

Tema-3.pdf



roro_pocha



Inteligencia Artificial



2º Grado en Ingeniería Informática



Escuela Técnica Superior de Ingenierías Informática y de Telecomunicación
Universidad de Granada



Estamos de
Aniversario

De la universidad al
mercado laboral:
especialízate con los posgrados
de EOI y marca la diferencia.



EOI Escuela de
organización
industrial



saber más

Importante

Puedo eliminar la publi de este documento con 1 coin

¿Cómo consigo coins? → Plan Turbo: barato
→ Planes pro: más coins

pierdo espacio



Necesito concentración

ali ali ooh
esto con 1 coin me
lo quito yo...

WUOLAH

Tema 3:

Diseño de un agente deliberativo: búsqueda

Vamos a suponer que:

- El agente dispone un modelo del mundo en el que habita, pero esto no significa que conozca todo, porque un robot que simplemente no se tiene que chocar con las paredes no necesita reconocer los colores.

La búsqueda en un espacio de estados

Un estado es como se observa el mundo en un momento determinado.

A partir de un estado, puedo proyectar el estado obtenido tras realizar una serie de acciones. Esto sería el espacio de estados.

Lo que queremos es que el robot explore el espacio de estados para que determine por sí mismo la solución.

Descripción de un problema

- Estados
- Estado inicial
- Acciones
- Objetivo
- Costo de las acciones

Sistemas de búsqueda y estrategias

Búsqueda en grafos/árboles

Uso de grafos explícitos para encontrar la solución del problema, diseñas tú el grafo con todas las posibles acciones y estados. Tras el diseño del árbol, se diseñan algoritmos que exploren los grafos. Este método solo se puede implementar en problemas sencillos.

Establecemos un costo de las acciones para intentar obtener el estado objetivo en el menor número de pasos posible.

Estrategias de control

Estrategias irrevocables

Al aplicar una acción no se puede volver al estado anterior.

Estrategias tentativas

Retroactivas (Backtracking)

Hago un modelo tentativo con el mínimo gasto de memoria. Exploro solo el primer camino, cuando no puedo seguir y no he hallado la solución, vuelvo al estado anterior y sigo probando otros caminos.

Puede que tengamos caminos infinitos que no lleven a ninguna solución, por lo que podemos ponerle un límite de acciones para que vuelva hacia atrás y pruebe otros caminos.

Búsqueda en grafos

Tienen máxima flexibilidad, en vez de dar un paso hacia atrás puedo volver a otra opción que había mucho antes sin más, como al principio del grafo.

Infraestructura para los algoritmos de búsqueda

El nodo no solo contiene el estado, si no también quién es su padre, que acción ha generado el estado y que costo tiene.

WUOLAH

Medidas del comportamiento de un sistema de búsqueda

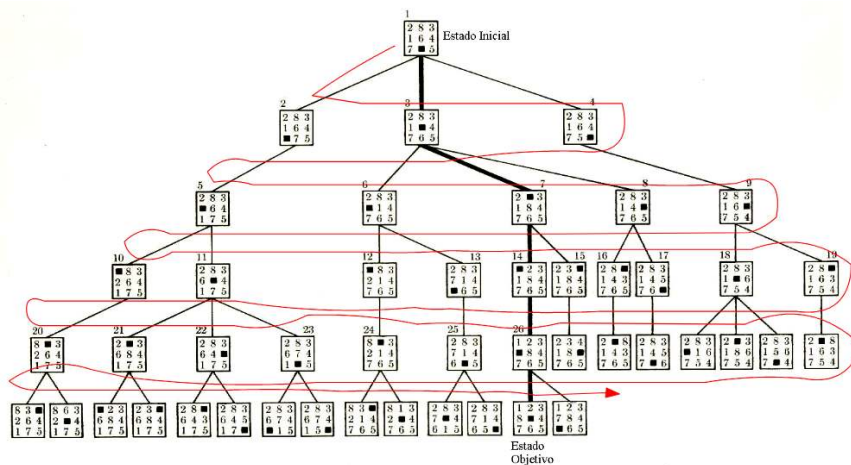
- **Complejitud:** Encontrar soluciones completas
- **Optimalidad:** Encontrar la solución óptima.
- **Complejidad en tiempo:** Saber cuanto tarda.
- **Complejidad en el espacio:** Cuanta memoria necesita

Búsqueda sin información

Algoritmos que no proporciona la IA, se heredan de otros algoritmos.

