

Estrategias de Búsqueda en Inteligencia Artificial

Este trabajo práctico explora los conceptos fundamentales de búsqueda en inteligencia artificial, comparando estrategias informadas y no informadas, y analizando su implementación en problemas concretos.



Búsqueda Informada vs. No Informada

Búsqueda No Informada

El agente no tiene información adicional sobre los estados más allá de la definición del problema. Solo sabe si un estado es objetivo o no, y va explorando el espacio sin guía.

Ejemplos: búsqueda en amplitud, búsqueda en profundidad, búsqueda de costo uniforme.

Búsqueda Informada

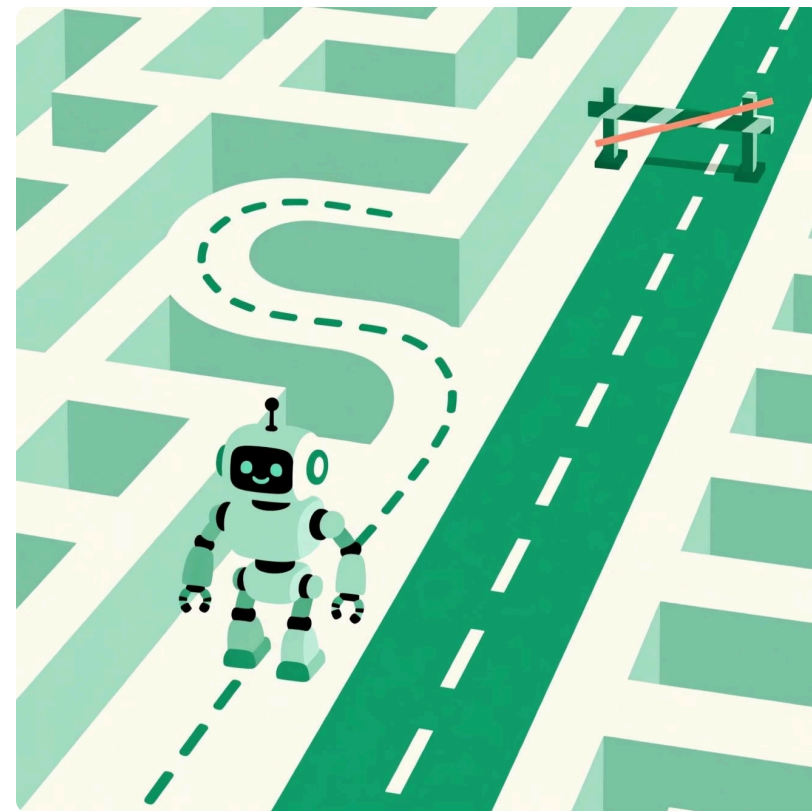
Utiliza conocimiento específico del problema más allá de la definición. El agente usa información adicional que lo orienta sobre qué tan cerca está un estado de la solución.

Ejemplos: búsqueda voraz primero el mejor, A*.

Heurísticas: Guías para la Búsqueda

La función heurística representa el costo estimado del camino más barato desde el nodo actual hasta un nodo objetivo. Sirve como una "pista" para guiar la búsqueda.

Aunque no garantiza que sea la ruta real más corta, ayuda a priorizar los caminos que parecen más prometedores, reduciendo el espacio de búsqueda y mejorando la eficiencia.



Un ejemplo: si un robot debe ir desde un punto A a un punto B, la heurística podría ser la distancia en línea recta entre ambos puntos.

Limitaciones de los Algoritmos de Búsqueda

Un algoritmo de búsqueda puede no encontrar solución por tres razones principales:

1

Problema sin solución

El problema no tiene estados objetivo alcanzables desde el estado inicial.

2

Algoritmo incompleto

Como en búsqueda en profundidad con bucles infinitos, que puede quedar atrapado en un camino sin fin.

3

Limitaciones prácticas

Restricciones de tiempo y memoria que impiden explorar todo el espacio de estados.

Recorrido del Árbol: Primero en Amplitud

Estado actual							
A	D	F	I				
D	F	I	H	J			
F	I	H	J	C	E		
I	H	J	C	E			
H	J	C	E	P	O		
J	C	E	P	O	K		
C	E	P	O	K	Z	W	
E	P	O	K	Z	W		
P	O	K	Z	W			
O	K	Z	W				
K	Z	W	B				
Z	W	B					
W	B						
B							

La búsqueda en amplitud explora todos los nodos de un nivel antes de pasar al siguiente nivel del árbol.

