

## Tema 2. Estrategias de búsqueda

## Tema 2. Estrategias de búsqueda

### Contenidos

1. Introducción
2. Especificación de problemas
3. Caracterización del problema
4. Problemas clásicos
5. Estrategias de búsqueda básicas
  1. Estrategia irrevocable
  2. Estrategia tentativa
    1. No informadas
    2. Informadas
6. Esquema de tipos básicos de búsqueda.
7. Estrategias híbridas.
8. Búsqueda heurística
  1. Conceptos básicos.
  2. Búsqueda A\*. Búsqueda óptima.
  3. Nivel de información heurístico.
  4. Ejemplos de heurísticas.
  5. Inconvenientes de mantener la admisibilidad
  6. Relajación de la restricción de optimalidad
9. Bibliografía

## Tema 2. Estrategias de búsqueda

### Objetivos

- Conocer la importancia de la búsqueda en IA
- Conocer los sistemas de producción en búsqueda
- Ser capaz de especificar un problema mediante un Sistema de Producción (SP)
- Aprender problemas clásicos de IA
- Entender las distintas estrategias de búsqueda

## 1. Introducción

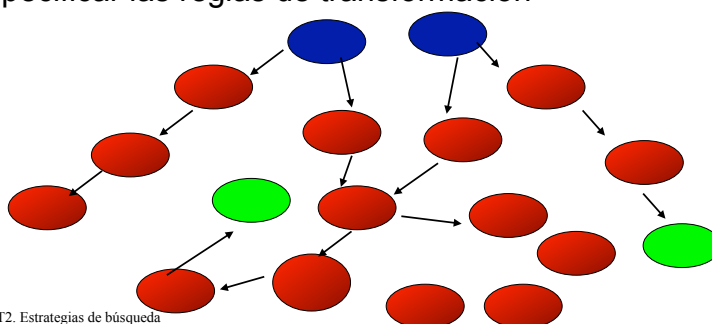
¿Qué podemos resolver mediante búsqueda?

- Algunos problemas que se resuelven con técnicas de búsqueda:
  - Problemas de búsqueda en rutas
    - Planificación líneas aéreas
    - Rutas en redes de computadores
  - Problemas turísticos
  - El viajante de comercio
  - La Distribución VLSI
  - Navegación de un robot
  - Secuenciación para el ensamblaje automático
  - Diseño de proteínas
  - Búsqueda en Internet
  - ...

## 2. Especificación de problemas

¿Cómo resolver un problema? (I)

- Definición del problema como una búsqueda en un espacio de estados
  - Definir un espacio de estados.  
Definición por extensión que evite enumerar todos los estados que contiene
  - Especificar el/los estados iniciales
  - Especificar los estados meta
  - Especificar las reglas de transformación



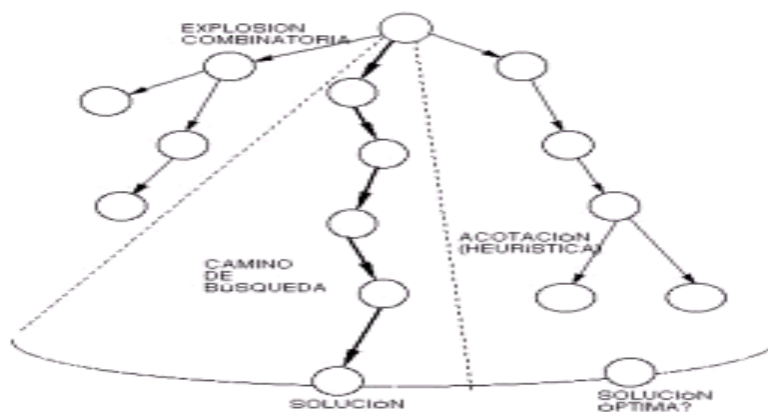
T2. Estrategias de búsqueda

5

## 2. Especificación de problemas

¿Cómo resolver un problema? (II)

