede ser considerado más eficiente de cara a encontrar el camino óptimo?

   El algoritmo de búsqueda en anchura es sin duda el más óptimo. Si bien es cierto que para mapas grandes puede llegar a ocupar demasiada memoria siempre encuentra el camino óptimo. La búsqueda en profundidad es más rápida y "ligera" en comparación, pero sólo basta observar la distancia de las rutas para comprobar que son mucho más largas en comparación. Esto se debe al orden de expansión de los hijos, en nuestra práctica primero expandimos los nodos superior e inferior y luego los horizontales, por lo que cuando ejecutamos el plan vemos cómo el agente recorre los huecos sin obstáculos de arriba abajo, avanzando sólo en horizontal cuando se encuentra un obstáculo en una de estas direcciones.

2. ¿Se podría decir que A* es más eficiente que DFS?

   De hecho es mucho más eficiente. Como hemos visto antes, DFS está lejos de alcanzar el camino óptimo de forma sistemática (es posible que lo encuentre pero no depende del algoritmo sino del mapa) mientras que A* sí que lo encuentra. Además, los tiempos de búsqueda son bastante buenos y la memoria que llega a ocupar no dista mucho de la que ocupa DFS.

   (Cabe decir que mi algoritmo A* no siempre encuentra el óptimo, creo que esto es debido a algún problema con el manejo de las estructuras utilizadas y no del algoritmo en sí)

3. ¿Cuáles son las principales diferencias entre A* e IDA*? ¿En qué contextos es más conveniente usar uno u otro?

   Las diferencias más importantes son el tiempo de ejecución y la memoria consumida. Ambos logran hallar el camino óptimo, sin embargo, A* lo encuentra mucho más rápido pero IDA* ocupa mucho menos espacio.

   El mayor potencial de IDA* es el uso de la recursividad de la misma manera que lo hace DFS. Al hacer uso de esta, logra encontrar la solución teniendo solo la rama actual en memoria. Pero para encontrar la solución óptima tiene un sistema de cotas con el cual va explorando todas las ramas hasta cierta profundidad, si no encuentra una solución en esa cota la aumenta y vuelve a probar llegando un poco más profundo. Debido a esto encuentra la solución con el mejor coste pero también aumenta significativamente el tiempo de ejecución ya que explora los mismos nodos una y otra vez. Por otro lado, A* obtiene los mejores tiempos ya que almacena todos los nodos explorados con un sistema de dos listas, pero debido a esto mismo ocupa mucho espacio en memoria.

   El contexto para elegir uno u otro es sencillo, depende de si buscamos mejores tiempos de ejecución o si buscamos un algoritmo que encuentre la mejor solución sin importar el tiempo pero ocupe lo menos posible en memoria.

4. ¿Se podría decir que RTA* es más eficiente que A*?

   No es justo comparar en eficiencia ya que su uso es distinto. Mientras que A* da una solución completa, la cual requiere más tiempo de búsqueda, RTA* da múltiples soluciones parciales las cuales llevan al objetivo.

   Si comparamos lo que le cuesta a RTA* encontrar un solución parcial con lo que le cuesta a A* encontrar la solución obviamente RTA* va a ganar. Si lo hacemos al revés, A* llevará al agente a la solución final antes que RTA*.

   La clave está en el contexto del problema. Si nos encontramos en un entorno estático no merece la pena usar RTA* ya que el camino no va a cambiar nunca, con lo cual A* es la mejor decisión. Por otro lado, si el entorno es dinámico (hay objetos móviles, enemigos que nos persiguen y nos cortan el paso, nuevas trampas, etc...) A* no será capaz de llevarnos al objetivo casi con total seguridad ya que no podrá anticiparse a estos cambios, mientras que RTA*, al ir calculando el camino de firma parcial, podrá ver cual es la mejor decisión en cada paso tal cual esté el mapa en su estado actual.