

15 - PUZZLE

Profesor:
Blai Bonet

Estudiantes	Carnet
Gamar Azuaje	10-10227
Juan Escalante	10-10051
Wilmer Bandres	10-10055

05/05/2014, Caracas.

En el campo de la computación, se encuentran numerosos tipos de problemas, unos más difíciles de resolver que otros que son difíciles de computar por su amplio espacio de búsqueda, como por ejemplo el 15-puzzle, que es un juego que trata sobre ordenar 15 piezas que inicialmente tienen una configuración inicial, y en la menor cantidad de pasos posibles, siendo éste un problema NP-hard.

Dado el problema del 15-puzzle, no existe un algoritmo razonablemente eficiente para resolverlo, por lo que el ser humano tiene que recurrir a algoritmos que no son tan eficientes pero que garantizan encontrar la solución buscada. Tal es el caso de el algoritmo de A* e IDA*, que aunque tu tiempo de ejecución puede llegar a ser alto, con una buena heurística (admisible) se puede lograr que estos tiempos bajen considerablemente.

Existen diversas heurísticas que se utilizan para este problema en particular, tales como distancia manhattan, distancia manhattan con linear collision, pattern database (PDB), entre otras.

La heurística de la distancia de manhattan es una heurística admisible que basa su observación en la mínima distancia que tienen las piezas a la posición del goal, asumiendo que éstas pueden moverse sin colisionar con las demás fichas, al final la heurística total de una configuración del puzzle, es la suma de estas distancias.

La heurística de la distancia de manhattan con linear collision, es similar a la anterior, con la particularidad de que suma 2 unidades al total por cada dos piezas que sabemos que deben intercambiar sus posiciones para llegar a su posición final, esto es, cuando dos piezas ya están ubicadas en su fila (o columna) objetivo, pero están al revés, entonces sabemos que al menos necesitaremos dos pasos más para poder llegar al objetivo aunado a la distancia de manhattan de esas piezas.

Por último, la heurística pattern database, se concentra en escoger varios subconjuntos de las piezas del puzzle, tomando como piezas ignoradas a las demás y conseguir la mínima cantidad de pasos para colocar esas piezas en sus posiciones finales. Cuando estos subconjuntos son disjuntos, la heurística toma el nombre de disjoint pattern database y la heurística termina siendo la suma de las mínimas cantidades de pasos para colocar las piezas de cada patrón, ignorando a las demás.

El puzzle puede ser dividido cualquier forma, formando subconjuntos de piezas de 5-5-5, 6-6-3, entre otros.

A continuación, se mostrarán una serie de experimentos realizados recientemente, utilizando tanto el algoritmo de A* como el algoritmo de IDA*, y que muestran como una buena heurística puede ayudar a reducir el tiempo para la solución de problemas complejos, se mostrará como la heurística de distancia de manhattan es mucho menos potente que la heurística de manhattan con linear collision, así como también esta es mucho menos potente que las heurísticas de pattern database. Y también se observará como parece haber una tendencia, que a medida que un patrón, en la heurística de pattern database, sea un subconjunto más grande de las piezas parece tener una mejora en el tiempo de búsqueda.

Los resultados están basados en las primeras 14 instancias de las 100 casos que provee el profesor Blai Bonet:

Resultados utilizando el algoritmo de A*:

\Heurística Instancia\	Manhattan	Manhattan con linear collision	PDB 5-5-5	PDB 6-6-3
1	9.56	1.95	0.17	0.15
2	173.07	16.83	0.44	0.53
3	38.73	8.28	0.1	0.4
4	299.85	101.41	4.64	5.2
5	97.77	23.95	0.59	0.87
6	19.95	6.04	2.47	0.89
7	8.72	2.54	0.44	0.2
8	116.00	25.64	0.92	1.21
9	39.12	3.17	0.73	0.69
10	3.54	0.34	0.1	0.08
11	174.49	43.64	7.9	2.48
12	53.05	11.12	3.21	2.68
13	2.59	0.64	0.11	0.13
14	8.43	1.09	0.38	0.26
Tiempo promedio de los 100 casos	Algunos casos no lograban terminar por la cantidad de memoria requerida	34.1848	3.4947	2.4135

Resultados utilizando el algoritmo de IDA*:

\Heurística Instancia\	Manhattan	Manhattan con linear collision	PDB 5-5-5	PDB 6-6-3
1	2.43	0.76	0.08	0.08
2	37.79	13.09	0.15	0.13
3	4.01	3.04	0.08	0.21
4	127.34	86.78	2.31	5.15
5	16.95	7.41	0.14	0.15
6	2.42	2.1	0.88	0.33
7	2.78	0.71	0.11	0.05
8	25.39	20.73	0.15	0.19
9	7.72	1.29	0.51	0.31
10	0.97	0.14	0.02	0.04
11	43.2	83.36	2.36	1.43
12	23.64	5.25	1.94	2.46
13	0.11	0.15	0.02	0.02
14	3.18	1.23	0.26	0.09
Tiempo promedio de los 100 casos		25.9347	1.7829	1.2496