



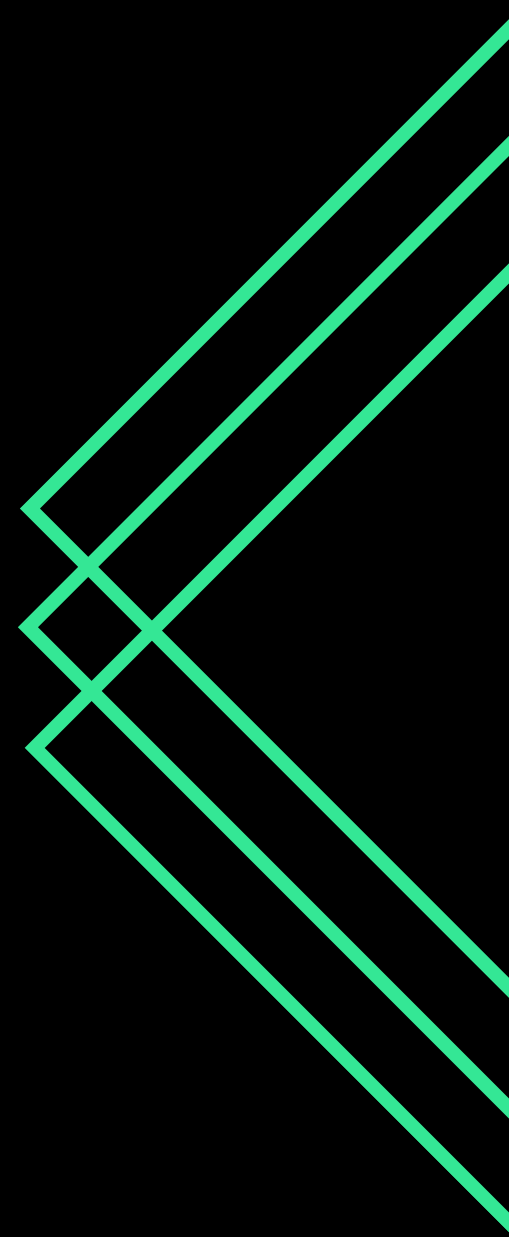
13 de noviembre, 2019

Sistemas Inteligentes

PRÁCTICA 1: 15-PUZZLE



Víctor Calabuig Rodríguez



Índice

1 - Introducción	2
2 - Resultados	3
3 - Análisis	6
3.1 - Complejidad	6
3.2 - Optimalidad	8
3.2 - Completitud	9
4 - Bibliografía	10

1 - Introducción

Esta memoria corresponde a la realización de la primera práctica de la asignatura Sistemas Inteligentes, en la que se trata de implementar varios algoritmos de búsqueda para resolver el juego 15-puzzle (imagen):

	1	2	3
4	5	6	7
8	9	10	11
12	13	14	15

La memoria está formada por 2 partes principales:

- a. Resultados: En esta parte se recogen los resultados obtenidos (tiempo de ejecución, nodos explorados, y profundidad de la solución) al aplicar cada algoritmo de búsqueda a 11 configuraciones iniciales (de puzzles) diferentes.
- b. Análisis: En esta parte se analizan los resultados obtenidos previamente, con el objetivo de comparar las diferentes búsquedas (su rendimiento, optimalidad, y completitud).

2 - Resultados

A continuación se muestran los resultados obtenidos para cada configuración inicial del puzzle (desde el puzzle 0 hasta el 10) con los diferentes métodos de búsqueda:

- Primero en anchura
- Primer en profundidad
- Profundidad iterativa
- Heurística de fichas descolocadas
- Heurística de distancias Manhattan

****Nota:** las búsquedas que no han hallado una solución para algún puzzle después de ejecutarse durante 5 minutos (seguían buscando), o que han producido un de tipo “Out of memory” están señaladas con la etiqueta “na” (no acaba).

