



SISTEMAS DE INTELIGENCIA ARTIFICIAL

TRABAJO PRÁCTICO N°1

Métodos de búsqueda no informados e informados

Autores:

Pablo Ballesty - 49359

Nicolás Magni - 48008

Guillermo Liss - 49282

20 de marzo de 2012

Resumen

El objetivo del presente informe es detallar el diseño e implementación de un Sistema de Producción para la resolución del juego *DeepTrip* (<http://www.minijuegos.com/Deep-Trip/10185>) mediante métodos de búsqueda no informados e informados. La finalidad de este informe es analizar cada método en particular, y estudiar la *performance* de cada uno para diferentes entradas iniciales. También para los métodos informados se ensayan diferentes heurísticas, con el fin de descubrir cuál resulta mejor en cada caso.

1. El problema

DeepTrip se compone de un tablero de $n \times m$ casilleros con fichas de k colores. El objetivo del juego es realizar rotaciones de las filas formando grupos de 3 o más fichas adyacentes de un mismo color, los cuales van desapareciendo, para lograr dejar el tablero vacío.

2. Diseño y Modelado del problema

El tablero se modela mediante una matriz de números enteros que representan los distintos colores. No se sigue un diseño totalmente *OOP* ya que esto degradaría la *performance* al requerir del trabajo del *Garbage Collector* al eliminar fichas.

Con respecto al problema, los estados del mismo van a estar formados por una instancia del tablero antes descripto, entendiendo como instancia al tablero con los colores correspondientes luego de aplicar una regla, con la excepción del estado inicial que no se deriva de la aplicación de una regla sino que contiene los datos iniciales. Se considera como estado final al tablero completamente vacío.

En resumen la definición formal del problema es la siguiente:

- Estado inicial: Tablero completamente lleno, sin grupos de fichas eliminables.
- Conjunto de posibles acciones: Rotar fila i hacia la derecha, j veces, donde $0 \leq i < n$ y $0 < j < m$.
- Modelo de transición: Luego de aplicar la regla se obtendrá un nuevo tablero sin aquellas fichas que hayan formado grupos de 3 o más fichas de un mismo color.
- Condición de solución: Tablero vacío.
- Función de costo de ruta: 3.2.

Notar que no se rota a izquierda, ya que se aprovecha la equivalencia de la rotación derecha con la misma. También, cabe aclarar que una regla se considera *no aplicable* a un estado, si la fila a rotar se encuentra vacía o el tablero no tiene solución¹.

Luego de aplicar una regla a un estado, se eliminan las fichas que formen grupos de 3 o más fichas adyacentes, después de haber eliminado todos los grupos, se aplica *gravedad* la cual hace desplazar hacia arriba las fichas situadas por debajo de espacios vacíos.

Una de las decisiones de implementación que vale la pena destacar es que se modificó el *engine* que provee la cátedra, agregando un *HashSet* donde se guardan todos los estados explotados, con el fin de evitar estados repetidos.

3. Heurísticas

Para todas las funciones heurísticas que se plantean se considera:

- T como la cantidad total de fichas del juego, $n \times m$.
- $B(n)$ como la cantidad de espacios vacíos en el tablero contenido en el estado del nodo n .
- $P(n)$ como la cantidad de *clusters* de dos fichas en el tablero contenido en el estado del nodo n . Estos pueden estar tanto en forma horizontal como vertical.
- $C(n)$ como la cantidad de colores en el tablero contenido en el estado del nodo n .
- $M(n)$ como la cantidad de movimientos realizados desde el estado inicial a estado contenido en el nodo n .
- $C_e(n)$ como la cantidad de colores eliminados en el tablero del estado contenido en el nodo n , con respecto al tablero contenido en el nodo padre de n .

¹Si para algún color presente en el tablero la cantidad de fichas de ese color es 1 o 2 resulta imposible eliminarlas, por lo que el tablero no tendrá solución

3.1. Heurísticas propuestas

3.1.1. Primera propuesta

$$h_0(n) = \frac{1}{(B(n) + 1) \cdot M(n)} \quad (1)$$

En esta propuesta se busca favorecer a los estados que obtuvieron más espacios vacíos en menos cantidad de movimientos. Esta estrategia resulta favorable para las primeras iteraciones, en donde es poco probable que el problema sea resuelto, y resulta muy costoso analizar todas los posibles movimientos de eliminación. De esta forma se llegaría a un tablero con menos fichas de forma mas acelerada, en donde no resulta tan costoso analizar todas las opciones.

