Sistemas de Inteligencia Artificial
Eternity 2

Algoritmos de búsqueda informados y no informados

Cristian Pereyra – Esteban Pintos – Matías De Santi

Eternity 2

- Tablero cuadrado de *n* piezas de ancho
- Las piezas tienen un color en cada lado
- El color gris en cualquier pieza debe ser adyacente a un borde
- Dos piezas adyacentes deben tener el mismo color en el borde que comparten

Cristian Pereyra - Esteban Pintos - Matías De Santi

Definición del problema

- **Estado inicial**: tablero vacío
- **Conjunto de posibles acciones**: dado un estado, se puede colocar sobre el tablero cualquier ficha siempre y cuando esta no haya sido colocada antes y no haya una ficha en la posición que se quiere colocar.
- **Modelo de transición**: colocar una pieza en una posición del tablero da como resultado un nuevo tablero con todas las fichas que tenía el anterior más la que ha sido colocada
- **Condición de solución**: un tablero es solución del problema si las fichas con color gris han sido colocadas con el color gris del lado del borde y para toda ficha adyacente en el tablero, los colores de los bordes que comparten son iguales.

Cristian Pereyra - Esteban Pintos - Matías De Santi

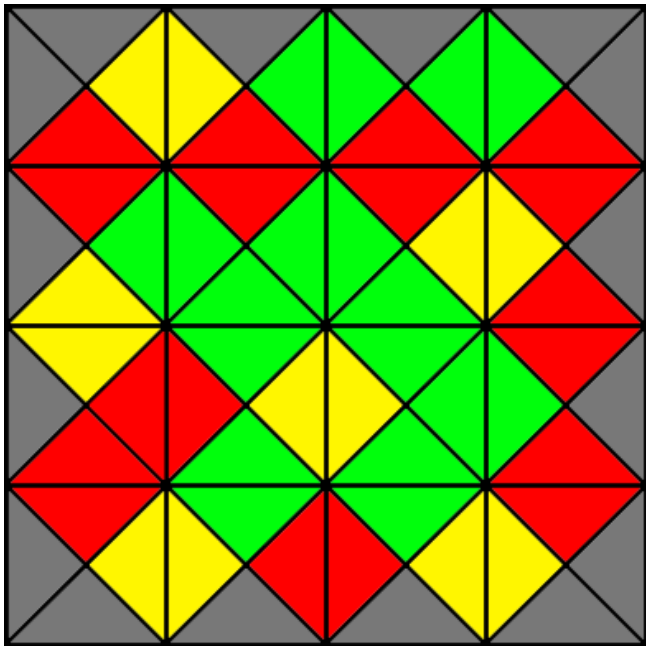
Representación del problema
Estados y piezas

Cada estado guarda:

- La pieza colocada a partir del estado anterior y la posición.
- Una referencia al estado padre

Cada pieza guarda:

- El color que corresponde a cada borde
- Si esta rotada y cuántas veces



Cristian Pereyra - Esteban Pintos - Matías De Santi

Representación del problema
Acciones posibles

Reglas del juego

- Colocar una pieza *p* en una posición (i,j)
- Rotar una pieza

Reglas que representamos

- Colocar una pieza *p* en una posición (i,j). Al crear las reglas para *p* se crean las reglas para las 3 posibles rotaciones de *p*

Cristian Pereyra - Esteban Pintos - Matías De Santi

Representación del problema
Modelo de transición

- Al aplicar una regla a un nodo, se obtiene otro nodo en cuyo estado se encuentra el mismo tablero que en el estado padre pero con la regla aplicada

Cristian Pereyra - Esteban Pintos - Matías De Santi

Representación del problema
Condición de terminación

El tablero debe estar completo y las piezas deben cumplir las siguientes condiciones:

- Si la pieza contiene color 0, entonces este color debe estar pegado a un borde del tablero.
- Las piezas adyacentes deberán tener los mismos colores en los bordes que tienen en común

Algoritmos implementados

Algoritmos no informados:

- DFS
- BFS
- Iterative Deepening

Algoritmos iformados:

- A*
- Greedy

Cristian Pereyra - Esteban Pintos - Matías De Santi

Heurísticas
Heurística basada en distancias Manhattan

- Es más fácil resolver el juego empezando desde los bordes

4	3	2	3	4
3	2	1	2	3
2	1	0	1	2
3	2	1	2	3
4	3	2	3	4

Distancias mahnattan para un tablero de 5x5

4	3		3	4
			1	2
	2	1		
	3			4

Distancias mahnattan para un tablero de 5x5 con algunas piezas colocadas

- $h(n) = 60 - 27 = 33$

Cristian Pereyra - Esteban Pintos - Matías De Santi

Heurísticas
Heurística basada en el orden de colocación

Intenta que las piezas sean colocadas en orden de espiral. Primero en (0,0), luego en (0,1) y así hasta completar la primer fila. Luego desciende por la última columna y así sucesivamente

1	2	3	4	5
16	17	18	19	6
15	24	25	20	7
14	23	22	21	8
13	12	11	10	9

Recorrido en espiral para un tablero de 5x5

Cristian Pereyra - Esteban Pintos - Matías De Santi

Heurísticas
Heurística basada en colores

- Intenta evaluar si quedan colores para continuar con un tablero. En caso de que no haya, los sucesivos tableros generados a partir del actual van a ser inválidos.

Cristian Pereyra - Esteban Pintos - Matías De Santi

Funciones de costo

Se desarrollaron 3 funciones de costo:

- Costo uniforme
- Costo basado en rotaciones
- Costo basado en distancia al centro

Cristian Pereyra - Esteban Pintos - Matías De Santi

Factor de ramificación

- Reglas generadas: $4 \times n^4$
- Un nodo del árbol ubicado a una profundidad *p* tiene un factor de ramificación $4 \times n^4 - 4p$
- Unos ejemplos…

2x2 => 11182080

3x3 => 2481742868091450163200 0

4x4 => 905703814554358516996854857467489926355353600000

Cristian Pereyra - Esteban Pintos - Matías De Santi

Resultados

Conclusiones

- La representación del problema es importante a la hora de ahorrar memoria.
- Conocer el factor de ramificación es importante para saber con qué tipo de problemas uno esta lidiando.
- Los algoritmos no informados no dan buenos resultados ya que exploran una gran cantidad de nodos.