

## TEMA 2 BÚSQUEDA SIN INFORMACIÓN (a ciegas, no heurística)

- 1.- Formulación de problemas en espacios de estados.
- 2.- Ejemplos de problemas. Concepto de solución.
- 3.- Búsqueda en anchura.
- 4.- Búsqueda en profundidad. Profundidad limitada. Profundidad iterativa.
- 5.- Búsqueda de coste uniforme.
- 6.- Búsqueda bidireccional.
- 7.- Características y comparación de las distintas búsquedas.

### BIBLIOGRAFÍA

- Poole-Mackworth, secs. 3.1-3.5
- Russell-Norvig II.3
- Nilsson II.8
- Galán-Boticario-Mira, Problemas resueltos de IA, Addison-Wesley, cap.1

#### 1.- FORMULACIÓN DE PROBLEMAS EN ESPACIOS DE ESTADOS.

Los problemas de que se va a tratar en este tema son los más sencillos. Se trata de situaciones en las que el agente tiene unos fines o metas, que puede perseguir buscando sin incertidumbre en la representación que posea sobre un micromundo, representación que consiste en un modelo formal (estructuras de datos sobre los que se pueda computar) del mismo, que lo represente abstractamente en sus rasgos fundamentales (en relación con el problema) como un **espacio de estados** (estado que el agente puede identificar cuando está en ellos) donde uno o algunos de los mismos son **iniciales** o de partida, desde los cuales se ha de llegar o encontrar, mediante **transformaciones o cambios estado** a uno o varios **estados meta, solución u objetivo**.

Los estados meta se identifican o valoran mediante algún o algunos **test** o **criterios** de que dispone el agente. Los cambios de estado se producen mediante la aplicación de **operadores de cambio de estado** o de **funciones sucesor** que determinan totalmente los estados a los que se pueda pasar a partir de uno dado (estados sucesores del mismo), y que el agente conoce perfectamente, como conoce el **coste** (si existe) de dichos cambios.

#### 2.-EJEMPLOS DE PROBLEMAS. CONCEPTO DE SOLUCIÓN.

Para ejemplos, ver los clásicos (dos jarras, misioneros y caníbales, n-puzzle , n-reinas etc. ) presentados en la relación de ejercicios 2.1 y en las secciones introductorias de las referencias bibliográficas dadas.

Por **resolución** del problema se entiende el proceso de determinar la serie de acciones de cambio de estado que hay que realizar (el camino a seguir) para llegar desde un estado inicial a uno, o varios estados meta o solución, proceso sometido o no, según el contexto del problema, a la consecución de un coste mínimo (**solución óptima**) o a la obtención de estados que, de acuerdo con algún criterio de valoración dado, sean lo bastante satisfactorios

Cuando un problema haya sido formulado de este modo, basta identificar cada uno de sus estados con un nodo de un grafo, y cada uno de los operadores de cambio de estado con la arista que une los nodos correspondientes al estado de partida y al de llegada, y se tendrá el problema convertido en uno de búsqueda en grafos.

## PROCEDIMIENTO GENERAL DE BÚSQUEDA EN GRAFOS

```

1: Procedure Búsqueda( $G$ ,  $S$ ,  $EsMeta$ )
2:   Inputs
3:      $G$ : grafo con nodos  $N$  y arcos  $A$ 
4:      $S$ : conjunto de nodos iniciales
5:      $EsMeta$  : función booleana de los estados (test de solución)
6:   Output
7:     camino de un elemento de  $S$  a nodo para el que  $EsMeta$  dé verdad
8:     o  $\perp$  (no encontrado camino) si no hay caminos solución
9:   Local
10:     $ABIERTOS$ : conjunto de caminos(listas de nodos de cada camino)
11:     $ABIERTOS \leftarrow \{(s): s \in S\}$ 
12:    while ( $ABIERTOS \neq \{\}$ )
13:      select y remove  $(s_0, \dots, s_k)$  de  $ABIERTOS$ 
14:      if ( $EsMeta(s_k)$ ) then
15:        return  $(s_0, \dots, s_k)$ 
16:       $ABIERTOS \leftarrow ABIERTOS \cup \{(s_0, \dots, s_k, s): (s_k, s) \in A\}$ 
17:    return  $\perp$ 
```

Nota:  $ABIERTOS$  suele ser llamada también *FRONTERA*.

Los diversos métodos de búsqueda (anchura , profundidad, óptima o de coste uniforme etc.) se diferencian esencialmente en cómo se seleccione - línea 13 del algoritmo- el elemento de la lista *ABIERTOS* ( tratándola como cola, como pila, como lista de prioridad según el coste acumulado etc) :

#### ***LOS PRINCIPALES MÉTODOS DE BÚSQUEDA DESINFORMADA SON:***

#### **3.-BÚSQUEDA EN ANCHURA**

La lista *ABIERTOS* se gestiona como una cola, donde los caminos se van extrayendo de la lista en el orden en que entraron a ella (FIFO), y sus nodos terminales se van examinando (mediante la función *test EsMeta*) para presentar al camino como solución (si lo fuera) o generar los sucesores del nodo terminal e incorporar los caminos resultantes de “empalmar” las aristas correspondientes a cada sucesor al camino examinado e incorporar estos caminos alargados a la lista *ABIERTOS* .