Recorrido del Árbol: Primero en Profundidad

Estado actual					
A	D	F	I		
D	H	J	F	I	
H	P	O	J	F	I
P	O	J	F	I	
O	J	F	I		
J	K	B	F	I	
K	B	F	I		
B	F	I			
F	C	E	I		
C	Z	W	E	I	
Z	W	E	I		
W	E	I			
E	I				
I					

La búsqueda en profundidad explora un camino hasta el final antes de retroceder y explorar otras alternativas.

Recorrido con Profundidad Limitada Iterativa

La búsqueda en profundidad con profundidad limitada iterativa combina las ventajas de la búsqueda en amplitud y en profundidad:

- Realiza múltiples pasadas de búsqueda en profundidad
- Incrementa gradualmente el límite de profundidad
- Garantiza encontrar la solución óptima con memoria limitada
- En el ejemplo, comienza con profundidad 1, luego 2, etc.

Esta estrategia es especialmente útil cuando el espacio de búsqueda es grande y la profundidad de la solución es desconocida.

Estado actual				
Límite 1:				
A				
D				
F				
I				
Límite 2:				
A				
D				
H				
J				
F				
C				
E				
I				

D	F	I		
F	I			
I				
D	F	I		
H	J	F	I	
J	F	I		
F	I			
C	E	I		
E	I			
I				

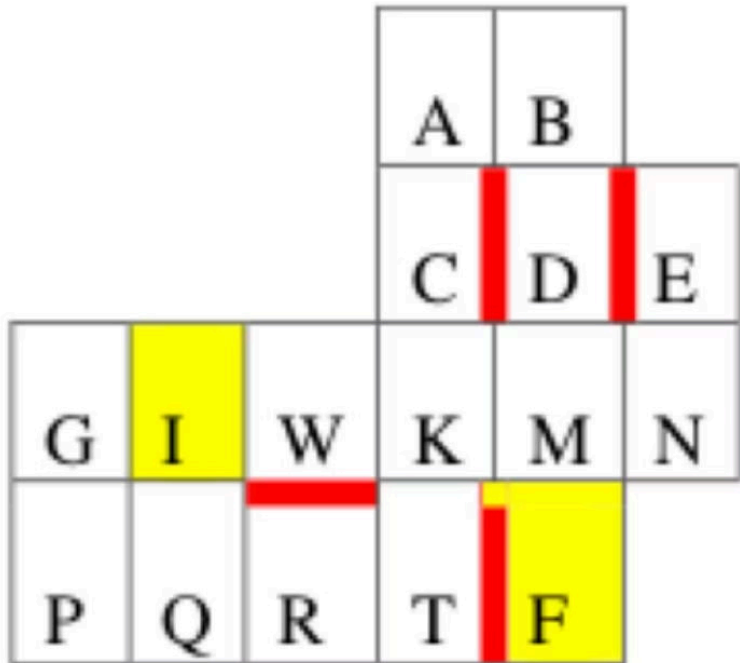
Límite 3:				
A				
D				
H				
P				
O				
J				
K				
F				
C				
Z				
W				
E				
I				

D	F	I		
H	J	F	I	
P	O	J	F	I
O	J	F	I	
J	F	I		
K	F	I		
F	I			
C	E	I		
Z	W	E	I	
W	E	I		
E	I			
I				

Límite 4:				
A				
D				
H				
P				
O				
J				
K				
B				
F				
C				
Z				
W				
E				
I				

D	F	I		
H	J	F	I	
P	O	J	F	I
O	J	F	I	
J	F	I		
K	B	F	I	
B	F	I		
F	I			
C	E	I		
Z	W	E	I	
W	E	I		
E	I			
I				

Implementación: Navegación en Tablero



Características del problema:

- El agente debe navegar desde la casilla I hasta F
- No puede atravesar paredes
- Heurística: Distancia de Manhattan
- Costo de atravesar casillas: 1 (excepto W: 30)
- En caso de empate: elección por orden alfabético

Comparación de Estrategias



Primero en Profundidad

Explora rápidamente caminos completos pero no garantiza la solución óptima. Puede quedar atrapado en caminos muy largos.