Búsqueda en anchura / Búsqueda con costo

Comprobar constantemente si hay algún nodo solución en la fila, es decir, que no tenga más hijos.



Características:

- **Completo:** encuentra la solución si existe
- **Optimalidad:** si todas las acciones tienen el mismo coste, encontrará la solución óptima
- **Eficiencia:** buena si las metas están cercanas.
- **Problema:** consume memoria exponencial

Búsqueda en profundidad (sobre grafos /retroactiva)

Empieza a bajar hasta que el camino ya no tiene más hijos. Si este nodo no es la solución, o se produce un fallo, retrocede y vuelve a probar por otros hijos del nodo anterior.

Características:

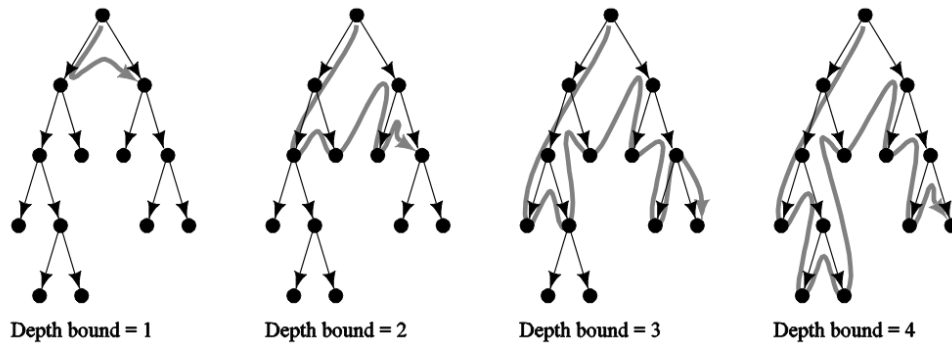
- **Complejitud:** no asegura encontrar la solución. Puede que empiece a bajar infinitamente sin encontrar una solución.
- **Optimalidad:** no asegura encontrar la solución más óptima.
- **Eficiencia:** Bueno cuando las metas están alejadas del estado inicial, o hay problemas de memoria
- No es bueno cuando hay ciclos.

Descenso Iterativo

El problema del anterior algoritmo era que podía llegar el caso en el estuviera bajando infinitamente, podría implementar un límite de profundidad, pero lo mismo pongo ese límite

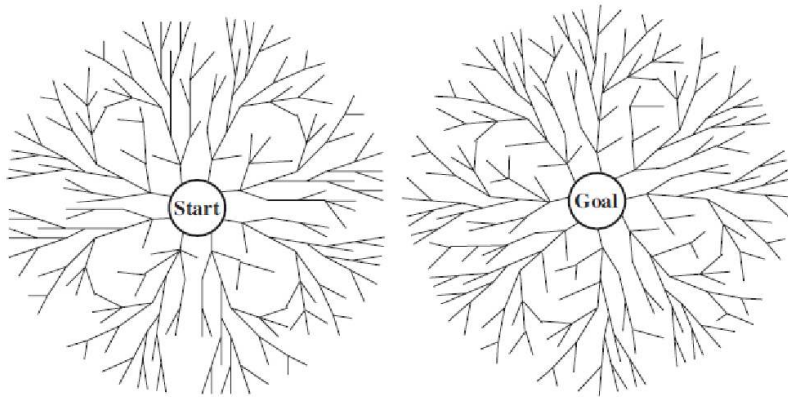
demasiado arriba y no encuentro la solución. Por lo que este algoritmo soluciona ese problema.

Consiste en ir explorando los nodos por profundidad, primero los nodos de profundidad 1, luego los de profundidad 2, etc.



Con este algoritmo conseguiremos encontrar la solución más óptima.

Búsqueda Bidireccional



Este algoritmo se basa en una búsqueda desde 2 puntos, el nodo inicio y el nodo objetivo. Estos dos van avanzando hasta que se encuentran los dos caminos.

Gracias a esto exploramos menos nodos, ya que cada camino tiene que recorrer la mitad del camino.

Búsqueda con información

Heurísticas

Se tiene conocimiento parcial sobre un problema/dominio. Las heurísticas son criterios para decidir qué acción promete ser la mejor. Este método no garantiza la solución óptima pero en media produce resultados satisfactorios.