Puzzle 0 (1,2,3,7, 8,4,5,6, 9,10,0,11, 12,13,14,15)

Tipo de búsqueda	Profundidad de la solución	Nodos explorados	Tiempo de ejecución (ms)
Anchura	10	5.979	39
Profundidad	30	1.218.120	3.904
Profundidad Iterativa	10	6.113	27
Heurística Descolocadas	10	12	6
Heurística Manhattan	10	11	7

Puzzle 1 (1,2,3,7, 8,4,5,6, 12,9,10,11, 13,14,15,0)

Tipo de búsqueda	Profundidad de la solución	Nodos explorados	Tiempo de ejecución (ms)
Anchura	12	15.256	66
Profundidad	30	1.331.405	5.725
Profundidad Iterativa	12	15.869	51
Heurística Descolocadas	12	14	6
Heurística Manhattan	12	13	7

Puzzle 2 (1,2,3,7, 8,4,5,6, 0,10,11,15, 12,13,9,14)

Tipo de búsqueda	Profundidad de la solución	Nodos explorados	Tiempo de ejecución (ms)
Anchura	18	890.007	8.457
Profundidad	na	-	-
Profundidad Iterativa	18	2.061.676	4.072
Heurística Descolocadas	18	588	27
Heurística Manhattan	18	128	19

Puzzle 3 (1,2,3,7, 8,4,5,6, 10,11,15,14, 9,12,13,0)

Tipo de búsqueda	Profundidad de la solución	Nodos explorados	Tiempo de ejecución (ms)
Anchura	20	2.625.988	36.652
Profundidad	na	-	-
Profundidad Iterativa	20	6.380.976	18.212
Heurística Descolocadas	20	362	22
Heurística Manhattan	20	29	10

Puzzle 4 (1,2,3,7, 8,4,5,6, 13,12,0,9, 14,15,11,10)

Tipo de búsqueda	Profundidad de la solución	Nodos explorados	Tiempo de ejecución (ms)
Anchura	na	-	-
Profundidad	na	-	-
Profundidad Iterativa	na	-	-
Heurística Descolocadas	22	2.843	55
Heurística Manhattan	22	45	12

Puzzle 5 (1,2,3,7, 8,4,5,6, 11,15,0,14, 10,12,13,9)

Tipo de búsqueda	Profundidad de la solución	Nodos explorados	Tiempo de ejecución (ms)
Anchura	na	-	-
Profundidad	na	-	-
Profundidad Iterativa	na	-	-
Heurística Descolocadas	28	235.048	2.511
Heurística Manhattan	28	519	34

Puzzle 6 (1,2,3,7, 8,4,5,6, 11,15,14,13, 10,0,9,12)

Tipo de búsqueda	Profundidad de la solución	Nodos explorados	Tiempo de ejecución (ms)
Anchura	na	-	-
Profundidad	na	-	-
Profundidad Iterativa	na	-	-
Heurística Descolocadas	30	651.078	6.464
Heurística Manhattan	30	661	38

Puzzle 7 (1,2,3,7, 8,4,5,6, 0,14,13,12, 15,11,10,9)

Tipo de búsqueda	Profundidad de la solución	Nodos explorados	Tiempo de ejecución (ms)
Anchura	na	-	-
Profundidad	na	-	-
Profundidad Iterativa	na	-	-
Heurística Descolocadas	32	2.389.564	55.331
Heurística Manhattan	32	2.711	89

Puzzle 8 (1,2,3,7, 8,4,5,6, 15,14,9,13, 11,10,12,0)

Tipo de búsqueda	Profundidad de la solución	Nodos explorados	Tiempo de ejecución (ms)
Anchura	na	-	-
Profundidad	na	-	-
Profundidad Iterativa	na	-	-
Heurística Descolocadas	na	-	-
Heurística Manhattan	36	23.991	522

Puzzle 9 (1,2,3,7, 15,8,4,5, 14,9,6,13, 11,0,10,12)