Búsqueda Voraz

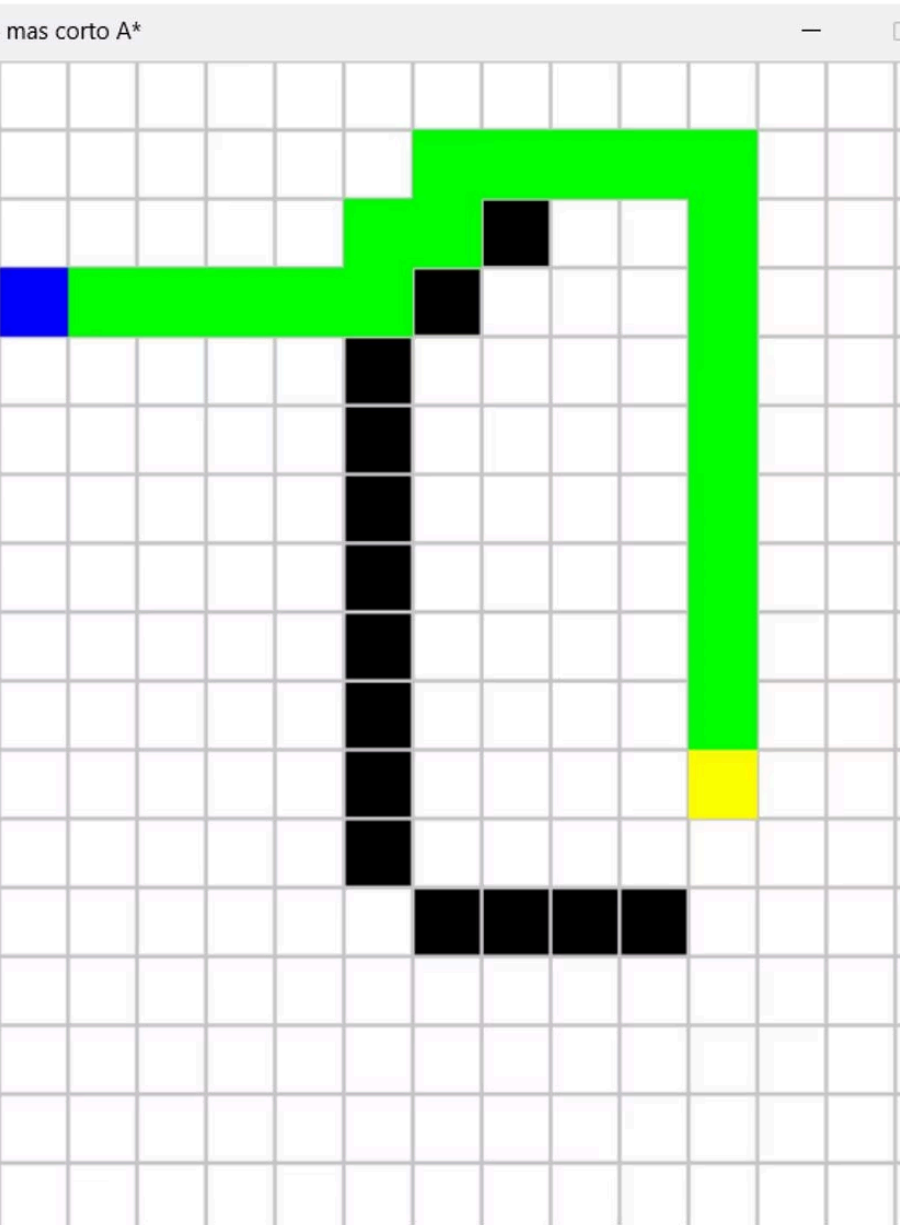
Siempre elige el nodo que parece más cercano al objetivo según la heurística. Puede encontrar soluciones rápidamente pero no siempre óptimas.



A*

Combina el costo real recorrido con la heurística. Garantiza la solución óptima si la heurística es admisible.

La elección de la estrategia depende del problema específico, los recursos disponibles y si se requiere la solución óptima o solo una solución aceptable.



Implementación de A* en Entorno Visual

La implementación visual de A* muestra:

- Casillas de inicio (azul) y destino (amarillo)
- Obstáculos (negro) que el agente debe evitar
- El camino óptimo encontrado (verde)
- La heurística de Manhattan guía la búsqueda hacia el objetivo

Esta visualización demuestra la eficacia de A* para encontrar el camino más corto en entornos con obstáculos.