
Resolución de problemas de búsqueda

Memoria de Prácticas de Inteligencia Artificial

Segunda Entrega

26 de noviembre de 2007

Autores:

Mariano Cabrero Canosa *cicanosa@udc.es*

Elena Hernández Pereira *elena@udc.es*

Directorio de entrega: XXXXX

Resumen. El 8-puzzle es un pequeño juego de mesa para un único jugador que consiste en 8 piezas cuadradas numeradas del 1 al 8 y un espacio vacío en un tablero de 3 x 3. El objetivo del juego es alcanzar una disposición determinada de las piezas realizando sólo movimientos permitidos. Esta práctica trata de dar solución al juego formulándolo como un problema de búsqueda. Se implementan distintos algoritmos de búsqueda, ciega e informada, se evalúan y se extraen las conclusiones pertinentes.

Índice

| | |
|--|----------|
| 1. Resultados | 4 |
| 1.1. Ejemplos de ejecución | 4 |
| 1.2. Caracterización de heurísticas | 4 |
| 1.3. Caracterización de métodos | 5 |
| 2. Discusión | 5 |
| 2.1. Conclusiones acerca de las heurísticas (apartado 1.2) | 5 |
| 2.2. Ventajas e inconvenientes de los métodos de búsqueda (apartado 1.3) | 6 |
| 2.3. Heurística que sobreestime | 6 |

1. Resultados

1.1. Ejemplos de ejecución

Aquí se mostraría la traza completa de ejecución sobre al menos un ejemplo de cada uno de los tres algoritmos implementados (búsqueda ciega, ascensión a colinas y A^). La traza seguirá fielmente las especificaciones del enunciado práctico.*

1.2. Caracterización de heurísticas

Aquí se expone la calidad de las heurísticas empleadas en función de la longitud del camino de solución óptimo.

Tabla 1: Tabla comparativa de rendimiento de heurísticas

| Estado s_i | h_1 | h_2 | $h^*(s)$ |
|--------------|-------|-------|----------|
| 6, 1, 4 | 7 | 12 | 14 |
| 2, □, 3 | | | |
| 7, 5, 8 | | | |
| □, 8, 4 | 8 | 18 | 18 |
| 2, 7, 3 | | | |
| 6, 1, 5 | | | |
| 1, 2, 6 | 4 | 6 | 20 |
| 5, □, 4 | | | |
| 7, 8, 3 | | | |
| 1, 8, □ | 7 | 10 | 24 |
| 6, 4, 3 | | | |
| 7, 5, 2 | | | |
| 8, 1, 4 | 8 | 16 | 26 |
| 6, □, 2 | | | |
| 5, 7, 3 | | | |
| 2, 6, 4 | 8 | 16 | 26 |
| 8, 3, 1 | | | |
| □, 5, 7 | | | |
| 6, 5, 3 | 7 | 14 | 28 |
| 8, 4, 2 | | | |
| □, 1, 7 | | | |
| ... | ... | ... | ... |
| Promedio | 7 | 13,14 | 22,28 |

donde los ejemplos en particular se han expresado como un vector y $h_1(s_i)$ y $h_2(s_i)$ son los valores correspondientes a las heurísticas propuestas (Fichas mal colocadas y Manhattan) en el estado s_i . Para todos los ejemplos se ha utilizado como estado meta:

| | | |
|---|---|---|
| 1 | 2 | 3 |
| 4 | 5 | 6 |
| 7 | 8 | |

1.3. Caracterización de métodos

Aquí se compara el rendimiento de los métodos de búsqueda ciega e informados con respecto al número de nodos generados en función de la longitud del camino. Es necesario, por tanto, diseñar un número suficiente de problemas aleatorios con distintas longitudes del camino solución, resolverlos con los distintos métodos estudiados (búsqueda ciega, ascensión a colinas y A con ambas heurísticas) y promediar sus resultados en función de dicha longitud de camino.*

Para construir la tabla de resultados, se han generado N problemas aleatorios con longitudes de solución entre 2 y 24 (k por cada una) ¿Cómo generar N problemas con estas propiedades? El estado meta no puede alcanzarse desde cada uno de los $9!$ posibles estados. Un modo de generar estados aleatorios es empezar desde la configuración meta y “barajar” las piezas aleatoriamente realizando movimientos válidos. Ya que A* encuentra la solución más corta, se han elegido k de los problemas en los que A* encontró una solución a profundidad 2, k a profundidad 4, y así sucesivamente para utilizarlos en el resto de algoritmos. Nótese que los resultados son totalmente inventados, pero están dentro de una cierta “lógica”.

