

Esto no son apuntes pero **tiene un 10 asegurado** (y lo vas a disfrutar igual).

Abre la Cuenta NoCuenta con el código **WUOLAH10**, haz tu primer pago y llévate 10 €.

Me interesa



1/6

Este número es indicativo del riesgo del producto, siendo 1/6 indicativo de menor riesgo y 6/6 de mayor riesgo.

ING BANK NV se encuentra adherido al Sistema de Garantía de Depósitos Holandés con una garantía de hasta 100.000 euros por depositante. Consulta más información en [ing.es](https://www.ing.es)

Tema 3: Búsqueda en espacios de estados

1. Diseño de un agente deliberativo: búsqueda

El **agente deliberativo** dispone tanto de un modelo del mundo en el que habita tanto de un modelo de los efectos que sus acciones tienen sobre el mundo, además es capaz de razonar sobre esos modelos para decidir que hacer para conseguir un objetivo.

1.1. Componentes de la descripción de un problema

- **Estado inicial:** estado en el que se encuentra el agente al inicio.
- **Estado objetivo:** estado en el que tiene que estar el agente al final.
- **Acciones:** repertorio de acciones del agente.
- **Efecto de las acciones o modelo de transición:** estado resultante de aplicar una acción en un estado dado, para cada estado podemos representar las acciones que se pueden ejecutar y su **Efecto**.
- **Espacio de estados:** es el conjunto de estados alcanzables desde el estado inicial mediante una secuencia de acciones.
- **Comprobación de objetivo:** función para comprobar si el estado dado es objetivo.
- **Costo de las acciones:** una función que asigna un coste numérico a cada camino.

Una **solución** a un problema es una secuencia de acciones que empieza en el estado inicial y permite alcanzar el estado objetivo.

Una **solución óptima** consiste en el camino con el coste más bajo de entre todas las posibles soluciones.

Un **grafo de estados** es una estructura donde un nodo representa un estado del sistema y un arco una posible acción.

A la secuencia de acciones que llevan al agente desde el inicio al destino se denomina **plan** y la búsqueda del plan se denomina **planificación**.

- **Grafos explícitos:** grafo que representa el espacio de estados explorados hasta el momento y se almacena físicamente en

Consulta condiciones aquí



do your thing

WUOLAH

memoria. Este grafo se va construyendo conforme aplicamos acciones, sabemos que se parte del estado inicial, pero puede ser necesario partir del estado destino y aplicar acciones en sentido inverso, haciendo lo que se llamaría un **razonamiento regresivo**", un proceso de exploración del grafo **regresivo**.

- **Grafos implícitos:** grafo que representa el espacio de estados en su totalidad y no está almacenado físicamente.

2. Sistemas de búsqueda y estrategias

Una vez formulado el problema hay que resolverlo, con un proceso de búsqueda encontramos un plan siendo el estado inicial el nodo raíz del grafo/árbol de búsqueda, los arcos son las acciones y los nodos estados del espacio de estados del problema.

2.1. Estrategias de control

Dependiendo de cómo representemos el grafo explícito podemos usar distintas estrategias para explorarlo.

2.1.1. Estrategias irrevocables

En cada momento, el grafo explícito lo constituye un único nodo E, que incluye la descripción completa del sistema en ese momento:

- Se selecciona una acción A.
- Se aplica sobre el estado E para obtener el nuevo estado $E' = A(E)$.
- Se borra de memoria E y se sustituye por E'.

Este proceso se detiene cuando E = estado objetivo.

2.1.2. Estrategias tentativas: Retroactivas (Backtracking)

En memoria sólo guardamos un hijo de cada estado y se mantiene el camino desde el estado inicial hasta el actual. El grafo explícito es una lista. El proceso para cuando hemos llegado al objetivo y no deseamos encontrar más soluciones o no hay más operadores aplicables al nodo raíz.

Se produce una vuelta atrás cuando:

- Se ha encontrado una solución, pero queremos una solución alternativa.
- Cuando se ha llegado a un límite en el nivel de profundidad explora o el tiempo de exploración en una misma rama.
- Cuando se ha generado un estado que ya existía en el camino.
- Cuando no existen reglas aplicables al último nodo del grafo.

2.1.3. Estrategias tentativas: Búsqueda en grafos/árboles

En memoria se guardan todos los estados generados hasta el momento, de forma que la búsqueda pueda proseguir por cualquiera de ellos.

- En **búsqueda en árboles** se usa una lista denominada **Frontera (o Abiertos)** donde se almacenan todos los nodos generados y que están pendientes de explorar.
- En **búsqueda en grafos** se usa una lista adicional denominada **Explorados (o Cerrados, o Visitados)**, donde se almacenan los nodos que estaban en Abiertos y ya se han explorado.

