

Memoria Práctica 2 - Inteligencia Artificial

Fernando Peña Bes (756012)

8 de noviembre de 2019, Universidad de Zaragoza

1. Introducción

El objetivo de esta práctica es realizar experimentos sobre el 8-puzzle para recopilar información relevante sobre la eficiencia de ciertos algoritmos de búsqueda.

Se realizarán búsquedas ciegas (BFS e IDS) y búsquedas informadas con el algoritmo A* con las heurísticas de fichas descolocadas y Manhattan. Una vez terminadas las pruebas, se comparará la eficiencia de los algoritmos mostrando el número de nodos generados y el factor de ramificación efectivo b^* para distintas profundidades.

Para realizar la práctica se ha usado el entorno de desarrollo Eclipse para Java y el código perteneciente al libro *Artificial Intelligence: A Modern Approach* [1], en la versión 1.8 (se puede descargar en <https://github.com/aimacode/aima-java/releases/tag/aima3e-v1.8.1>).

2. Metodología

La metodología de los experimentos sobre el 8-puzzle es la siguiente: Se generan 100 parejas de estado inicial (tablero inicial) y estado final (tablero final¹) de forma aleatoria, usando los métodos disponibles en la clase diseñada por los profesores `GenerateInitialEightPuzzleBoard.java`, `random(int depth, EightPuzzleBoard initialState)` y `random(int depth, EightPuzzleBoard initialState)`.

Estos tableros se generan de forma que cada pareja esté a una distancia fija `depth`. Como el algoritmo de generación de tableros no asegura que el camino óptimo al estado final esté a `depth` pasos, se comprueba si se cumple resolviendo el problema con el algoritmo A* con la heurística Manhattan. En el caso que la pareja de tableros no cumpla la restricción, se descarta y se calcula otra.

Una vez que se tienen 100 parejas de tableros válidas, se resuelve el problema del 8-puzzle usando esos tableros como estados iniciales y objetivos mediante diferentes algoritmos de búsqueda y se calcula la media de nodos generados por cada algoritmo y el factor de ramificación efectivo b^* . Para el cálculo de b^* , se ha usado el método de la bisección usando la clase `Biseccion.java`.

Este conjunto de experimentos se realiza 22 veces, para valores de `depth` $\in [2, 24]$.

Los algoritmos usados para obtener las métricas han sido BDF, IDS y A* con las heurísticas de fichas descolocadas y Manhattan. Se adaptó el código de AIMA para que el algoritmo IDS calculara el número de nodos generados, modificando la clase `IterativeDeepeningSearch.java`, disponible en `aima.core.search.uninformed` y `NodeExpander.java`, disponible en `aima.core.search.framework`. Después, se modificó la clase `GenerateInitialEightPuzzleBoard.java` en el paquete `aima.core.environment.eightpuzzle`, definiendo el tablero objetivo como estático y añadiendo el método estático `setGoalState(EightPuzzleBoard board)` y, por último, se reprogramaron

¹Para poder generar más parejas de estados diferentes, el tablero final no tiene por qué ser el tablero de la solución del 8-puzzle, donde todas las fichas están ordenadas crecientemente.

las heurísticas del 8-puzzle (las clases `ManhattanHeuristicFunction.java` y `MisplacedTilleHeuristicFunction`) para que se pudieran adaptar a diferentes estados objetivo.

3. Resultados

A continuación, en la tabla 1, se muestran los resultados obtenidos a partir de las pruebas descritas anteriormente:

Nodos Generados					b*				
d	BFS	IDS	A*h(1)	A*h(2)	BFS	IDS	A*h(1)	A*h(2)	
2	8	10	5	5	2,37	2,70	1,79	1,79	
3	18	33	9	8	2,22	2,81	1,66	1,58	
4	38	101	12	11	2,15	2,86	1,49	1,45	
5	69	270	18	14	2,05	2,81	1,46	1,37	
6	125	787	23	17	2,00	2,83	1,40	1,31	
7	216	2141	32	22	1,95	2,81	1,39	1,29	
8	377	6305	49	28	1,92	2,83	1,40	1,28	
9	609	16619	76	37	1,87	2,80	1,42	1,28	
10	1009	47000	113	46	1,85	2,81	1,43	1,27	
11	1711	146883	173	64	1,83	2,83	1,44	1,28	
12	2656	376434	266	82	1,80	2,81	1,45	1,28	
13	4325	1087849	419	125	1,79	2,82	1,46	1,30	
14	6963	3041002	626	146	1,77	2,81	1,46	1,28	
15	11168	8734955	999	210	1,76	2,82	1,47	1,30	
16	17913	---	1530	304	1,75	---	1,47	1,31	
17	28273	---	2521	438	1,74	---	1,48	1,32	
18	42008	---	3651	582	1,72	---	1,48	1,32	
19	63523	---	5527	745	1,71	---	1,48	1,31	
20	91362	---	9146	949	1,69	---	1,49	1,31	
21	129183	---	13011	1263	1,68	---	1,49	1,31	
22	176441	---	21197	1780	1,66	---	1,50	1,32	
23	232850	---	30628	2228	1,64	---	1,49	1,31	
24	288541	---	46842	2989	1,62	---	1,49	1,32	

Tabla 1: Resultados ejecución `EightPuzzlePract2.java`

No ha sido posible seguir realizando cálculos con IDS a partir de parejas de estados distancia 16 por la complejidad exponencial del tiempo de ejecución, ya que la búsqueda se realiza en árbol. Este problema es especialmente complicado resolverlo en árbol, ya que cada paso que se da se puede deshacer y las soluciones no se suelen encontrar en niveles profundos.

Las columnas **A*h(1)** corresponden al algoritmo de A* con la heurística de fichas descolocadas y las **A*h(2)**, al algoritmo A* con heurística Manhattan.

Los resultados obtenidos son muy similares a los propuestos en el enunciado de la asignatura.

Si el número de nodos generados por un algoritmo es N , el factor de ramificación efectivo es el factor de ramificación que debería tener un árbol uniforme con profundidad d para contener $N + 1$ nodos. Cuanto más se acerque a 1, mejor ya que se recorren menos nodos innecesarios para calcular la solución. En el problema del 8-puzzle el factor de ramificación siempre se mueve entre 2 y 4.

Se puede ver como la heurística Manhattan es más informada que la de fichas descolocadas ya que consigue un valor de b^* más bajo. El b^* de la búsqueda BFS va disminuyendo conforme más profunda está la solución. Se puede ver también, que el algoritmo IDS tiene factor de ramificación efectivo muy alto, en parte, debido a los nodos repetidos que se recorren.

4. Conclusiones

En esta práctica se han evaluado diferentes algoritmos y heurísticas y se ha visto cuales son más adecuadas para resolver el problema del 8-puzzle. El mejor algoritmo de los evaluados, es el A* con la heurística de Manhattan.

Referencias

- [1] Stuart Russell and Peter Norvig, *Artificial Intelligence: A Modern Approach*.
- [2] *Apuntes de la asignatura Inteligencia Artificial*, Curso 2019-20.