3.1.2. Segunda propuesta

$$h_1(n) = \frac{(T - B(n))}{2(P(n) + 1)} \quad (2)$$

En esta propuesta se busca favorecer aquellos tableros donde hay gran probabilidad de formar grupos de colores eliminables. Es por eso que se utiliza la cantidad de grupos de dos fichas del mismo color, ya que es probable encontrar un movimiento para convertirlo en un grupo eliminable².

3.1.3. Tercera propuesta

$$h_2(n) = \frac{C(n)}{M(n)} \quad (3)$$

En esta propuesta se busca favorecer aquellos tableros con menor diversidad de colores³. Esta estrategia resulta favorable en iteraciones más avanzadas, ya que al comenzar, es poco probable que la diversidad de colores cambie. De esta forma, a largo plazo se favorecen los tableros con menor cantidad de colores, en los cuales hay más probabilidad de solución.

3.1.4. Cuarta propuesta

$$h_3(n) = \min : \left\{ \frac{\#C_i(n)}{M(n) \cdot C_e(n)}, \dots, \frac{\#C_k(n)}{M(n) \cdot C_e(n)} \right\} \text{ con } i \text{ tal que } C_k > 0 \quad (4)$$

Esta heurística favorece a la eliminación de un color a la vez, y en caso de eliminar completamente uno, el de mayor cantidad de fichas.

3.1.5. Quinta propuesta

$$h_4(n) = h_0(n) + h_2(n) \quad (5)$$

En esta heurística se busca aprovechar las ventajas de h_0 y h_2 . La primera resultaba buena para las primeras iteraciones, y la segunda para las últimas.

3.2. Función de costo

La función de costo que se utiliza es:

$$g(n) = M(n)$$

Para el caso de la estrategia de búsqueda A*, en donde se utilizan en conjunto la función de costo y la función heurística, la función de evaluación $f_k(n)$ asociada a la función heurística $h_k(n)$ se define de la siguiente forma:

$$f_k(n) = \{ h_k(n) + g(n) \quad \text{si } k \text{ es } 0, 1, 2, 3, 4 \quad (6)$$

²**Grupo eliminable:** grupo de 3 o más fichas adyacentes del mismo color.

³**Diversidad de colores:** cantidad de colores distintos en el tablero, independiente de la cantidad de fichas del color.

4. Resultados

Para analizar los métodos se tomaron dos grupos distintos tableros.

- El primero, *easy*, son tableros de 4 x 4 con 3 colores.
- El segundo, *hard*, son tableros de 6 x 5 con 4 colores.

4.1. Resultados para A*

Tamaño	Heurística	Nodos explotados		Profundidad		Nodos en frontera		Tiempo (mili seg)	
		Media	Desvío	Media	Desvío	Media	Desvío	Media	Desvío
Easy	h_0	133.2	185.21	11.3	11.70	57.0	49.74	28.6	42.53
	h_1	59.90	144.96	5.0	3.01	55.4	87.38	7.9	20.84
	h_2	84.2	143.55	8.8	6.59	46.7	28.3	5.3	9.55
	h_3	87.3	146.51	8.3	6.34	36.3	18.76	4.6	10
	h_4	97.79	171	7.4	6.36	41.3	31.12	4.4	8.95
Hard	h_0	1837.8	2470.16	21.4	8.98	248.8	77.31	463.2	778.08
	h_1	1079.6	1673.35	7.2	1.78	150.2	129.11	166	297.47
	h_2	1118	558.75	13.4	5.45	157.4	49.48	110.8	72.9
	h_3	940.0	703.66	17.6	8.14	143.8	33.28	84.4	78.65
	h_4	293.8	374.86	11	4.3	144.8	37.13	30.8	38.27

4.2. Resultados para Greedy Search

Tamaño	Heurística	Nodos explotados		Profundidad		Nodos en frontera		Tiempo (mili seg)	
		Media	Desvío	Media	Desvío	Media	Desvío	Media	Desvío
Easy	h_0	133.2	185.21	11.3	11.8	57	49.75	28.5	43.27
	h_1	82.2	184.85	6	4	29.9	10.15	5.6	13.37
	h_2	84.2	143.55	8.8	6.59	46.7	28.3	3.9	7.26
	h_3	87.3	146.5	8.3	6.35	36.3	18.80	4.3	8.49
	h_4	97.79	171	7.4	6.36	41.3	31.12	4.4	8.82
Hard	h_0	1837.8	2470.16	21.4	8.98	248.8	77.31	398.2	653.43
	h_1	818.6	1058.35	7.2	1.78	91.6	24.78	92.6	143.07
	h_2	1118	558.75	13.4	5.45	157.4	49.48	104.8	71.42
	h_3	940	703.66	17.6	8.15	143.8	33.28	112.6	126.37
	h_4	293.8	374.86	11	4.3	144.8	37.13	27	36.11

