



UNIVERSIDAD DE
GUADALAJARA

Red Universitaria e Institución Benemérita de Jalisco

CUCEI

CENTRO UNIVERSITARIO DE
CIENCIAS EXACTAS E INGENIERÍAS

Inteligencia Artificial

1

Practica 2 Puzzle búsqueda en profundidad y profundidad iterativa

Veloz Alcaraz Axel Abraham

Espinoza Sucilla Samuel

Cornejo Chavez Edwin Joel | 220791268 | 27-10-2023

LUIS ANGEL BELTRAN CARRILLO

DIEGO ALBERTO OLIVA NAVARRO



Objetivo:

Implementar los algoritmos de búsqueda no informada en problemas de prueba para poder comparar su desempeño.

Problema:

Implementar la búsqueda en amplitud y en profundidad para dar solución de forma automática al problema 8-puzzle (Fig. 1).

El estado inicial debe ser aleatorio.

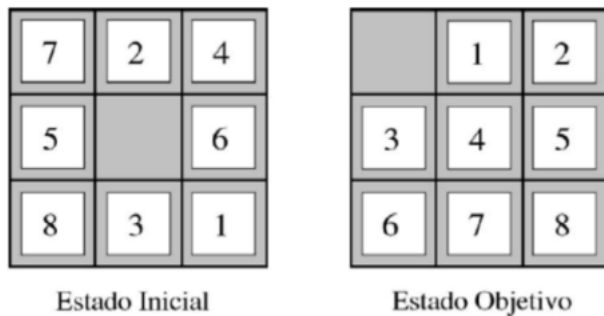


Fig.1 Ejemplo del problema 8 Puzzle

Introducción:

La búsqueda en profundidad es un algoritmo de búsqueda no informada utilizado en inteligencia artificial y ciencias de la computación para explorar un espacio de estados en un grafo o árbol. A diferencia de la búsqueda por amplitud, que explora todos los vecinos de un nodo antes de moverse a los nodos vecinos de los vecinos, la búsqueda en profundidad explora tan profundamente como sea posible antes de retroceder.

La búsqueda en profundidad iterativa es una variante de la búsqueda en profundidad que aborda la limitación de no garantizar la solución óptima. Esta técnica combina la profundidad de la búsqueda en profundidad con la amplitud de la búsqueda en amplitud al realizar una búsqueda iterativa con diferentes límites de profundidad.

Desarrollo

Esta es la función que se encarga de resolver el 8-Puzzle usando la búsqueda en profundidad.

```
def dfs(estado_actual, estado_objetivo, profundidad_maxima):  
    if profundidad_maxima < 0:  
        return None  
    if son_iguales(estado_actual, estado_objetivo):  
        return [estado_actual]
```

```

    for direccion in ['arriba', 'abajo', 'izquierda', 'derecha']:
        nuevo_estado = mover_ficha(estado_actual, direccion)
        resultado = dfs(nuevo_estado, estado_objetivo,
profundidad_maxima - 1)
        if resultado:
            return [estado_actual] + resultado

```

Solución

El programa debe mostrar la solución tal y como la anterior

```

TERMINAL  PROBLEMS  OUTPUT  DEBUG CONSOLE  COMMENTS  PORTS

PS D:\Documents\CUCEI\Semestre 7\Inteligencia Artificial 1\Practicas> & C:/Users/
hon.exe "d:/Documents/CUCEI/Semestre 7/Inteligencia Artificial 1/Practicas/Pu
Estado inicial:
[4, 1, 0]
[2, 5, 3]
[8, 6, 7]

Solución encontrada:
[1, 2, 3]
[4, 5, 6]
[7, 8, 0]
PS D:\Documents\CUCEI\Semestre 7\Inteligencia Artificial 1\Practicas>

```

Pero debido a la naturaleza del algoritmo y la manera en la que se implementa, el algoritmo puede tardar ya sea de 5 minutos hasta incluso un año en encontrar la solución, para resolver esto hay solución pero esto ya lleva otros conceptos más avanzados así que por el momento nos quedaremos con esto.

```

PS D:\Documents\CUCEI\Semestre 7\Inteligencia Artificial 1\Practicas> & C:/Users/
ments\CUCEI\Semestre 7\Inteligencia Artificial 1\Practicas/PuzzleProfundidad.py"
Estado inicial:
[5, 2, 7]
[6, 0, 4]
[1, 8, 3]

```

Conclusión

En conclusión, tanto la búsqueda en profundidad como la búsqueda en profundidad iterativa son algoritmos de búsqueda valiosos en el campo de la inteligencia artificial y las ciencias de la computación, y cada uno tiene sus propias características y aplicaciones.