#### **4.-BÚSQUEDAS EN PROFUNDIDAD**

-**Búsqueda en profundidad:** *ABIERTOS* se gestiona como pila (LIFO).

Variantes de profundidad:

- **Profundidad acotada  $L$**  : se procede como en la búsqueda en profundidad, pero hasta un límite preestablecido  $L$  para la misma , es decir, “podando” las ramas del árbol de búsqueda con dicha longitud  $L$ .

-**Profundidad iterativa:** se establece una sucesión creciente de límites  $L_1 < L_2 < L_3 < \dots$

y se repite, desde el principio, una serie de búsquedas en profundidad limitada  $L_i$  hasta que una de ellas encuentre la solución.

Nota: Tanto en la búsqueda en anchura como en la de profundidad y sus variantes, se da por supuesto que los costes de las aristas (es decir, de aplicar los operadores para transitar de unos nodos a otros) son siempre 1, de modo que los “gastos” hechos al recorrer los caminos dependen sólo de (son iguales a) el número de etapas de los mismos, que coincide con la profundidad de las ramas del árbol de búsqueda correspondientes.

#### **5.-BÚSQUEDA DE COSTE UNIFORME**

-Búsqueda llamada “**optimal**”, “**de menor coste**” o “**de coste uniforme**” (sic), se aplica cuando los costes de los arcos entre nodos no son constantes, de modo que el gasto hecho al seguir

los caminos no depende sólo del número de etapas de los mismos, sino que caminos con menor número de etapas que otros puedan resultar más caros que éstos. La lista *ABIERTOS* se trata como una lista de prioridad, eligiendo primero para ser examinado con *EsMeta* aquel camino de la lista que menor gasto acumulado lleve (o uno de los éstos si hubiera varios con idéntico menor gasto).

## 6.-BÚSQUEDA BIDIRECCIONAL

La **Búsqueda bidireccional** es aplicable sólo cuando se conocen los operadores de paso inversos (es decir, se sabe cómo generar todos los “padres” de cada nodo) y además es conocido explícitamente algún estado solución. En esta situación, se podría resolver el problema del modo habitual, *buscando hacia adelante* desde el estado inicial hacia la meta, o también se podría *buscar hacia atrás*, es decir desde la meta hacia el inicial.

Pero una tercera posibilidad es hacer ambas búsquedas simultáneamente, en pasos alternados, hasta que los segmentos de caminos generados se “empalmen” al coincidir en un estado común (es decir, cuando aparezca un estado común en ambas listas de abiertos). Ello ocurrirá necesariamente si por lo menos una de las dos búsquedas se efectúa en anchura, con lo cual siempre aparecen en su lista *ABIERTOS* nodos de cada uno de los caminos solución.

Se podría hacer las dos búsquedas en anchura, pero hacerlo combinando los dos tipos permite economizar gasto en memoria en la dirección en que el factor de ramificación sea mayor. En cualquier caso, el número de operaciones de examen y generación de sucesores disminuirá sustancialmente, al llegar cada una de las dos búsquedas sólo hasta la mitad de profundidad. Esta ventaja en coste computacional hay que ponerla en balance frente a la desventaja de tener que comprobar en cada paso (dado que en principio la longitud del camino solución será desconocida) si ha aparecido alguna coincidencia entre los nodos de las dos listas de abiertos.

## 7.- CARACTERÍSTICAS Y COMPARACIÓN DE SOLUCIONES

Para comparar la eficacia de los métodos se estudian generalmente las siguientes características:

**Compleitud:** un método es completo si siempre encuentra solución, cuando esta existe.

**Optimalidad:** cuando encuentra una solución, esta es la más cercana al estado de partida, sea en número de pasos, sea en coste acumulado de recorrerlos cuando este no es constante.

**Complejidad espacial o en memoria:** dada por la estimación del tamaño de las listas a gestionar y mantener, en función del factor de ramificación y la profundidad de las soluciones

**Complejidad temporal o en operaciones:** dada por la estimación de la cantidad de nodos o estados a generar y examinar, también en función del factor de ramificación y la profundidad de las soluciones.