El proceso de búsqueda se puede realizar explorando un árbol (**árbol de búsqueda**) o en general un grafo (eliminando repeticiones de estados)



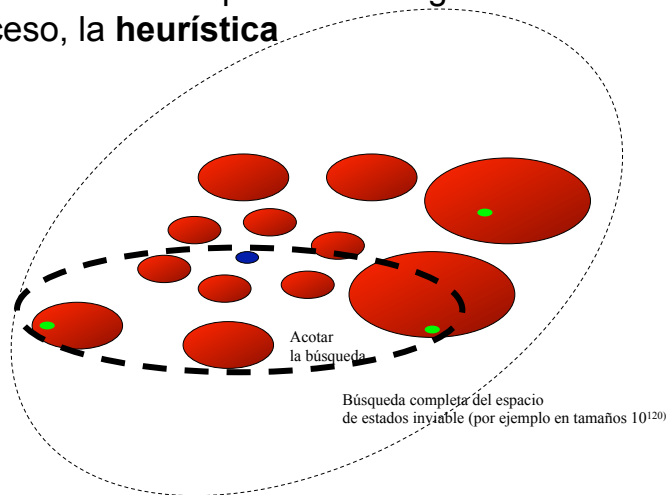
T2. Estrategias de búsqueda

6

## 2. Especificación de problemas

### ¿Cómo resolver un problema? (III)

- ¿Cuál es el principal problema de la búsqueda de estados?
  - Introducir una componente inteligente en el proceso, la **heurística**



T2. Estrategias de búsqueda

## 2. Especificación de problemas

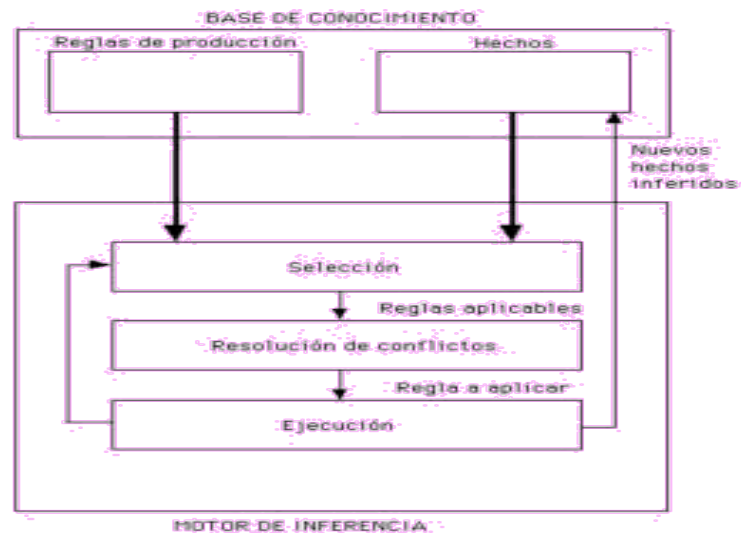
### Sistemas de producción (I)

- Sistemas de producción
  - Formalizan los problemas de búsqueda de estados
  - Propuesto por POST en 1943
  - Un sistema de producción es una terna (BH, RP, EC) donde:
    - **BH (Base de Hechos).**  
Conjunto de representaciones de uno o más estados por los que atraviesa el problema. Constituye la estructura de datos global
    - **RP (Reglas de Producción).**  
Conjunto de operadores para la transformación de los estados del problema, es decir, de la base de hechos. Cada regla tiene dos partes:
      - Precondiciones
      - Postcondiciones
    - **EC (Estrategia de control).**  
Determina el conjunto de reglas aplicables mediante un proceso de pattern-matching y resuelve conflictos entre varias reglas a aplicar mediante el filtrado

T2. Estrategias de búsqueda

## 2. Especificación de problemas

### SP. Estructura y características (I)



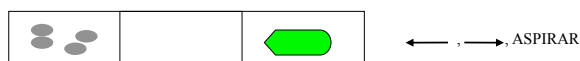
## 2. Especificación de problemas

### SP. Estructura y características (II)

- Algoritmo SP( $BH_0, RP, EC$ )  
 $BH = BH_0$   
**repetir**  
      $R = \text{Aplicables}(BH);$   
      $R_i = \text{Seleccionar}(R);$   
      $BH = R_i(BH);$   
**hasta**  $\text{CondicionesTerminación}(BH)$

1. Determinar el conjunto de reglas aplicables y aplicar el filtrado
2. Aplicar la regla seleccionada. La selección depende de la información de control
3. Repetir hasta que se den las condiciones de terminación

→ Ejercicio.