Tipo de búsqueda	Profundidad de la solución	Nodos explorados	Tiempo de ejecución (ms)
Anchura	na	-	-
Profundidad	na	-	-
Profundidad Iterativa	na	-	-
Heurística Descolocadas	na	-	-
Heurística Manhattan	44	253.550	5.858

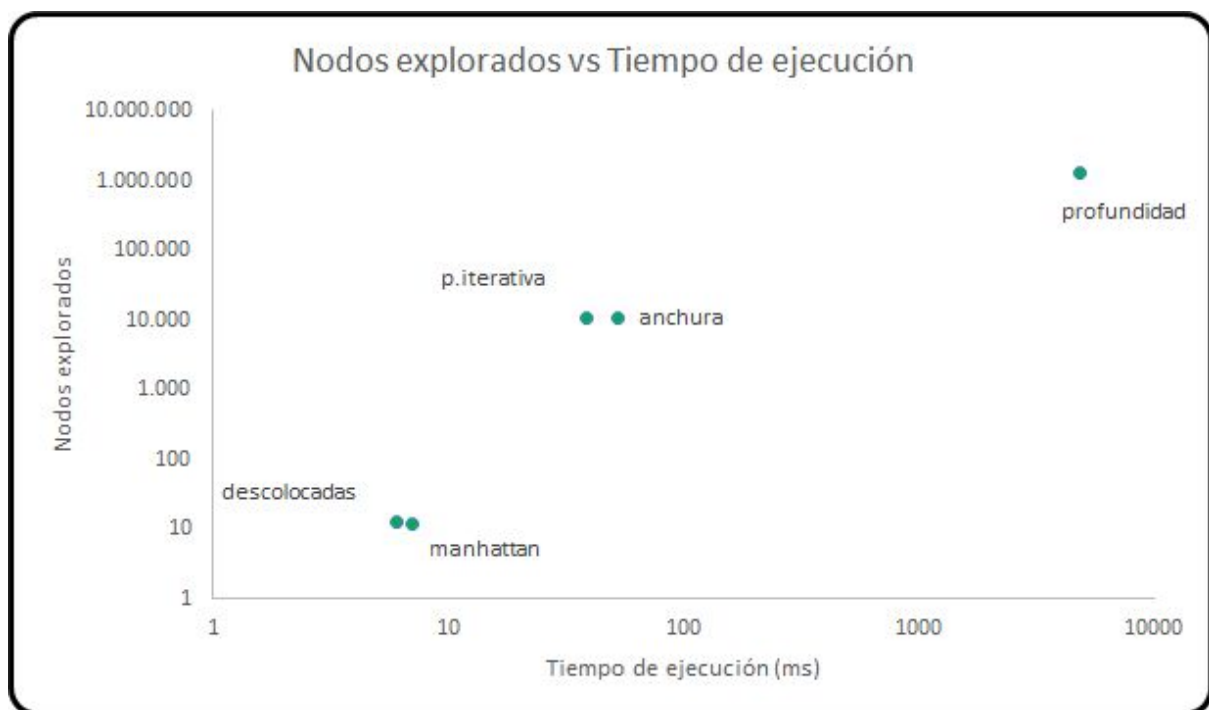
Puzzle 10 (1,2,3,4, 5,6,7,8, 9,10,11,12, 13,15,14,0)

Tipo de búsqueda	Profundidad de la solución	Nodos explorados	Tiempo de ejecución (ms)
Anchura	na	-	-
Profundidad	na	-	-
Profundidad Iterativa	na	-	-
Heurística Descolocadas	na	-	-
Heurística Manhattan	44	944.503	31.934

3 - Análisis

3.1 - Complejidad

A continuación se muestra una gráfica en la que se ha calculado la media de nodos explorados y tiempo de ejecución de cada búsqueda para los 2 primeros puzzles, que son los únicos que logran ser resueltos por todas las búsquedas:



A primera vista pueden observarse 3 categorías: En la peor de todas la búsqueda en profundidad; en una categoría intermedia las búsquedas en anchura y profundidad iterativa, y en la mejor categoría las búsquedas heurísticas.