4.3. Resultados para métodos no informados

Tamaño	Heurística	Nodos explotados		Profundidad		Nodos en frontera		Tiempo (mili seg)	
		Media	Desvío	Media	Desvío	Media	Desvío	Media	Desvío
Easy	DFS	348.1	591.38	19.3	10.69	123	73.89	48	70.35
	BFS	2109.3	5297.51	3.8	1.13	4190.7	9360.03	102.7	196
	ID	249.2	542.19	18.9	22	38.8	56.24	65.1	114.16
Hard	DFS	3841.8	4283.91	41.4	22.27	464.8	173.60	311.2	294.31
	BFS	23404	22831.4	4	0	284737	249537.36	2834.8	2167.53
	ID	1266.4	2439.05	21.2	4.86	230.8	148.85	237.8	303.88

5. Conclusiones

5.1. Métodos no informados

Mediante el análisis de los resultados para los métodos no informados podemos ver que para un mismo tablero, varían mucho el tiempo de resolución, nodos explotados, y profundidad. Sin embargo, no se puede hablar de la conveniencia de un método sobre otro ya que esto depende del tablero inicial y no se aplica ninguna estrategia de juego para la explosión de nodos, por lo tanto encontrar una solución de forma rápida va a depender de que la solución aparezca en los nodos que primero explota cada método. Por la naturaleza del problema, cada tablero suele tener más de una solución y los mismos suelen resolverse al aplicar una sucesión de varias reglas. Es por esto que podemos ver que el algoritmo de búsqueda DFS (e ID) es el que ha llegado a encontrar una solución de forma más rápida, ya que aplica varias reglas a un mismo tablero, antes de probar con otro, y BFS va aplicando pocas reglas a todos los tableros.

5.2. Métodos informados

5.2.1. Heurísticas

Con respecto a las heurísticas es importante ver la complejidad del cálculo de los datos utilizados en cada una. A continuación se resumen los mismos (considerando n como el tamaño del tablero):

- T se define al inicio, por lo tanto su cálculo es de $O(1)$.
- $B(n)$ se calcula en cada caso y es de $O(n)$.
- $P(n)$ se calcula en cada caso, y cabe notar que aunque parezca un cálculo complejo al necesitarlo siempre luego de aplicar la eliminación de fichas y gravedad, solo pueden quedar clusters de tamaño 1 y 2, por lo tanto puede realizarse en $O(n)$.
- $C(n)$ se calcula en cada caso, aunque el tablero guarda un arreglo donde mantiene la cantidad de fichas para cada color y éste se actualiza al ejecutar las eliminaciones, por lo tanto su utilización demanda $O(1)$.
- $M(n)$ es la profundidad del nodo en el árbol, y es parte de la estructura del mismo, por lo tanto demanda $O(1)$.
- $C_e(n)$ este caso depende del nodo padre, aunque para su cálculo se utilizan los arreglos de cantidad de fichas por color, lo cual no demanda gran complejidad.

Como todas las heurísticas combinan en su cálculo los datos anteriores mediante operaciones aritméticas, es significativo distinguir que la complejidad del cálculo de cada heurística comparada con la complejidad general de los métodos de resolución resulta despreciable. Y éste resulta uno de los factores claves por lo cual los métodos informados arrojan resultados en tiempos muy aceptables.

5.2.2. Análisis de resultados

Para los tableros del grupo *easy*, no se nota gran diferencia entre los resultados obtenidos mediante las diferentes heurísticas. Esto era esperado, ya que los tableros de este grupo son de fácil solución, y no influyen mucho las estrategias que se toman para la explosión de nodos.

Para los tableros del grupo *hard*, se puede notar que las heurísticas h_3 y h_4 encuentran la solución en menor tiempo, en la mayoría de los casos el tiempo es el 50 % de las demás heurísticas propuestas.

Particularmente, h_4 explota hasta 4 veces menos nodos que h_3 , por lo que se podría concluir que es la heurística más conveniente. A modo de experiencia se evaluó h_3 y h_4 con A* en tableros similares al del juego real pero con dos colores menos, es decir dimensión 10 x 8, con 4 colores. De esta prueba, se obtuvieron los siguientes resultados, que confirman la conveniencia de estas heurísticas.

Heurística	Nodos explotados	Profundidad	Tiempo (mil seg)
h_3	6478	11	4011
h_4	3905	48	2764