### 3. Caracterización del problema (I)

#### ¿Cómo elegir la EC?

- ¿Se puede **descomponer** el problema en subproblemas independientes?
- ¿Pueden ignorarse o al menos deshacerse pasos hacia la solución si se constata que son erróneos?
  - **Ignorables** -> Demostración de teoremas
  - **Recuperables** -> Podemos retroceder
  - **Irrecuperables** -> No se puede retroceder
- ¿Es el universo de discurso **predecible**?
- La **bondad** de una solución.
  - ¿Es absoluta o relativa?
- La solución,
  - ¿es un **estado** o es un camino?
- ¿Qué papel desempeña **el conocimiento**?

### 4. Problemas clásicos (I)

- **El problema de las jarras de agua**: Se tienen dos jarras de agua
  - Una de tres litros y otra de cuatro.
  - No disponen de marcas de medición.
  - Las jarras se pueden llenar y también vaciar.

**¿Cómo se puede lograr tener exactamente dos litros de agua en la jarra de cuatro?**

El espacio de estados:  $(x, y)$

- »  $x = 0, 1, 2, 3$  ó  $4$  e  $y = 0, 1, 2$  ó  $3$ .
- »  $x$  representa el número de litros de agua en la jarra de cuatro litros
- »  $y$  ídem en la jarra de tres litros.

*El estado inicial corresponderá al estado  $(0, 0)$  y el estado final (objetivo) al  $(2, n)$  donde  $n$  significa cualquier valor en la jarra de tres litros.*

**Ejercicio:** Plantear las reglas (precondiciones y postcondiciones) y aplicarlas para solucionar el problema

## 4. Problemas clásicos (II)

- Las reglas de producción

1	$(x, y)$ si $x < 4$	$\rightarrow (4, y)$
2	$(x, y)$ si $y < 3$	$\rightarrow (x, 3)$
3	$(x, y)$ si $x > 0$	$\rightarrow (x-d, y)$
4	$(x, y)$ si $y > 0$	$\rightarrow (x, y-d)$
5	$(x, y)$ si $x > 0$	$\rightarrow (0, y)$
6	$(x, y)$ si $y > 0$	$\rightarrow (x, 0)$
7	$(x, y)$ si $x+y \geq 4$ e $y > 0$	$\rightarrow (4, y-(4-x))$
8	$(x, y)$ si $x+y \geq 3$ e $x > 0$	$\rightarrow (x-(3-y), 3)$
9	$(x, y)$ si $x+y \leq 4$ e $y > 0$	$\rightarrow (x+y, 0)$
	$(x, y)$ si $x+y \leq 3$ e $x > 0$	$\rightarrow (0, x+y)$

