APUNTES TEMA 2

INTELIGENCIA ARTIFICIAL

Una de las mayores limitaciones de un agente reactivo es que no tiene capacidad para conocer de forma anticipada cómo se transforma el mundo mediante la ejecución de sus acciones. Además, no tiene capacidad para conocer cuál es su meta u objetivo para tomar una decisión correcta entre varias alternativas.

## Diseño de un agente deliberativo

El agente dispone de:

* Un modelo del mundo en el que habita.
* Un modelo de los efectos de sus acciones sobre el mundo **(sabe anticipadamente en qué casilla se encontrará cuando aplica una acción).**
* Puede razonar sobre esos modelos para decidir que hacer para conseguir un objetivo.

La clave es que el agente con varias opciones inmediatas puede usar esa información para proyectar y considerar etapas subsecuentes sobre un recorrido hipotético, tratando de encontrar un camino que le lleve finalmente hasta su objetivo.

Diagrama, Gráfico de cajas y bigotes

Descripción generada automáticamente

Interfaz de usuario gráfica, Diagrama, Texto

Descripción generada automáticamente

### Componentes de la descripción de un problema

El enunciado es el siguiente: un agente está situado en un mundo cuadriculado de dimensiones 4x4, inicialmente en la casilla (2,2). El agente se puede mover a las casillas vecinas que sean transitables. El agente debe alcanzar el premio que está situado en la casilla (4,4). ¿Cómo debe actuar el agente para alcanzar el premio?

#### Estado inicial

El agente está en la casilla (2,2).

#### Estado objetivo

El agente deberá estar en (4,4). El objetivo puede proporcionarse como entrada al agente (un humano) o bien podemos tener un agente pro-activo que sea capaz de formular objetivos para poder llegar a la autonomía plena.

#### Acciones

Las acciones son el repertorio de acciones que puede hacer el agente. Para cada estado particular podemos representar las acciones que se pueden ejecutar.

#### Efectos de las acciones

Se denomina también **modelo de transición, una forma de especificar el estado resultante de aplicar una acción en un estado dado.** Para cada estado particular podemos conocer las acciones que se pueden ejecutar y su efecto, lo que da lugar al estado resultante.

#### Espacio de estados

Es el conjunto de estados alcanzables desde el estado inicial mediante una secuencia de acciones. El espacio de estado de un problema lo definen **el estado inicial, el objetivo y el modelo de transición.**

El espacio de estados tiene la forma de un grafo dirigido en el que cada nodo representa una casilla y cada arco una acción de movimiento de casillas.

Un **camino** en el espacio de estados es una secuencia de estados conectados por una secuencia de acciones.

#### Comprobación de objetivo

Una función se encarga de comprobar que un estado dado es el objetivo que se buscaba. A veces, hay un conjunto explícito de estados objetivos, que puede contener un único estado, y esta función simplemente comprueba que el estado dado es uno de ellos.

#### Coste del camino

Una función se encarga de asignar un coste numérico a cada camino. Se usa para obtener una medida del rendimiento del agente. El coste sería entonces la suma de los costes individuales de cada acción del camino.

Con todos estos elementos podemos definir un problema y podemos representarlo todo en una única estructura que proporcionamos a un algoritmo de resolución de problemas, por tanto, una solución a un problema es una secuencia de acciones que empieza en el estado inicial y permite alcanzar el estado objetivo.

En los problemas en los que las acciones tengan un coste podemos hablar de solución óptima que consiste en el camino con el coste más bajo de entre todas las posibles soluciones.

## El mundo de bloques

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Diagrama

Descripción generada automáticamente

La secuencia de acciones que lleva al agente desde un estado inicial hasta un estado destino se denomina plan. La búsqueda de dicha secuencia se denomina planificación.

* Grafos explícitos: representa el espacio de estados explorado hasta el momento y que se almacena físicamente en memoria.
* Grafos implícitos: grafo que representa el espacio de estados en su totalidad, no está almacenado físicamente en memoria.

Imagen que contiene Mapa

Descripción generada automáticamente

### Búsqueda

La búsqueda se hace por razonamiento regresivo. El grafo explícito se va construyendo por etapas conforme vamos descubriendo sus nodos aplicando acciones. En el primer ejemplo hemos visto cómo el grafo se va construyendo progresivamente a partir del estado inicial. Hay problemas en los que puede ser conveniente partir del objetivo y aplicar las acciones en sentido inverso, haciendo un proceso de exploración del grafo regresivo.

