



Universidad Simón Bolívar
Departamento de Computación y Tecnología de la Información
CI5437 Inteligencia Artificial I
05 de mayo de 2014

15 - PUZZLE

Prof. Blai Bonet

Elaborado por:
Gamar Azuaje 10-10051
Wilmer Bandres 10-10055
Juan A. Escalante 10-10227

En el campo de la computación se encuentran numerosos tipos de problemas: unos más difíciles de resolver que otros. Esta diferencia de dificultad viene dada por el espacio de búsqueda de cada uno, que determina, según su amplitud, cómo se va a computar la solución, de manera que a mayor amplitud, mayor dificultad. Un ejemplo de un problema con amplio espacio de búsqueda es el 15-puzzle, un juego que trata sobre ordenar un rompecabezas de 15 piezas (que cuentan con una configuración inicial) en la menor cantidad de movimientos posible. Este problema califica como NP-Hard.

Dado el 15-puzzle como ejemplo, podemos decir que no existe un algoritmo razonablemente eficiente para resolverlo, por lo que el ser humano tiene que recurrir a algoritmos que, si bien no garantizan la eficiencia, encuentran la solución deseada. En este proyecto, los algoritmos a utilizar son A* e IDA*. Es importante aclarar sobre su uso que, aunque el tiempo de ejecución puede llegar a ser alto, con una heurística admisible bien seleccionada, se puede lograr que estos tiempos se reduzcan considerablemente.

Existen diversas heurísticas que se pueden elegir para este problema en particular, algunas de ellas son:

- La heurística de la Distancia Manhattan: 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 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 Manhattan de esas piezas.

- Por último, la heurística Pattern Database (PDB): se concentra en escoger varios subconjuntos de las piezas del puzzle, tomando como piezas ignoradas a las demás y consiguiendo 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 como tal 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 mostrará una serie de experimentos realizados recientemente, utilizando tanto el algoritmo de A* como el algoritmo de IDA*, y que muestran cómo una buena heurística puede ayudar a reducir el tiempo para la solución de problemas complejos. Se observará cómo la heurística de la Distancia 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 (PDB). También se apreciará cómo parece haber una tendencia que indica que a medida que un patrón (en la heurística de Pattern Database) sea un subconjunto más grande de las piezas, parece existir una mejora en el tiempo de búsqueda.

Esta conclusión a la que hemos llegado se base principalmente en la interacción del movimiento de varias fichas simultáneamente, mientras Manhattan no toma en consideración ningún tipo de colisión, las heurísticas basadas en PDBs divide el problema en secciones que conviven buscando el goal.

Análisis de PDB

15-puzzle 5 fichas	
Permutaciones a visitar	$16*15*14*13*12*11 = 5765760$
Número de entradas	$16*15*14*13*12 = 524160$

15-puzzle 6 fichas	
Permutaciones a visitar	$16*15*14*13*12*11*10 = 576557600$
Número de entradas	$16*15*14*13*12*11 = 5765760$

15-puzzle 7	
Permutaciones a visitar	$16*15*14*13*12*11*10*9 = 518918400$
Número de entradas	$16*15*14*13*12*11*10 = 57657600$

15-puzzle 8	
Permutaciones a visitar	$16*15*14*13*12*11*10*9*8 = 4151347200$
Número de entradas	$16*15*14*13*12*11*10*9 = 518918400$

24-puzzle 4	
Permutaciones a visitar	$25*24*23*22*21 = 6375600$
Número de entradas	$25*24*23*22 = 303600$

24-puzzle 5	
Permutaciones a visitar	$25*24*23*22*21*20 = 127512000$
Número de entradas	$25*24*23*22*21 = 6375600$

24-puzzle 6	
Permutaciones a visitar	$25*24*23*22*21*20*19 = 2422728000$
Número de entradas	$25*24*23*22*21*20 = 127512000$

En las tablas anteriores se presentan los cálculos necesarios por sección del PDB. Particularmente, el PDB 7-8 para el 15-puzzle y el PDB 6-6-6-6 para el 24-puzzle, tienen un espacio de búsqueda demasiado extenso para poder ser almacenados en la memoria sin una técnica más avanzada de la que se está utilizando, por lo que se decidió manejar únicamente las particiones de 5-5-5 y 6-6-3 para el 15-puzzle y las de 5-5-5-5-4 para el 24-puzzle.

El método utilizado para almacenar los valores de los costos hasta el goal del PDB, viene dado por archivos ubicados en la carpeta “pddb”. En ella también se encuentran las divisiones seleccionadas en cada caso, utilizando una función de Hash perfecta, basada en una función biyectiva que incluye combinatorios y matemática discreta. Para la carga de los PDBs en memoria, se utiliza este precálculo, leído desde un archivo para llenar un arreglo en el que cada posición representa un patrón del tablero.