Jarra 1   Jarra 2   Regla a aplicar

0	0	2
0	3	9
3	0	2
3	3	7
4	2	5
0	2	9
2	0	5

## 4. Problemas clásicos (III)

- Problema del 8-puzzle

2	8	3
1	6	4
7		5

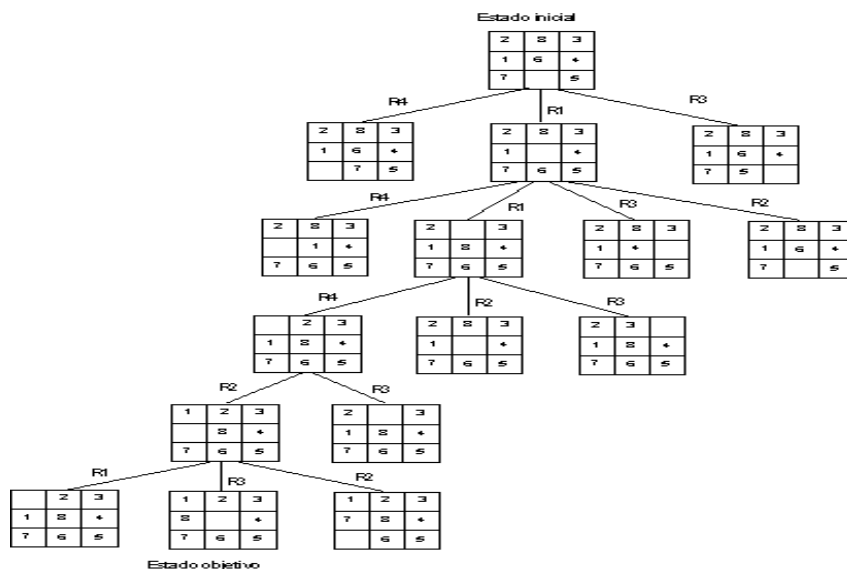
→

1	2	3
8		4
7	6	5

- Espacio de estados (e)
- Consta de todas las configuraciones de las 8 fichas y el cuadro vacío (Card (e) = 9!) (en general  $N!$ , siendo  $N-1$  el número de fichas).
- Estados iniciales (ei)
- La búsqueda puede arrancar desde cualquier combinación que no cumpla la condición de estado meta.
- Estados finales (ef)
  - Configuración que cumpla la condición de estado meta
- Reglas de producción (R1 Vacía arriba, R2 v. abajo, R3 v. a la derecha, R4 v. a la izquierda)

## 4. Problemas clásicos (IV)

- La solución consiste en presentar las reglas aplicadas. En este caso: R1-R1-R4-R2-R3.

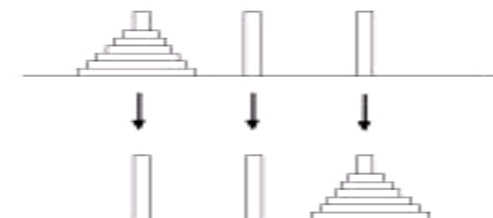


T2. Estrategias de búsqueda

15

## 4. Problemas clásicos (V)

- El problema del viajante de comercio
- El dilema del granjero
- El dilema de los misioneros
- Las torres de Hanoi



T2. Estrategias de búsqueda

16



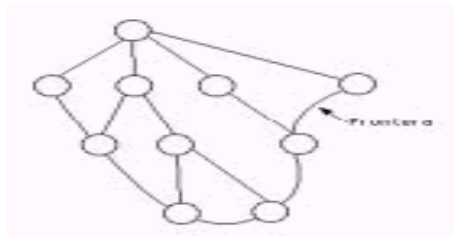
## Ejercicio

- Modela el siguiente enunciado como un sistema de producción y resuélvelo:
  - Ayuda a la familia a cruzar el puente. Ten en cuenta que es de noche y necesitan la linterna para cruzar.
    - Cada miembro cruza a una velocidad distinta (1 seg., 3 seg., 6 seg., 8 seg y 12 seg.
    - El puente sólo resiste un máximo de 2 personas.
    - Un par debe cruzar a la velocidad del miembro más lento.
    - La linterna sólo dura 30 segundos.