Para los 2 primeros puzzles, la búsqueda en profundidad es la que más nodos explora y la que más tiempo tarda en hallar una solución. Esto se debe a que la búsqueda al llegar hasta el límite de profundidad (30) numerosas veces sin encontrar ninguna solución, es decir, en ambos puzzles se ha dado un caso cercano al peor caso (si se llega a encontrar una solución en las primeras ramas, esta búsqueda podría haber sido más rápida y utilizar menos memoria que anchura y profundidad iterativa fácilmente).

En cuanto a las búsquedas p.iterativa y anchura, ambas funcionan considerablemente mejor (memoria y tiempo) que la búsqueda en profundidad. En la primera gráfica parece que p. iterativa es mejor ya que de media, para los mismos nodos generados, encuentra la solución en menos tiempo. Sin embargo, esto solo ocurre para soluciones con poca profundidad (la gráfica sólo utiliza los 2 primeros, que tienen profundidad 10 y 12), ya que si observamos los resultados de soluciones con profundidades mayores, observamos que p. iterativa explora muchos más nodos que anchura. Esto se aprecia en la siguiente tabla, que muestra la media de nodos explorados y tiempo de ejecución para los primeros 4 puzzles (los que conseguían resolver estas dos búsquedas):

Búsqueda	Media de nodos explorados	Media de Tiempo de ejecución
Anchura	884.308	11.304
P. Iterativa	2.116.159	5.591

Efectivamente, p. iterativa es más rápida pero explora más nodos. Estos resultados son los esperados ya que la búsqueda en p. iterativa comienza desde el estado inicial cada vez que aumenta el límite de profundidad, por lo que explora los mismos nodos repetidas veces. Aquí es importante destacar que aunque profundidad iterativa explore más nodos, no necesariamente ocupa más memoria (que anchura) ya que para cada iteración se generan nuevas estructuras de abiertos y repetidos/cerrados.

Por otro lado, anchura es más lenta y esto podría deberse a que para cada nodo que explora, debe generar todos sus sucesores, mientras que p.iterativa tan solo genera sucesores hasta el límite de profundidad de esa iteración. Por tanto, una vez profundidad iterativa llega al nivel de profundidad de la solución, explora todos los nodos de esa profundidad sin tener que generar, para cada uno de ellos, sus sucesores.

Por último, las 2 mejores búsquedas en cuanto a nodos explorados y tiempo de ejecución son las 2 búsquedas heurísticas. En la primera gráfica no se aprecian diferencias significativas entre ambas (debido a la profundidad de los dos primeros puzzles). Sin embargo, los resultados para puzzles con soluciones a mayor profundidad muestran

claramente que búsqueda es mejor. A continuación se muestra la media de nodos explorados y tiempo de ejecución de las dos búsquedas heurísticas para los primeros 8 puzzles:

Búsqueda	Media de nodos explorados	Media de tiempo de ejecución
Descolocadas	409.939	8.053
Manhattan	515	27

Cómo se observa en la tabla, la búsqueda heurística con Distancias Manhattan es infinitamente mejor que la búsqueda heurística con Fichas Descolocadas. Además, destacar que a medida que aumenta la profundidad, la diferencia de rendimiento entre una búsqueda y otra crece exponencialmente. Esto nos indica que la estimación que realiza la heurística Distancias Manhattan del coste estimado hasta la solución más cercana ($h(n)$) se aproxima más al coste verdadero hasta dicha solución que en el caso de la heurística Fichas Descolocadas. Por lo tanto, informa mejor sobre cuál es el nodo que más cercano está de una solución, y esto es lo que marca la diferencia entre una búsqueda y otra.