Diagrama

Descripción generada automáticamente

Diagrama

Descripción generada automáticamente

Diagrama

Descripción generada automáticamente

Diagrama

Descripción generada automáticamente

## Sistemas de búsqueda y estrategias

Una vez formulado un problema, hay que resolverlo, con un **proceso de búsqueda,** que tiene que encontrar una secuencia de acciones partiendo del estado inicial como nodo raíz del grafo explícito, los arcos son acciones y los nodos son estados del espacio de estados del problema.

El pseudocódigo del procedimiento general de búsqueda sería el siguiente:

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Dependiendo de la representación que adopte el grafo explícito podemos usar distintas estrategias para explorarlo: estrategias irrevocables y estrategias tentativas (retroactivas y búsqueda en grafos).

### 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 del sistema E para obtener el nuevo estado E’.
* Se borra de memoria E y se sustituye por E’.

El proceso se detiene cuando **E** cumple la condición del estado objetivo.

### Estrategia retroactiva *(Backtracking)*

En memoria solo guardamos un hijo de cada estado, por lo que, se mantiene el camino desde el estado inicial hasta el actual. El grafo explícito es, a fin de cuentas, una lista de nodos desde el inicial al objetivo.

El proceso para cuando hemos llegado al objetivo y no deseamos encontrar más soluciones, o bien no hay más operadores aplicables al nodo raíz del espacio de búsqueda. Además, el proceso **vuelve atrás** cuando se ha encontrado una solución, pero deseamos encontrar otra solución alternativa; cuando se ha llegado a un límite en el nivel de profundidad explorado o en el tiempo de exploración en una misma rama; cuando se ha generado un estado que ya existía en el camino y cuando no existen reglas aplicables al último nodo de la lista.

### Búsqueda en grafos/árboles

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

* En búsqueda en árboles se usa una lista denominada **Frontera (o abiertos)** donde se almacenan los nodos sucesores que se han generado en etapas del algoritmo y que están pendientes de explorar.
* En búsqueda en grafos se usa una lista adicional llamada **Explorados (o Cerrados, o Visitados)** donde se almacenan los nodos que fueron frontera y que se han explorado (para llevar una traza de los nodos visitados).

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

### 

## Algoritmos de búsqueda

### Infraestructura para los algoritmos de búsqueda

Diagrama

Descripción generada automáticamente

Captura de pantalla de un celular

Descripción generada automáticamente

(pag 113?)

### Medidas del comportamiento de un sistema de búsqueda

* **Completitud**: hay garantía de encontrar la solución si esta existe.
* **Optimalidad:** hay garantía de encontrar la solución óptima.
* **Complejidad en tiempo:** ¿Cuánto tiempo se requiere para encontrar la solución?
* **Complejidad en espacio:** ¿Cuánta memoria se requiere para realizar la búsqueda?

### Búsqueda sin información

#### Búsqueda en anchura

Una captura de pantalla de un celular con texto e imagen

Descripción generada automáticamente

* Es un algoritmo **completo,** es decir, encuentra la solución si existe y el factor de ramificación es finito en cada nodo.
* En cuando a **optimalidad,** si todos los operadores tienen el mismo coste, encontrará la solución óptima.
* Si hablamos de **eficiencia,** es buena si las metas están cercanas.
* Uno de los mayores **problemas** es que consume memoria exponencial.

#### Búsqueda con costo uniforme

Una captura de pantalla de un celular con texto

Descripción generada automáticamente

#### Búsqueda en profundidad

La primara versión (búsqueda en grafos) es igual que la búsqueda en anchura cambiando FIFO por LIFO.

La segunda versión es la **búsqueda en** **profundidad retroactiva,** no se usan listas ni bucles, se usa la recursividad:

Texto, Carta

Descripción generada automáticamente

* **Completitud:** no asegura encontrar la solución, puede que cicle en un problema que presente ciclos.
* **Optimalidad:** no asegura encontrar la solución óptima.
* **Eficiencia:** bueno cuando las metas están alejadas del estado inicial, o hay problemas de memoria.

#### Descenso iterativo

#### Una captura de pantalla de un celular Descripción generada automáticamente con confianza media

Trata de integrar las ventajas de la búsqueda en profundidad retroactiva con la búsqueda en anchura. Iterativo es porque aplica iterativamente búsqueda en profundidad retroactiva con un límite de profundidad y lo incrementa en cada iteración.

No es un proceso costoso porque no almacena los nodos visitados.