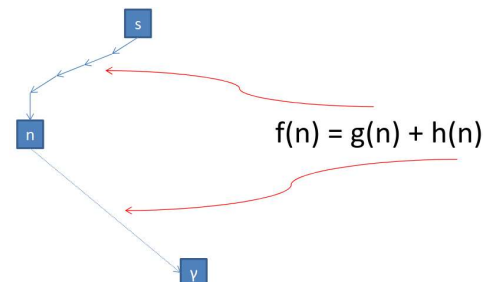
Heurística para el algoritmo A*

Se guía por el coste de los movimientos.

g(n): costo acumulado desde el nodo inicial hasta el nodo n.

h(n): función heurística que estima el costo desde n al objetivo.

f(n): la función de evaluación total para decidir qué nodo expandir.



Importante

Puedo eliminar la publi de este documento con 1 coin

¿Cómo consigo coins? → Plan Turbo: barato
→ Planes pro: más coins

pierdo espacio



Necesito concentración

ali ali oohh
esto con 1 coin me
lo quito yo...

WUOLAH

Búsqueda en grafos donde abiertos es una cola con prioridad ordenada de acuerdo a $f(n)$.

- ABIERTOS contiene el nodo inicial, CERRADOS está vacío.
- Comienza un ciclo que se repite hasta que se encuentra solución o hasta que ABIERTOS queda vacío.
 - Seleccionar el mejor nodo de ABIERTOS
 - Si es un nodo objetivo terminar
 - En otro caso se expande dicho nodo y se calculan los valores $f(n)$ y $g(n)$ de sus vecinos.
 - + Si un vecino ya está en la CERRADOS con un costo menor, se ignora.
 - + Si no, se agrega a ABIERTOS con su costo actualizado.*

Características:

- **Complejidad:** si existe solución, la encuentra
- **Admisibilidad:** si hay una solución óptima, y la función $h(n)$ es admisible, la encuentra:

Función $h(n)$ es admisible: $h(n) \leq h^*(n)$.

Esto significa que para todos los nodos del grafo, la $h(n)$ del nodo inicial debe ser menor o igual que la $h(n)$ de todos los nodos del grafo.

Búsqueda Dirigida (Beam Search)

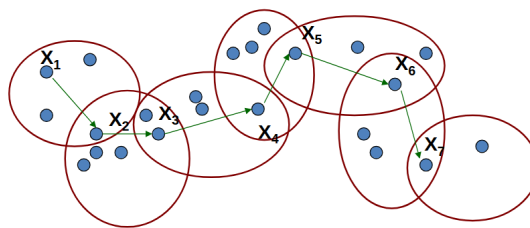
Es una extensión de algoritmo A^* .

Cada vez que se expande un nodo, se generan sus sucesores, se evalúa $f(n)$ de cada uno y se eliminan los que tengan peor $f(n)$.

Si $h(n)$ es suficientemente buena te puede asegurar encontrar la solución óptima.

Métodos de escalada

Consiste en seleccionar la mejor solución e ir viajando entre ellas hasta encontrar la solución óptima.

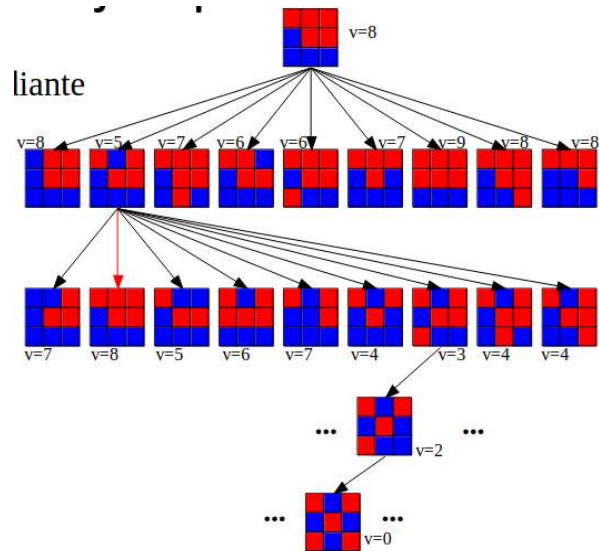


Características:

- **Complejidad:** no tiene porque encontrar la solución.
- **Admisibilidad:** no siendo completo, aún menos será admisible.
- **Eficiencia:** rápido y útil si la función es monótona (de)creciente.