3.2 - Optimalidad

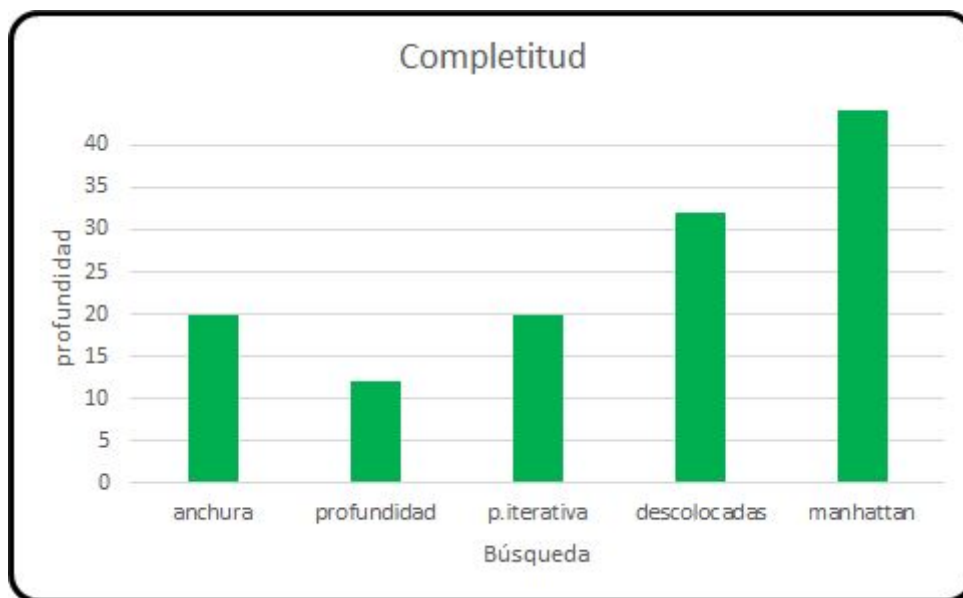
Observando los resultados, la única búsqueda que haya soluciones a una profundidad mayor de la necesaria es la búsqueda en profundidad: Para el puzzle 0 y para el puzzle 1 obtiene soluciones a profundidad 30, frente a las soluciones óptimas (10 y 12 respectivamente).

Destacar que ambas búsquedas heurísticas siempre encuentran soluciones óptimas, lo que nos indica que tanto las distancias Manhattan como las fichas descolocadas son heurísticas admisibles.

3.2 - Completitud

En este apartado se evalúa si las búsquedas son capaces de hallar una solución en un tiempo no superior a 5 minutos, o sin provocar el error “out of memory”. Por lo tanto, si exceden los 5 minutos o se produce el error indicado, se considera que esa búsqueda no es capaz de resolver ese puzzle. Sin embargo, todas las búsquedas son completas, por lo que si no se limitarían los recursos de memoria o tiempo, eventualmente hallarían una solución de cualquier puzzle.

Teniendo en cuenta estas consideraciones, se han resumido los resultados en la siguiente tabla, en la que se indica, para cada búsqueda, la profundidad (de la solución óptima) del puzzle más complejo que es capaz de resolver.



De forma análoga al análisis de complejidad, la búsqueda menos capaz es la búsqueda en profundidad, que solo consigue resolver los 2 primeros puzzles. A partir del tercero, pasados 5 minutos, la búsqueda sigue en ejecución.

Anchura y p. iterativa consiguen ambos resolver hasta el puzzle 3, a una profundidad de 20. Pasado este umbral, la búsqueda en anchura produce el error ‘Out of memory’, mientras que profundidad iterativa supera el límite de 5 minutos.

Finalmente, la búsqueda heurística con fichas descolocadas excede los 5 minutos de búsqueda a partir del puzzle 8, mientras que la heurística con distancias Manhattan consigue resolver todos los puzzles suministrados.

4 - Bibliografía

- Oracle, "Java SE - JDK 13 Documentation",
<https://docs.oracle.com/en/java/javase/13/>
- Vega, O. Resolución de problemas mediante búsqueda [Diapositivas de PowerPoint]. Recuperado 10 de Noviembre de 2019, de
https://poliformat.upv.es/access/content/group/GRA_11683_2019/teor%C3%ADa/tema_2.pdf

-