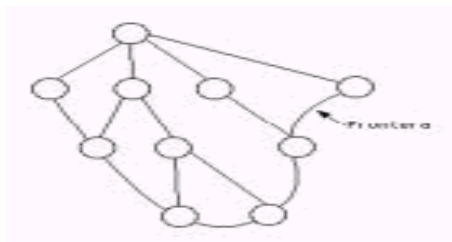
## 5. Estrategias de búsqueda básica (I)

- **Ciclo de control básico** dentro de una estrategia de control:
  - E1 Exploración de la frontera.
  - E2 Cálculo de reglas aplicables
  - E3 Resolución de conflictos.
  - E4 Aplicación de regla y memorización de estado.
    - En el paso E1 debemos seleccionar el **mejor estado** candidato entre aquellos que no hayan sido todavía seleccionados. Hablamos pues de mantener una frontera de búsqueda, compuesta por los estados que no han sido todavía expandidos



## 5. Estrategias de búsqueda básica(II)

- En el siguiente paso E2 debemos de obtener la aplicabilidad de las reglas sobre el estado seleccionado, es decir, qué reglas entre todas las posibles son aplicables.
- En el paso E3 se elige la regla a aplicar definitivamente. En este paso podemos **incorporar el conocimiento para decidir qué regla nos acercará más a la solución**.
- Por último, en el paso E4 aplicamos la regla seleccionada y el resultado lo almacenamos dentro del árbol de búsqueda, actualizando éste.



T2. Estrategias de búsqueda

19

## 5. Estrategias de búsqueda básica(III)

- Las estrategias a considerar las podemos subdividir en:
  - **Irrevocables**
    - Presenta la característica de que no se permite la vuelta atrás. Mantenemos una frontera unitaria.
  - **Tentativas**
    - La búsqueda es **multi o mono camino**. Se mantienen estados de vuelta atrás por si el estado actual no llega a buen fin.
- Los requerimientos exigibles a las estrategias empleadas son:
  - En todo momento se debe producir un avance y este debe ser dirigido.
  - Este avance debe ser metódico

T2. Estrategias de búsqueda

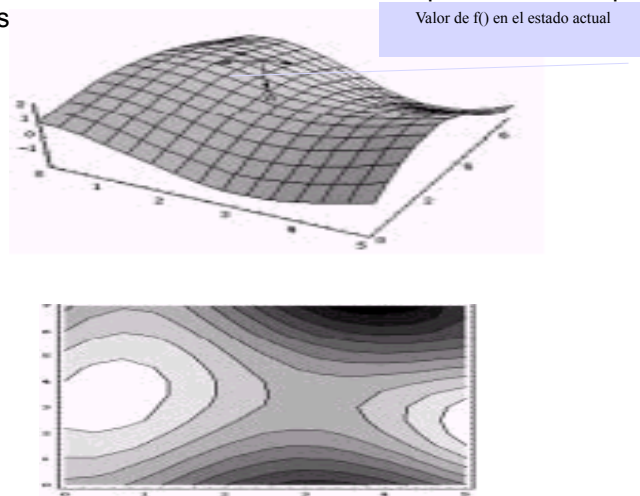
20

## 5.1 Estrategia Irrevocable

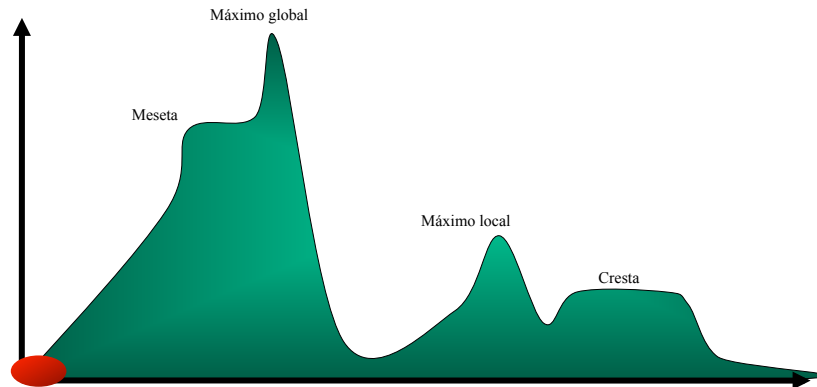
- Búsqueda irrevocable. Descenso por gradiente
  - Supuestos de partida:
    - Disponemos del suficiente conocimiento local.
    - Las equivocaciones sólo alargan la búsqueda
  - Pretendemos buscar optimalidad global a partir de la local.
    - Debemos especificar una función de evaluación  $f()$  que nos proporcione un mínimo (máximo) en el estado final. En la literatura esta estrategia aparece como búsqueda por gradiente o ascenso (descenso).
  - En el ciclo de control, concretamente en el paso E3 se debe elegir aquella regla que optimiza localmente  $f()$ 
    - En el problema del 8-puzzle, podemos definir una función de evaluación sumando el número de fichas descolocadas. En el estado final, el resultado de la función es cero, por lo que tenemos el mínimo buscado.

