

Alberto García Doménech

Pablo Daurell Marina

Inteligencia Artificial

Práctica 2

Conclusiones ejercicio 5 (8-puzzle)

En cuanto a los algoritmos de búsqueda ciega utilizados (primero en anchura, primero en profundidad y coste uniforme) podemos ver que en la mayoría de los casos obtenemos mejores resultados con la búsqueda en anchura y peores (bastante peores) con la búsqueda en profundidad, mientras que con la búsqueda de coste uniforme obtenemos resultados muy similares a la búsqueda en profundidad lo cual se debe a que el coste de los caminos es proporcional a la profundidad del árbol.

Al tener un espacio de estados muy amplio, la búsqueda en profundidad pierde mucho tiempo explorando caminos con profundidad muy grande pero que no llegan a la solución (salvo en el puzzle E3 que tardamos muy poco en encontrar la solución, ya que este algoritmo puede obtener resultados rápido si tenemos suerte y existe una solución que no requiere explorar demasiado el espacio de estados). Por otro lado, la búsqueda primero en anchura nos garantiza encontrar una solución de longitud mínima y es mucho más improbable que quede atrapada en caminos sin salida.

Por otro lado, los algoritmos con búsqueda heurística utilizados (a^* y primero el mejor) obtenemos mejores resultados de tiempo con la búsqueda primero el mejor ya que empezamos a buscar por el nodo más prometedor y vamos evaluando los nodos a expandir para elegir el más prometedor también, sin embargo la búsqueda a^* consigue encontrar las soluciones más óptimas siempre, a diferencia de la búsqueda primero el mejor.

Conclusiones del problema de los misioneros

(Algoritmos usados: primero en profundidad y primero en anchura, tanto graph search como tree search, y búsqueda uniforme)

Una solución óptima constaría como mínimo de once pasos.

La búsqueda en anchura y la búsqueda en profundidad llegan a distintas soluciones pero ambas tienen una longitud de once pasos, por lo tanto ambas soluciones son óptimas.

Los algoritmos de graph search consumen más memoria dado que aparte de las lista de nodos abiertos, también conservan una lista de los cerrados.

En cuanto a los tiempos de ejecución, el algoritmo de búsqueda en profundidad (graph search) es el más rápido ya que, en este caso, solo necesita recorrer pocas ramas del árbol de estados hasta encontrar el nodo con la solución. Sin embargo esto podría variar si cambiáramos la implementación del problema.