Tabla 2: Tabla comparativa de rendimiento de métodos de búsqueda

| d | Coste de la búsqueda | | | | | Factor Ramificación Eficaz | | | | |
|----|----------------------|----------|----------|--------|--------|----------------------------|----------|----------|--------|--------|
| | Ciega | Hill(h1) | Hill(h2) | A*(h1) | A*(h2) | Ciega | Hill(h1) | Hill(h2) | A*(h1) | A*(h2) |
| 2 | 7 | 3 | 3 | 6 | 6 | 2,45 | 1 | 1 | 1.79 | 1.79 |
| 4 | 50 | 8 | 6 | 13 | 12 | 2,87 | 1,24 | 1,09 | 1.48 | 1.45 |
| 6 | 354 | 15 | 12 | 20 | 18 | 2,73 | 1,25 | 1,18 | 1.34 | 1.30 |
| 8 | 2.506 | 22 | 20 | 39 | 25 | 2,80 | 1,21 | 1,19 | 1.33 | 1.24 |
| 10 | 17.734 | 31 | 28 | 93 | 39 | 2,79 | 1,19 | 1,18 | 1.38 | 1.22 |
| 12 | 125.481 | 64 | 60 | 227 | 73 | 2,78 | 1,24 | 1,23 | 1.42 | 1.24 |
| 14 | - | 102 | 94 | 539 | 113 | - | 1,24 | 1,23 | 1.44 | 1.23 |
| 16 | - | 156 | 142 | 1301 | 211 | - | 1,24 | 1,23 | 1.45 | 1.25 |
| 18 | - | 301 | 289 | 3056 | 363 | - | 1,26 | 1,25 | 1.46 | 1.26 |
| 20 | - | 508 | 468 | 7276 | 676 | - | 1,26 | 1,26 | 1.47 | 1.27 |
| 22 | - | 1104 | 1024 | 18094 | 1219 | - | 1,28 | 1,28 | 1.48 | 1.28 |
| 24 | - | 1316 | 1189 | 39135 | 1641 | - | 1,26 | 1,26 | 1.48 | 1.26 |

La cantidad de nodos examinados por un algoritmo informado depende no sólo de la heurística sino de la profundidad en la que se encuentra la solución y la máxima profundidad del árbol de búsqueda del problema, el orden de aplicación de los operadores, etc. Por ello, una buena medida cuantitativa para comparar dos heurísticas es comprobar su *factor de ramificación efectivo*: el factor de ramificación de un árbol uniforme de profundidad d que contenga n nodos. Recordemos que un árbol uniforme es aquél en el cual todos los nodos expandidos (no hojas) tienen el mismo factor de ramificación y la profundidad de todos los caminos de búsqueda es constante.

2. Discusión

2.1. Conclusiones acerca de las heurísticas (apartado 1.2)

Aquí debemos realizar una pequeña discusión sobre los resultados obtenidos por las heurísticas para los ejemplos planteados en relación a la función h^ , función de coste*

óptimo. Cuanto más se acerquen los resultados a estos valores, más precisa será la heurística.

Los resultados mostrados en la Tabla 1 vienen a demostrar en primer lugar que ninguna de las heurísticas sobrestiman el coste óptimo, algo que ya sabíamos teóricamente. En segundo lugar demuestran que una heurística está más informada que la otra, es decir, que si n es un estado del espacio de estados del problema, se cumple que

$$\forall n, h_1(n) \geq h_2(n) \text{ o bien } \forall n, h_2(n) \geq h_1(n) \quad (1)$$

Así la heurística “más informada” es la distancia Manhattan, h_2 , puesto que *domina* a la heurística “fichas mal colocadas”. La razón de este comportamiento es que la heurística Manhattan ha sido obtenido considerando menos restricciones, es decir, *relajando* el problema original.