#### Búsqueda bidireccional

Esta búsqueda integra búsqueda desde el estado objetivo y búsqueda desde el estado inicial, con la esperanza de que se encuentren. Sería búsqueda progresiva y búsqueda regresiva. La complejidad es mucho mayor que si recorremos todo el espacio de búsqueda (sería solo la suma de complejidades por separado).

### Búsqueda con información

#### Heurísticas

Si se tiene conocimiento perfecto del entorno, podemos usar un algoritmo exacto, si no lo tenemos, empleamos la búsqueda sin información. No obstante, en la mayor parte de los problemas que resolvemos, nos encontramos en posiciones intermedias. Una **heurística** es conocimiento parcial sobre un problema/dominio que permite resolver problemas eficientemente en ese problema/dominio.

Las **heurísticas** son criterios para decidir cuál de entre varias acciones promete ser la mejor para alcanzar una determinada meta. En IA, es un método para resolver problemas que en general no garantiza la solución óptima, pero que en media produce resultados satisfactorios en la resolución de un problema. Encapsulan el conocimiento específico que se tiene sobre un problema y sirve de guía para que un algoritmo de búsqueda pueda encontrar una solución válida aceptable.

Imagen que contiene Texto

Descripción generada automáticamente

#### Métodos de escalada

Si dibujamos las soluciones como puntos en el espacio, una búsqueda local consiste en seleccionar la solución mejor en el vecindario de una solución inicial, e ir viajando por las soluciones del espacio hasta encontrar un óptimo (local o global).

Diagrama

Descripción generada automáticamente

A continuación, se adjuntan los pseudocódigos para **Algoritmo de escalada simple y Algoritmo de escalada por la máxima pendiente,** respectivamente.

Texto

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

* **Completitud:** no tiene por qué encontrar la solución.
* **Admisibilidad:** no siendo completo, aun menos será admisible.
* **Eficiencia:** rápido y útil si la función es monótona (de)creciente.

Gráfico

Descripción generada automáticamente con confianza baja

### Algunas variaciones estocásticas

**Algoritmo de enfriamiento simulado**

Es un método de búsqueda local que se basa en principios de Termodinámica. Al contrario que otros métodos de ascensión de colinas, permite visitar soluciones **peores que la actual** para evitar óptimos locales.

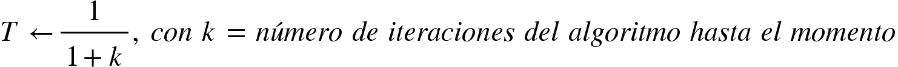
* Los estados por los que pasa el sistema físico de partículas equivalen a las soluciones factibles del algoritmo.
* La energía E del estado actual del sistema es el valor de la función objetivo de la solución actual.
* Un cambio de estado en el sistema equivale a explorar el entorno de una solución y viajar a una solución vecina.
* El estado final estable (congelado) es la solución final del algoritmo.

Texto

Descripción generada automáticamente

La **solución inicial** se puede generar de forma aleatoria, por conocimiento experto, o por medio de otras técnicas algorítmicas como greedy.

La **actualización de temperatura** es heurística y hay varios métodos.

* T espacio flecha izquierda espacio alfa por T, con alfa espacio pertenece espacio abrir paréntesis 0 coma 1 cerrar paréntesis
* 

El **número de vecinos a generar** puede ser, fijo N(T) = cte, dependiente de la temperatura N(T) = f(T), etc.

Tanto la temperatura inicial como la temperatura final **Ti y Tf** son parámetros de entrada del algoritmo. Aún así, es difícil asignar un valor concreto a la temperatura final, por lo que la condición de parada se suele sustituir por un número específico de iteraciones a realizar.

* Al ser un método probabilístico, tiene capacidad para salir de **óptimos locales.**
* Es **eficiente.**
* Es fácil de **implementar.**
* No obstante, ajustar los parámetros óptimos para la temperatura inicial, la actualización de esta, el número de vecinos a generar y el número de iteraciones óptimo es una tarea que requiere de **muchas pruebas de ensayo y error.**

#### Algoritmos genéticos

Son algoritmos de optimización basados en el proceso de la evolución natural de Darwin.

Existe una **población de individuos.** Los mejores de su entorno se **reproducen y tienen descendencia** (que a veces **muta**). Los más adecuados según diversos criterios se mantienen para la siguiente generación. Su objetivo es encontrar una solución cuyo valor de función objetivo sea óptimo.

* **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.

Diagrama

Descripción generada automáticamente

### Búsqueda primero el mejor

Pag 173