## 5.1 Estrategia Irrevocable

- Gráfico de la función de evaluación del problema del 8-puzzle. Des



## 5.1 Problemas



Problemas de las búsquedas irrevocables

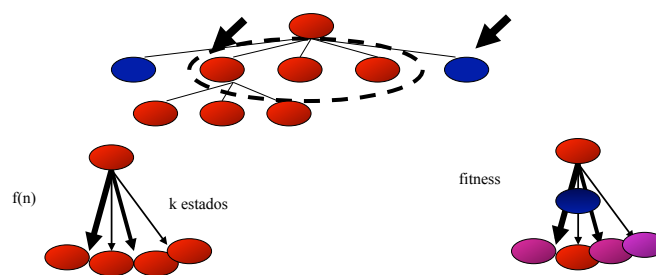
- Mesetas
- Máximos locales
- Crestas

T2. Estrategias de búsqueda

23

## 5.1 Otras estrategias

- Otras soluciones, para aplicarlas es necesario que no importe el camino al objetivo:
  - Ascenso por gradiente **estocástico**
  - Ascenso por gradiente de primera opción
  - Ascenso por gradiente de reinicio aleatorio
  - Temple simulado (simulated annealing)
  - Búsqueda por haz local
  - Algoritmos genéticos



T2. Estrategias de búsqueda

24

## 5.2.1 Estrategia Tentativa

### Tentativas No informadas

- **Estrategias desinformadas (No informadas)**  
“son ciegas en el sentido de que el orden en el cual la búsqueda progresa **no depende de la naturaleza de la solución que buscamos**”
  - **Búsqueda en profundidad**
    - También conocida como primero el mejor, es una variación del conocido **backtracking**.
    - El siguiente estado a desarrollar es el de mayor profundidad en el grafo.
  - **Búsqueda en anchura**
    - Asigna **mayor prioridad** a aquellos nodos que se encuentran a **menor profundidad** en el grafo. De esta manera nos estamos asegurando que la búsqueda se realiza por todo el grafo.
  - **Coste uniforme**
    - Esta estrategia selecciona aquel nodo tal que la **suma de los costes de aplicación** de las reglas en el camino desde el nodo inicial sea mínima. Esta estrategia es similar al procedimiento en anchura cuando el coste de aplicación de cada regla es unitario.
- **Pregunta:** En grafos finitos ¿Se asegura obtener la solución óptima con todas las estrategias anteriores?

## 5.2.2 Estrategia Tentativa

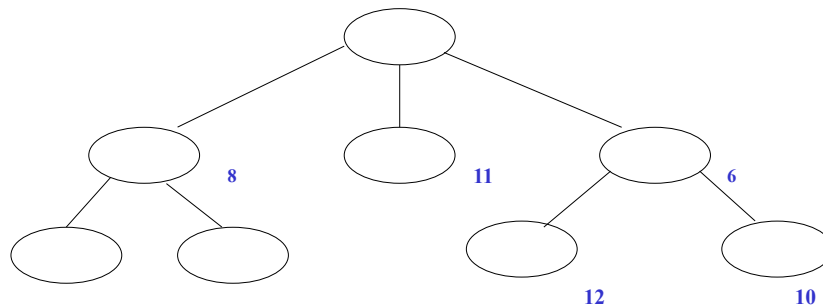
### Tentativas Informadas

- Al contrario de las “ciegas” las informadas sí que van a disponer de información de lo **prometedor que es un nodo** para llegar desde él a la solución.
- Estimación de lo que nos va a costar **llegar** a la **solución óptima**. Esta función la vamos a denominar **heurística**  $h(n)$ .
- En general vamos a disponer de una función  $f(n)$  que va a estimar el coste del camino de coste mínimo desde el nodo inicial BHo hasta un nodo objetivo, condicionado este camino a pasar por  $n$ .
- En esta estrategia el criterio de selección de un nodo de la lista de frontera es el de menor valor de  $f()$ .
- En la literatura aparecen estas estrategias como **Best-first** (primero el mejor).