WUOLAH

En el ejemplo de la derecha:
 Generamos distintos movimientos posibles.
 Elegimos el nodo que tenga el mejor valor. A
 partir de él seguimos generando los
 movimientos posibles. Y así continuamente.



Algunas variaciones estocásticas (Con elementos aleatorios)

Algoritmo de escalada estocástico: Selecciona un sucesor aleatoriamente entre los que mejoran la solución. No elige siempre el mejor, sino uno “bueno” al azar. Evita que el algoritmo siga siempre el mismo camino y mejora la exploración del espacio.

Algoritmo de escalada de primera opción: En lugar de explorar todos los vecinos, recorre uno por uno y se queda con el primero que mejora la solución actual. Es más rápido, aunque no garantiza la mejor mejora. Puede ser más útil en espacios grandes.




























Algoritmo de escalada de reinicio aleatorio: Ejecuta el algoritmo de escalada varias veces, cada vez desde un estado inicial aleatorio. Se queda con la mejor solución encontrada al final. Es útil para evitar quedar atrapado en máximos locales.

Enfriamiento simulado: Acepta, de forma probabilística, soluciones peores al principio, para permitir salir de máximos locales. A medida que avanza el tiempo, se vuelve más exigente y acepta menos soluciones peores. Muy eficaz para encontrar óptimos globales en espacios complejos.

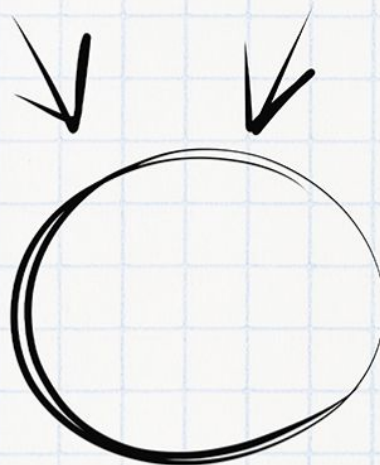
Estas variaciones estocásticas son formas inteligentes de evitar los problemas de quedarse atrapado en soluciones subóptimas, permitiendo explorar el espacio de estados de forma más diversa y efectiva.

Imagínate aprobando el examen

Necesitas tiempo y concentración

Planes	 PLAN TURBO	 PLAN PRO	 PLAN PRO+
 Descargas sin publi al mes	10 	40 	80 
 Elimina el video entre descargas			
 Descarga carpetas			
 Descarga archivos grandes			
 Visualiza apuntes online sin publi			
 Elimina toda la publi web			
 Precios Anual <input type="checkbox"/>	0,99 € / mes	3,99 € / mes	7,99 € / mes

Ahora que puedes conseguirlo,
¿Qué nota vas a sacar?



WUOLAH

Algoritmos genéticos

- **Cromosoma.** Vector representación de una solución al problema
- **Gen.** Característica/Variable/Atributo concreto del vector de representación de una solución
- **Población.** Conjunto de soluciones al problema.
- **Adecuación al entorno.** Valor de función objetivo (fitness).
- **Selección natural.** Operador de selección.
- **Reproducción sexual.** Operador de cruce.
- **Mutación.** Operador de mutación.
- **Cambio generacional.** Operador de reemplazamiento.

Procedimiento:

1. **Generar y evaluar población inicial:** Cada solución posible se representa como un cromosoma. Cada cromosoma es un individuo de la población. Se calcula la función de aptitud (fitness) para cada individuo, que mide qué tan buena es la solución.
2. **Selección de individuos padres:** Se eligen los mejores individuos para reproducirse.
3. **Cruzamiento de padres:** Se combinan partes de dos padres para crear descendientes.
4. **Mutación de hijos/padres:** Se cambia aleatoriamente una parte del cromosoma, para introducir diversidad. Previene que todos converjan a la misma solución local.
5. **Evaluación de nuevos individuos:** Se eligen los mejores individuos para el reemplazo
6. **Reemplazo:** Se forma una nueva generación de soluciones. Se repite el ciclo por muchas generaciones.

Este método es muy útil cuando el espacio de búsqueda es grande, complejo o desconocido.