2.2. Medidas del comportamiento de un sistema de búsqueda

- **Complejidad:** hay garantía de encontrar la solución si esa existe.
- **Optimalidad:** hay garantía de encontrar la solución óptima.
- **Complejidad en tiempo:** ¿Cuánto se tarda en encontrar la solución?
- **Complejidad en espacio:** ¿Cuánta memoria se necesita para realizar la búsqueda?

Esto no son apuntes pero **tiene un 10 asegurado** (y lo vas a disfrutar igual).

Abre la Cuenta NoCuenta con el código **WUOLAH10**, haz tu primer pago y llévate 10 €.

Me interesa



1/6

Este número es indicativo del riesgo del producto, siendo 1/6 indicativo de menor riesgo y 6/6 de mayor riesgo.

ING BANK NV se encuentra adherido al Sistema de Garantía de Depósitos Holandés con una garantía de hasta 100.000 euros por depositante. Consulta más información en [ing.es](https://www.ing.es)

3. Búsqueda sin información

El siguiente nodo a explorar una vez generados los sucesores, se escoge de forma ciega siguiendo la misma táctica.

3.1. Búsqueda en anchura

Es una técnica de búsqueda en grafos, en la que se usan dos listas (frontera y explorados). Al inicio tenemos la lista de cerrados vacía y la de abiertos con un único nodo (estado inicial), entonces la búsqueda en anchura trata de seleccionar el nodo más superficial de la lista de abiertos, se inserta en cerrados y se expanden los nodos, si no están en explorados ni abiertos, se introducen en abiertos en caso de que este no sea el objetivo. **(Esto más que nada es para saber cómo funciona, sólo entenderlo).**

Características:

- **Completo:** encuentra la solución si existe y el factor de ramificación es finito en cada nodo.
- **Optimalidad:** si todos los operadores tienen el mismo coste, encontrara la solución óptima.
- **Eficiencia:** buena si las metas están cercanas.
- **Problema:** consume memoria exponencial.

3.2. Búsqueda en profundidad

Saber que hay dos versiones, y una de ellas es igual que la búsqueda en anchura, pero cambiando en FIFO por LIFO.

3.2.1. Búsqueda en profundidad retroactiva

Usamos una técnica basada en la recursividad. Como hemos visto en backtracking, lo que hace es mirar su primer hijo, si es solución sigue si es posible y si no vuelve atrás, al padre para que mire en otro hijo.

Características:

- **Complejidad:** no asegura encontrar la solución.
- **Optimalidad:** no asegura encontrar la solución óptima.
- **Eficiencia:** bueno cuando las metas están alejadas del estado inicial, o hay problemas de memoria.
- No es bueno cuando hay ciclos.

Consulta condiciones aquí



do your thing

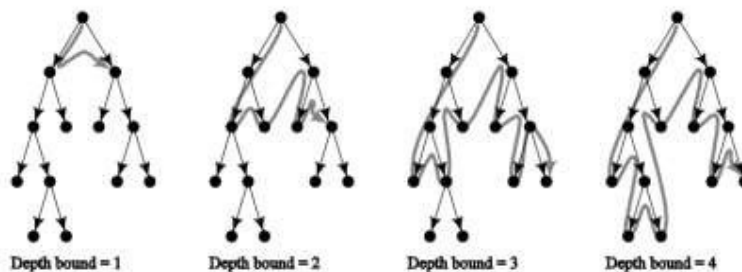
WUOLAH

3.3. Búsqueda con costo

Es como la búsqueda en anchura, pero el nodo guarda información sobre el coste del camino desde el nodo inicial hasta ese nodo. La diferencia está en cómo se organiza la lista de abiertos, que en este caso es una cola con prioridad que se ordena en función del coste del camino que se almacena en cada nodo.

3.4. Descenso iterativo

Es como una búsqueda en profundidad, pero en cada iteración incrementa el límite de profundidad, por tanto lo que hace es la búsqueda en profundidad, pero aumentando 1 el límite por cada iteración.



En cada iteración se revisan todos los nodos revisados anteriormente y no es costos porque no almacena los nodos visitados.

3.5. Búsqueda bidireccional

Integra búsqueda desde el estado inicial o búsqueda desde el estado final, con la esperanza de poderse encontrar en el medio. Se denomina búsqueda progresiva o búsqueda regresiva y tienen estados sucesores o predecesores.