## 5.2.2 Estrategia Tentativa

- Algoritmo de búsqueda **tentativa informada**

El nodo a expandir es aquel que proporciona el coste más pequeño para ir desde el nodo inicial hasta un nodo objetivo



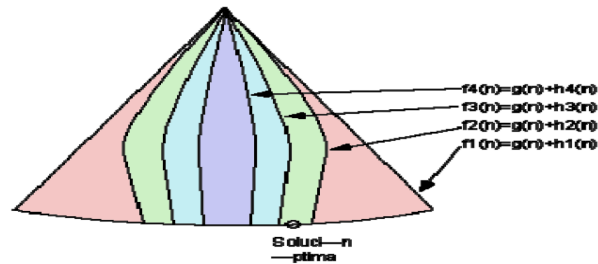
## 5.2.2 Estrategias Informadas. Funciones de Evaluación

- Algoritmos A (de aditivos).
  - Presentan una función de evaluación de la forma:  $f(n) = g(n) + h(n)$
  - donde:
    - $g(n)$  Estimación del coste del camino de **coste mínimo** desde el estado inicial hasta el nodo  $n$ .
    - $h(n)$  Estimación del **coste del camino** de coste mínimo desde  $n$  hasta algún nodo objetivo o meta. Esta función incluye el conocimiento heurístico sobre el problema a resolver.
- Nuestro objetivo
  - Diseñar la heurística
  - Elección compleja
    - Podemos llegar a dejar fuera el óptimo
    - Podemos explorar demasiados nodos

## 5.2.2 Estrategias Informadas. Funciones de Evaluación (II)

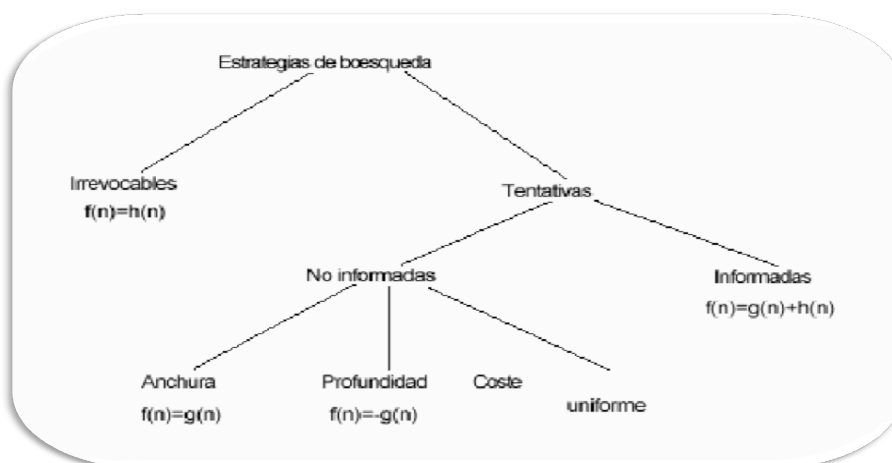
En la figura:

- $h_1(n) = 0$ ,
- $h_2 < h_3 < h_4$ .