### **Cuestiones 2.1**

- 1) Comprobar que la búsqueda en anchura es un caso particular de la búsqueda óptima.
- 2) Comprobar, mediante ejemplos, que las búsquedas en profundidad y en profundidad acotada, pueden fallar, es decir, no encontrar ninguna meta, aunque las haya accesibles desde un estado inicial (es decir, estas búsquedas no son **completas**).
- 3) Razonar por qué las búsquedas en anchura, profundidad iterativa y óptima no pueden fallar, es decir, si existe alguna meta accesible desde un estado inicial, acaban encontrándola (es decir, son **completas**).
- 4) Comprobar que las búsquedas en anchura y óptima son **óptimas**, es decir que siempre encuentran primero el mejor camino (el de menor longitud=profundidad=gasto o menor gasto respectivamente)
- 5) Comprobar que para un problema con soluciones, en la búsqueda en anchura, siempre hay en la lista *ABIERTOS* algún nodo de cada camino solución. ¿Ocurre lo mismo en la búsqueda en profundidad? ¿y en la óptima?
- 6) Comprobar que si  $L_{i+1} = L_i + 1$  para cada  $i$  también la búsqueda en profundidad iterativa es óptima. En otro caso puede ser subóptima, es decir, puede encontrar una solución con un camino cuya longitud difiera de la del óptimo como máximo en  $L_{i+1} - L_i$  para los  $L_i$  y  $L_{i+1}$  correspondientes.

### **Ejercicios 2.1**

Para los siguientes problemas interprete, abstraiga y formalice (es decir, describa formalmente en términos computables): (a) alguna formulación del espacio de estados; (b) descripción de los estados iniciales; (c) funciones u operadores de transición entre estados (o acciones de un agente que cambien el estado); (d) función que establece el coste del paso de unos estados a otros; (e) conjunto de estados meta o solución y criterio que permite valorarlos como tales. Cuando se pueda, presente varias formulaciones alternativas.

1.- Problema de las jarras: Dadas dos jarras de tres y de cuatro litros de capacidad, sin ninguna graduación, averiguar cómo llegar a tener en una de ellas exactamente dos litros sin usar medidas de capacidad auxiliares.

2.- Misioneros y caníbales: Tres misioneros acompañados de tres caníbales han de atravesar en su marcha un río usando una barca de remos de dos plazas.

3.- N-reinas: Colocar en un tablero como los de ajedrez de NxN casillas tantas reinas como quepan sin poder agredirse.

4.- N-puzzle: En un tablero cuadrado de N=mxm casillas hay N-1 fichas numeradas. Si éstas pueden moverse hacia una casilla contigua que esté vacía, obtener una configuración predeterminada a partir de otra inicial mediante los movimientos correspondientes.

5.- Coloreado de mapas: Dado un mapa (de visión de una zona o zonas cerradas del plano en subzonas distintas de área no nula), pintarlo con la menor cantidad de colores posibles.

6.- Construcción de crucigramas: Dados casilleros rectangulares, con segmentos dados de filas o columnas coloreados de negro, asignar palabras a los segmentos verticales u horizontales completos restantes de modo que se ocupen todos.

7.- Problemas criptoaritméticos: Dadas ecuaciones formadas por palabras (como HOCUS + POCUS = PRESTO) encontrar cifras que cambiadas por las letras originen operaciones aritméticas correctas.

8.- Laberintos: Dada una superficie divisible en casillas, con algunas de ellas señaladas como obstáculos (prohibidas), encontrar un camino constituido por casilla adyacentes desde una dada de ellas (entrada) hasta otra dada (salida).

9.- Granjero: Un granjero que transporta un conejo y una col al mercado en compañía de su perro ha de atravesar un río en una barca de remo en la que caben dos entes.

## **Ejercicios 2.2**

- 1) Sea un problema formulado como de búsqueda en espacio de estados. Si el árbol de búsqueda tiene un factor de ramificación constante  $r$  y hay una sola solución a profundidad  $p$ , estúdiense para los distintos tipos de búsqueda cual sería el número de nodos a generar y analizar en el mejor de los casos posible, en el peor y en uno intermedio.
- 2) Calcúlese el tamaño de la lista de abiertos en la situación anterior, para los diversos métodos.
- 3) Compárese el coste espacial y temporal de la búsqueda en profundidad iterativa con los de las búsquedas en anchura y en profundidad (comprobando que no es esencialmente peor ).
- 4) Dar ejemplos de problemas en los que sea preferible la búsqueda en profundidad sobre la de anchura y al revés.
- 5) Un turista exótico trata de encontrar la dirección de un hotel, situado en una calle de una urbanización a lo largo de una carretera, en la que los inmuebles están indicados (rotulados) con numeración romana, cuyo sistema desconoce. Formular el problema como uno de búsqueda, examinar la pertinencia de los diversos métodos y resolver, en un caso concreto, mediante uno de ellos.
- 6) Resolver formalmente, mediante varios métodos de búsqueda en el espacio de estados el problema que, formulado tiene como estados  $E=\{0,1,2,3,4,5,6,7,8,9,a,b,c\}$  como transiciones  $T = \{(0,b), (b,0), (8,a), (5,a), (5,6), (8,6), (6,3), (3,6), (b,9), (5,4), (4,5), (7,4), (1,8), (3,b), (4,1), (4,7), (3,4), (6,5), (2,c), (c,2), (b,2), (b,3), (a,5) \text{ y } (a,8)\}$ , en con  $a$  como el estado inicial y donde las metas son **9** y **1**.