



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 amplitud

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.

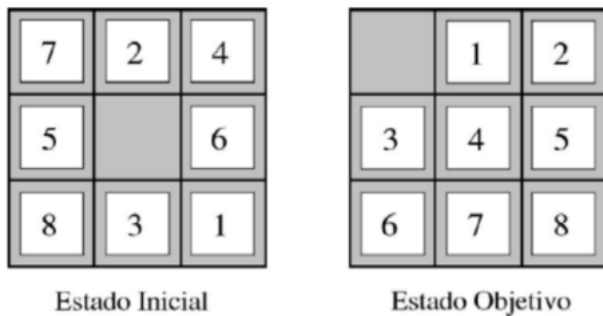


Fig.1 Ejemplo del problema 8 Puzzle

Introducción:

La búsqueda no informada, también conocida como búsqueda ciega o búsqueda a ciegas, es un enfoque utilizado en la inteligencia artificial y la informática para encontrar soluciones a problemas sin utilizar información adicional sobre el problema más allá de las reglas y restricciones proporcionadas. Este tipo de búsqueda se utiliza comúnmente en algoritmos de búsqueda en grafos y es útil para resolver problemas en los que no se dispone de información heurística o específica sobre el estado inicial y el estado objetivo.

Búsqueda en amplitud (Breadth-First Search - BFS): En este algoritmo, se comienza desde el estado inicial y se explora cada posible movimiento en todas las direcciones antes de avanzar a estados más profundos en el árbol de búsqueda. El BFS garantiza encontrar la solución más corta en términos de número de pasos, pero puede ser computacionalmente costoso si el espacio de búsqueda es grande.

Desarrollo

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

```
def resolver_puzzle(estado_inicial, estado_objetivo):  
    frontera = deque()  
    visitados = set()
```

```

frontera.append(estado_inicial)
visitados.add(tuple(estado_inicial))

while frontera:
    estado_actual = frontera.popleft()

    if son_iguales(estado_actual, estado_objetivo):
        return estado_actual

    for direccion in ['arriba', 'abajo', 'izquierda', 'derecha']:
        nuevo_estado = mover_ficha(estado_actual, direccion)
        if tuple(nuevo_estado) not in visitados:
            frontera.append(nuevo_estado)
            visitados.add(tuple(nuevo_estado))

```

## Solución

```

TERMINAL  PROBLEMS  OUTPUT  DEBUG CONSOLE  COMMENTS  PORTS

PS D:\Documents\CUCEI\Semestre 7\Inteligencia Artificial 1\Practicas> & C:/Us
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>

```

## Conclusión

La búsqueda por amplitud es un enfoque fundamental en la resolución de problemas en inteligencia artificial, Aplicar el algoritmo de búsqueda en amplitud para encontrar una solución al 8-puzzle, y si se encuentra una solución, la muestra en la consola. Esto demuestra cómo la búsqueda en amplitud se puede aplicar de manera efectiva para resolver problemas reales en la inteligencia artificial.