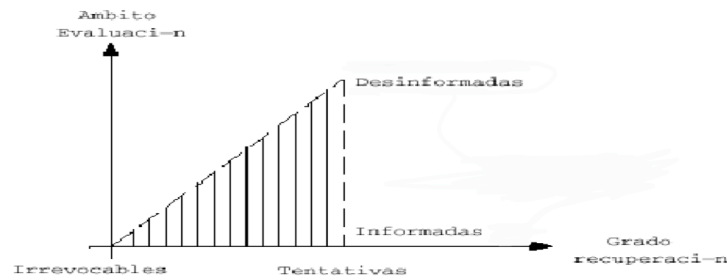
- ¿Podemos diseñar una función heurística de tal manera que el óptimo se queda fuera?
- ¿Cuáles son las condiciones que debe cumplir dicha función para que se garantice la optimalidad?

## 6. Esquema tipos básicos de búsqueda.



## 6. Comparativa tipos de búsqueda

- Comparación de las estrategias irrevocables, desinformadas e informadas
  - Los ejes de la gráfica mostrada en la figura tienen el siguiente significado:
    - **Alcance en la recuperación:** Grado en el que una estrategia permite recuperación de alternativas suspendidas previamente.
    - **Ámbito de evaluación:** Número de alternativas consideradas en cada decisión.



- *Ejercicio* - Situar en la gráfica anterior las siguientes estrategias de búsqueda:  
**Irrevocables, Tentativas Informadas, Tentativas Desinformadas**

## 7. Estrategias híbridas

- Combinación informada + backtracking



Otra posible combinación de estas dos estrategias puede ser realizar una búsqueda informada local dentro de una búsqueda backtracking global

- Combinación informada + irrevocable:

Búsqueda informada pero se eliminan aquellos nodos de la lista frontera que son menos prometedores





## 8.1 Búsqueda heurística. Conceptos básicos

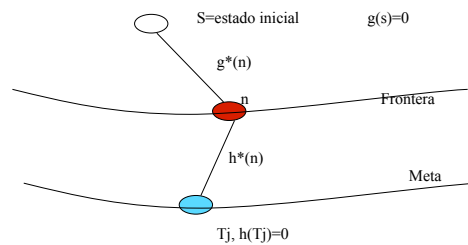
- **Complejidad** encuentra una solución si ésta existe
- **Admisibilidad** encuentra la solución óptima
- **Dominación**
  - Un algoritmo  $A_1$  es *dominante* sobre  $A_2$  si cada nodo expandido por  $A_1$  es también expandido por  $A_2$
- **Optimalidad**
  - Un algoritmo es el *óptimo* de un conjunto de algoritmos si es el dominante sobre todos los algoritmos del conjunto (es el que menos nodos expande)
- **La solución del problema vendrá dada por el camino de menor coste entre el estado inicial ( $s$ ) y cualquier estado objetivo ( $t_j$ )**

## 8.2 Búsqueda A\*. Búsqueda óptima

$$A^* : f^*(n) = g^*(n) + h^*(n)$$

- $g^*(n) = c(s, n)$ 
  - Coste del camino de **coste mínimo** desde el nodo inicial  $s$  al nodo  $n$ .
  - Estimada por  $g(n)$
- $h^*(n)$ 
  - Coste del camino de **coste mínimo** de todos los caminos desde el nodo  $n$  a cualquier estado solución  $t_j$
  - Estimada por  $h(n)$
- $f^*(n)$ 
  - Coste del camino de coste mínimo desde el nodo inicial hasta un nodo solución condicionado a pasar por  $n$
  - $f^*(n) = g^*(n) + h^*(n)$
  - Estimada por  $f(n)$
- $C^*$ 
  - Coste del camino de coste mínimo desde el nodo inicial a un nodo solución.

## 8.2 Búsqueda A\*. Búsqueda óptima



- $g(n) \geq g^*(n)$ ;  $g(n_j) = g(n_i) + c(n_i, n_j)$   $n_j$  es sucesor de  $n_i$
- Si tenemos una función  $h(n) = 0$  y el coste de cada regla es unitario,
- ¿Qué tipo de exploración realizaremos?
- $f^*(s) = h^*(s) = g^*(t_j) = f^*(t_j) = C^*$
- ¿ $f^*(n) = C^*$ ?

$$\forall t_j \in \Gamma$$