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	62.5448	25.9347	1.7829	1.2496

Los thresholds, que dio IDA* fueron:

\ Heurística Casos\	Manhattan	Manhattan con linear collision	PDB 5-5-5	PDB 6-6-3
1	41 43 45 47 49 51 53	41 43 45 47 49 51 53	45 47 49 51 53	45 47 49 51 53
2	41 43 45 47 49 51 53 55 57	41 43 45 47 49 51 53 55 57	47 49 51 53 55 57	47 49 51 53 55 57
3	43 45 47 49 51 53 55	43 45 47 49 51 53 55	45 47 49 51 53 55	45 47 49 51 53 55
4	41 43 45 47 49 51 53 55 57 59	41 43 45 47 49 51 53 55 57 59	45 47 49 51 53 55 57 59	45 47 49 51 53 55 57 59
5	42 44 46 48 50 52 54 56	42 44 46 48 50 52 54 56	44 46 48 50 52 54 56	44 46 48 50 52 54 56
6	42 44 46 48 50 52 54 56	42 44 46 48 50 52 54 56	44 46 48 50 52 54 56	44 46 48 50 52 54 56
7	36 38 40 42 44 46 48 50 52	36 40 42 44 46 48 50 52	42 44 46 48 50 52	44 46 48 50 52
8	30 32 34 36 38 40 42 44 46 48 50 52	30 32 34 36 38 40 42 44 46 48 50 52	42 44 46 48 50 52	40 42 44 46 48 50 52
9	32 34 36 38 40 42 44 46 48 50	32 36 38 40 42 44 46 48 50	40 42 44 46 48 50	40 42 44 46 48 50
10	32 34 36 38 40 42 44 46	32 36 38 40 42 44 46	38 40 42 44 46	40 42 44 46
11	43 45 47 49 51 53 55 57 59	43 45 47 49 51 53 55 57 59	47 49 51 53 55 57 59	49 51 53 55 57 59
12	43 45 47 49 51 53 55 57	43 45 47 49 51 53 55 57	45 47 49 51 53 55 57	45 47 49 51 53 55 57
13	35 37 39 41 43 45	35 37 39 41 43 45	37 39 41 43 45	39 41 43 45
14	36 38 40 42 44 46	36 38 40 42 44 46	36 38 40 42 44 46	36 38 40 42 44 46

Resultados de los tiempos de IDA* para el 24-puzzle:

\ Heurística Casos\	Instancia	PDB 5-5-5-5-4
1	6 5 3 4 9 1 8 2 18 13 0 12 11 23 14 10 15 7 16 19 20 21 17 22 24	0.01
2	0 5 6 3 4 10 2 1 7 8 16 15 13 18 12 20 11 14 19 9 22 17 21 23 24	0.4
3	10 1 7 4 9 2 6 15 3 13 11 5 12 17 0 20 16 18 14 8 21 22 19 23 24	1.52
4	1 6 2 4 9 10 5 11 8 3 20 15 19 13 18 7 12 23 16 17 21 22 0 24 14	0.05
5	1 0 4 13 9 10 3 6 19 14 21 16 5 2 8 11 15 12 23 18 20 22 17 7 24	5.69
6	10 2 7 9 3 13 15 6 4 14 16 0 5 1 8 11 12 18 24 19 20 21 17 22 23	40.55
7	5 0 2 8 4 10 12 7 9 14 15 1 6 3 13 20 11 17 18 19 21 16 22 23 24	1.54
8	1 7 3 8 4 0 6 5 13 9 21 11 12 14 18 10 2 15 17 16 20 22 23 24 19	1.87
9	1 13 7 9 2 11 6 3 4 14 16 15 12 19 5 20 17 18 8 0 21 22 10 23 24	89.29
10	12 0 9 7 8 1 5 6 3 4 11 10 17 13 14 15 2 16 18 19 20 21 22 23 24	3.61
11	5 1 7 8 3 10 19 6 2 0 15 23 11 4 9 20 16 12 17 14 21 22 18 13 24	57.34
12	2 13 4 0 9 1 6 8 16 12 5 15 3 7 24 11 10 21 18 14 20 22 19 23 17	48.37
13	1 10 2 3 4 5 12 15 9 14 17 0 7 6 19 16 11 22 8 13 20 21 23 18 24	0.27
14	6 3 12 8 4 1 7 11 2 9 5 23 13 0 14 15 20 21 18 19 16 10 22 17 24	7.06
15	5 3 4 8 9 6 1 7 2 14 21 15 10 12 13 0 17 16 18 19 11 20 22 23 24	71.61
16	5 1 8 2 4 6 7 13 16 9 10 11 17 3 19 15 14 24 22 23 20 0 21 12 18	2.59

17	5 1 2 3 4 11 10 7 8 9 15 6 13 22 14 23 16 0 18 19 20 21 12 17 24	0.96
18	11 1 2 8 4 16 3 6 9 14 10 17 12 13 19 20 7 5 18 0 21 15 22 23 24	0.7
19	6 5 10 3 4 11 7 12 2 13 15 16 18 9 8 17 1 0 19 14 20 21 22 23 24	2.02
20	6 8 4 14 9 1 5 7 3 2 15 10 18 24 13 0 11 17 12 23 20 16 21 22 19	37.94
21	5 3 12 8 13 2 11 4 1 9 10 16 6 7 18 20 15 17 19 14 21 0 22 23 24	11.69
22	5 1 2 9 3 0 10 11 7 4 23 6 13 8 14 15 16 22 17 24 20 21 12 19 18	537.94
23	1 2 8 13 9 3 10 12 19 4 15 5 7 6 14 0 21 11 17 18 16 20 22 23 24	0.29
24	11 1 6 3 4 5 16 0 2 9 10 7 12 8 14 15 21 17 13 19 20 23 22 18 24	278.3
25	6 3 13 9 7 1 2 8 11 4 5 10 20 0 14 21 16 17 12 18 15 22 23 24 19	5.09

Otros resultados:

- La cantidad de nodos generados por manhattan es en promedio 4609610 por segundo.
- La cantidad de nodos generados por manhattan pairwise es en promedio 1670000 por segundo.
- La cantidad de nodos generados por PDB 5-5-5 es en promedio 1390000 por segundo.
- La cantidad de nodos generados por PDB 6-6-3 es en promedio 1480000 por segundo.
- La cantidad de nodos generados por PDB 5-5-5-5-4 es en promedio 650000 por segundo.