4. Búsqueda con información

4.1. Heurísticas

- Si se tiene conocimiento perfecto: algoritmo exacto
- Si no se tiene conocimiento: búsqueda sin información.
- En la mayor parte de los problemas estamos en posiciones intermedias.
- **Heurística:** Conocimiento parcial sobre un problema/dominio que permite resolver problemas eficientemente en ese problema/dominio.
- Las **heurísticas** son criterios, métodos o principios para decidir cuál de entre varias acciones es la mejor para alcanzar una meta. Entendemos por heurística un método para resolver problemas que no garantiza la solución óptima, pero en media produce resultados satisfactorios en la resolución de un problema.
- Una heurística sirve de guía para que un algoritmo de búsqueda pueda encontrar una solución válida aceptable y eventualmente puede devolver siempre soluciones óptimas bajo ciertas condiciones (requiere demostración).

4.2. Métodos de escalada

4.2.1. Algoritmo de escalada simple

A partir de un estado activo, compara su heurística con la del estado vecino y si la del vecino es mejor, se sustituye el estado actual por el vecino. Si este nodo nuevo es el objetivo se detiene y si no es el objetivo, continua. El problema básico es que se puede quedar atascado.

4.2.2. Algoritmo de escalada por la máxima pendiente

Para solucionar el problema anterior, contempla todos los vecinos que se puedan generar a partir del estado actual y escoge el mejor de ellos evaluando la función heurística. Se puede seguir quedando atascado.

Características:

- **Complejidad:** No tiene por qué encontrar la solución.
- **Optimalidad:** No siendo completo, aún menos será óptimo.
- **Eficiencia:** Rápido y útil si la función es monótona (de)creciente.

Esto no son apuntes pero **tiene un 10 asegurado** (y lo vas a disfrutar igual).

Abre la Cuenta NoCuenta con el código **WUOLAH10**, haz tu primer pago y llévate 10 €.

Me interesa



1/6

Este número es indicativo del riesgo del producto, siendo 1/6 indicativo de menor riesgo y 6/6 de mayor riesgo.

ING BANK NV se encuentra adherido al Sistema de Garantía de Depósitos Holandés con una garantía de hasta 100.000 euros por depositante. Consulta más información en [ing.es](https://www.ing.es)

4.2.3. Algunas variaciones estocásticas

El objetivo de estas variaciones es evitar que el algoritmo se quede atrapado utilizando la aleatoriedad.

4.2.3.1. Algoritmo de escalada estocástico

La función heurística f se transforma en una distribución de probabilidad y se realiza un sorteo entre los descendientes que mejoran al actual.

4.2.3.2. Algoritmo de escalada de primera opción

Genera sucesores aleatoriamente hasta que genera uno que es mejor que el actual, esto es una buena estrategia cuando un estado tiene muchísimos sucesores.

4.2.3.3. Algoritmo de escalada de reinicio aleatorio

Trata de generar una función aleatoria cuando está cerca de atascarse con la que continuar el proceso. Es útil cuando las características del problema me permiten generar con facilidad soluciones aleatorias.

4.2.3.4. Enfriamiento simulado

Es un método de búsqueda local que se basa en principios de Termodinámica y es capaz de ver soluciones peores para evitar óptimos locales. La solución inicial se puede generar de forma aleatoria, por conocimiento experto o por otras técnicas algorítmicas como greedy.

La actualización de temperatura también es heurística y el número de vecinos a generar son dependiente de la temperatura. La temperatura inicial y final son parámetros de entrada al algoritmo y esta es difícil asignarle un valor concreto.

Ventajas:

- Puede salir de óptimos locales.
- Es eficiente.
- Es fácil de implementar.

Inconvenientes:

Consulta condiciones aquí



do your thing

WUOLAH

- Requiere de muchas pruebas de ensayo y error, aunque pese a todo el algoritmo puede proporcionar soluciones mejores que otros algoritmos no probabilísticos.

4.2.4. Algoritmos genéticos

Son algoritmos de optimización basados en el proceso de la evolución natural de Darwin, entonces en un proceso de evolución tenemos una **población de individuos** que se pueden reproducir, tener descendencia (a veces con mutaciones que mejoran su idoneidad al entorno) y pueden sobrevivir para la siguiente generación. Su objetivo es encontrar una solución cuyo valor de función objetivo sea óptimo y no parten de un nodo inicial porque hay toda una población.

Términos:

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

Todos los vistos son técnicas de búsqueda local, que explora el vecindario local y son irrevocables, son utilizados para problemas de optimización (tienen poca memoria y útiles en espacios enormes).

4.3. Búsqueda primero el mejor

4.3.1. Greedy (BFS) (Best First Search)

Lo que hace es expandir al hijo más cercano al objetivo y esto se estima mediante una función de evaluación. Si un nodo no tiene hijos se le asigna un coste infinito.

