

Práctica 1: Desarrollo de agentes basado en técnicas de búsqueda (heurística) dentro del entorno GVGAI

Juan Antonio Martínez Sánchez

Curso Académico 2021-22

Técnicas de los Sistemas Inteligentes

1 Tabla de resultados

Alg.	Mapa	Runtime (acumulado)	Tamaño de la ruta calculada	Número de nodos expandidos	Máximo número de nodos en memoria
BFS	Muy pequeño	6	15	96	97
	Pequeño	6	34	130	131
	Mediano	19	110	846	846
	Grande	55 → cd	209	4630	4633
DFS	Muy pequeño	2	27	71	71
	Pequeño	1	64	81	81
	Mediano	8	238	402	402
	Grande	17	935	1213	1213
A*	Muy pequeño	4	15	38	54
	Pequeño	5	34	86	104
	Mediano	21	110	593	617
	Grande	63 → cd	209	3607	3702
IDA*	Muy pequeño	1	16	122	16
	Pequeño	4	35	423	35
	Mediano	429 → cd	111	484625	111
	Grande				
RTA*	Muy pequeño				
	Pequeño				
	Mediano				
	Grande				

cd → Controller Discualified

2 Preguntas sobre la práctica

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

Como podemos observar en BFS obtenemos siempre un tamaño de ruta mucho menor que el obtenido con DFS. Sin embargo, con DFS obtenemos una ruta en un tiempo menor que BFS y además exploraremos menos nodos y guardaremos menos en memoria. También tendremos en cuenta que el coste por movimiento siempre será 1 que implicará que la ruta mas corta será la de menor coste.

En conclusión, si necesitamos el camino más optimo nos quedaremos con BFS, ya que DFS no siempre encontrará el camino más optimo y DFS sí. Sin embargo en algunos casos que necesitemos un camino en poco tiempo podremos optar por DFS.

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

Si consideramos que es más eficiente porque obtiene el camino más optimo, sí podríamos decir que A* es más eficiente que DFS pero tiene sus contras. Como podemos observar en la tabla A* tarda más tiempo en encontrar la solución que DFS. Además expandirá y guardará más nodos en memoria por tanto será más costoso, pero finalmente encontrará la solución óptima.

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

Ambos realizan una búsqueda heurística de escoger los caminos entre nodos con el menor coste, es decir, escogerá el mejor nodo. Sin embargo, A* hará una búsqueda en anchura mientras que IDA* la hará en profundidad. También podemos distinguir que el consumo de memoria en IDA* es mucho menor y que la complejidad de tiempo es exponencial a diferencia de A*.

A la hora de elegir uno u otro tendremos que ver si el consumo de memoria o el tiempo son un problema. Cuando el consumo de memoria sea un problema será conveniente utilizar IDA*, aunque expanda más nodos en memoria guardará menos y por esto será más conveniente frente a A*. Ahora si el tiempo es un problema utilizaremos A* ya que conforme los problemas sean más grandes, el tiempo de IDA* se incrementará exponencialmente y esto con A* no ocurrirá como podemos observar en la tabla.

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

RTA* al ser una búsqueda heurística en tiempo real irá guardando los nodos visitados y conforme vaya calculando rutas a los mejores nodos del espacio de búsqueda local también actualizará el valor de la heurística para posibles futuros movimientos. El principal problema de RTA* es la necesidad de replanificar la ruta en casos donde no pueda seguir adelante y la posibilidad de caer en bucles de los que no pueda salir.

Debido a esto, A* será más eficiente ya que no tendremos esos problemas que harían que la ruta final no fuera la más óptima y también no tendríamos la necesidad de replanificar, lo cual A* calcularía la ruta sin dar rodeos innecesarios y yendo directo al objetivo, todo esto sin tener en cuenta el costo de memoria que consume A* a diferencia de RTA* que es mucho menor.