En tercer y último lugar, una consecuencia directa que se deriva de lo anterior es la mayor eficiencia de los procesos de búsqueda que utilizan la distancia Manhattan, en el sentido de que expandirán menos nodos en promedio. Esto sin embargo, lo comprobaremos en el apartado siguiente.

En consecuencia ...

2.2. Ventajas e inconvenientes de los métodos de búsqueda (apartado 1.3)

Aquí debemos realizar una pequeña discusión sobre el comportamiento de los métodos de búsqueda sobre los ejemplos planteados.

A la vista de los resultados de la tabla 2 podemos observar un comportamiento lineal de los algoritmos en cuanto al número de nodos procesados. Como cabría esperar, el algoritmo de búsqueda preferente por amplitud es la estrategia que más nodos genera, incluso contemplando la posibilidad de no repetir estados (el primer estado alcanzado será el menos costoso porque el coste de aplicar un operador es siempre 1) y eliminando los ciclos. A medida que la longitud del camino solución se incrementa, también lo hace exponencialmente, el número de nodos generados.

Por otra parte, cabe resaltar los buenos resultados del algoritmo de ascensión a colinas. Su rendimiento se encuentra entre el algoritmo A^* y el algoritmo de anchura. Su principal inconveniente es que no es una estrategia completa ni óptima, algo que teóricamente ya sabíamos, por cuanto no es un algoritmo diseñado para obtener un camino, sino sólo una solución, que en este caso ya conocíamos.

Por último, destacar los resultados del algoritmo A^* en cuanto a las heurísticas. Vemos que, ciertamente, con la heurística más informada, h_2 , el algoritmo genera menos nodos en promedio. esto también está justificado observando los valores del factor de ramificación efectivo, para el que la heurística Manhattan obtiene mejores resultados (más cercanos a 1) lo cual viene a significar que no “expande” tanto su búsqueda por el árbol.

2.3. Heurística que sobreestime

Se propone la siguiente función heurística:

$$h_3(n) = h_1(n) + h_2(n) \quad (2)$$

| | | | |
|----------------|----------------|----------------|--|
| ¹ 4 | ¹ 1 | ¹ 2 | $h_3(n) = h_1 + h_2 =$ 16 Coste óptimo = 8 |
| ¹ 7 | | ¹ 3 | |
| ¹ 8 | ¹ 5 | ¹ 6 | |

Figura 1: Ejemplo de heurística que sobrestima

donde $h_1(n)$ es la heurística de las “fichas mal colocadas” y $h_2(n)$ es la heurística Manhattan .

Para el ejemplo de la figura 1 la heurística devuelve un valor de $h_3(n) = 8 + 8 = 16$. Analizando el comportamiento de esta nueva heurística podemos concluir que sobrestimará en la mayor parte de situaciones. La explicación a este hecho es que por una lado, la heurística Manhattan precisa mucho su valor cuando el número de pasos del camino solución es pequeño, y en este caso ofrece el valor exacto; y por otro lado, la heurística “mal colocadas” devuelve el número mínimo de pasos, que en este caso coincide con el camino óptimo.

El comportamiento del algoritmo A* utilizando la heurística

Referencias

- [1] Storey, W.E. Notes on the 15 puzzle. I. En *American Journal of Mathematics*, vol. 2 (4), pp. 399–404, 1879.
- [2] Johnson, W. W. Notes on the 15 Puzzle. I. En *American Journal of Mathematics*, vol. 2 (4), pp. 397–399, 1879.
- [3] A. Reinefeld. Complete Solution of the Eight-Puzzle and the Benefit of Node Ordering in IDA* En *International Joint Conference on Artificial Intelligence*, pp. 248-253, 1993.
<http://citeseer.ist.psu.edu/article/reinefeld93complete.html>
- [4] V. Moret, A. Alonso, M. Cabrero, B. Guijarro, E. Mosqueira. *Fundamentos de Inteligencia Artificial (2ª Ed)*. Servicio de Publicaciones, UDC, 2000
- [5] N. Nillson. *Inteligencia artificial: una nueva síntesis*. McGraw-Hill, 2001
- [6] S. Russell, P. Norvig. *Inteligencia Artificial: Un enfoque moderno*. Prentice-Hall, 2004.