$g(n)$: Valor desde el nodo inicial hasta el actual. Sabemos el valor exacto.

$h(n)$: Estimación del valor desde el nodo actual hasta el objetivo.

4.3.2. Algoritmo A*

Es una búsqueda en grafos donde abiertos es una cola con prioridad ordenada de acuerdo a $f(n)$, que $f(n) = g(n) + h(n)$. Lo cual nos dice que el próximo estado que elijamos será el que tenga menor f .

Para cada nodo hay que recordar el estado, los hijos, el mejor padre y tanto $g(n)$ como $h(n)$.

GESTIÓN DE NODOS REPETIDOS

Para que un nodo sea introducido en ABIERTOS no puede estar ni en abiertos ni en Cerrados.

- Si tenemos dos nodos con la misma función heurística, cogemos el nodo más joven (el que ha entrado más tarde a la cola).
- Si el nodo repetido está en abiertos, se mantiene eligiendo el mejor padre el cual será el que tenga menor valor de g y si es necesario, se cambia el coste del nodo y el padre.
- Si el nodo está en cerrados, hacemos igual que si estuviera en abiertos, solo que ahora en caso de tener hijos si se cambia el padre, haría falta actualizar el coste de los nodos hijos.

CARACTERÍSTICAS

- **Complejidad:** Si existe solución, la encuentra.
- **Admisibilidad:** si hay una solución óptima, la encuentra si:
 - o El numero de sucesores es finito para cada nodo.
 - o $c(n_i, n_j) > X > 0$ en cada arco.
 - o La función $h(n)$ es admisible: $h(n) \leq h^*(n)$ siendo $h^*(n)$ el coste exacto del camino óptimo desde n al objetivo.

CASOS PARTICULARES

Si $h = 0$:

- Coste = 1 \rightarrow A* se convierte en una búsqueda en anchura.
- Coste = $n \rightarrow$ A* se convierte en una búsqueda de coste uniforme.
- Coste = -1 \rightarrow A* se convierte en una búsqueda en profundidad.

Si $g = 0$:

- Si $h = 0 \rightarrow$ NADA
- Si $h = n$ ($n \in R$) \rightarrow A* se convierte en BFS (greedy).

Esto no son apuntes pero **tiene un 10 asegurado** (y lo vas a disfrutar igual).

Abre la Cuenta NoCuenta con el código **WUOLAH10**, haz tu primer pago y llévate 10 €.

Me interesa



1/6

Este número es indicativo del riesgo del producto, siendo 1/6 indicativo de menor riesgo y 6/6 de mayor riesgo.

ING BANK NV se encuentra adherido al Sistema de Garantía de Depósitos Holandés con una garantía de hasta 100.000 euros por depositante. Consulta más información en [ing.es](https://www.ing.es)

Este algoritmo es una extensión de los algoritmos básicos y puede tener algunas dificultades en el proceso.

5. Heurísticas sobre el proceso de búsqueda

Tenemos un entorno de aplicación real, que no nos van a garantizar siempre llegar al objetivo.

5.1. Búsqueda orientada a subobjetivos

Trata de identificar subobjetivos que forman parte del camino al objetivo y lo que hace es avanzar desde el estado inicial hasta el próximo subobjetivo hasta llegar al final. Se realizan búsquedas locales.

5.2. Búsqueda con horizonte

Trata de intentar buscar una solución con una profundidad máxima (horizonte) y plantear el proceso de búsqueda dentro de ese horizonte. A veces es necesario cambiar el criterio de búsqueda del objetivo.

5.3. Búsqueda jerárquica

Partimos de operadores primitivos o de nivel base y operadores más abstractos. Llevamos un proceso de búsqueda para encontrar una solución a un nivel alto y luego las soluciones de nivel alto las utilizamos como subobjetivos para encontrar pequeños trozos de problemas y encontrar la solución a nivel base o primitivo. Es una técnica de resolución de problemas descomponibles.

GRAFO Y/O

Conjunto de nodos asociados a un estado.

Nodos Y: Para resolver el problema asociado a su padre, es necesario resolver todos los componentes que lo suceden. Resuelve todos los hijos, combina la solución, soluciona el nodo y devuelve la solución.

Nodos O: Para resolver un subproblema, basta con procesar uno de los posibles operadores para encontrar una solución. Resolver un hijo y ver si devuelve la solución sino, resolver el siguiente hijo devolver sol etc...

Nodo terminal: Resolver subproblema asociado y devolver la solución.

Consulta condiciones aquí



do your thing

WUOLAH