

Búsqueda Heurística I

Pedro Meseguer
IIIA-CSIC
Bellaterra, Spain
pedro@iiia.csic.es

Temario curso

- Introducción
- Búsqueda sistemática
 - Búsqueda ciega
 - Búsqueda informada: primero el mejor, A*
 - Búsqueda en memoria acotada
 - Búsqueda con adversario (juegos)
 - Búsqueda en tiempo real
- Búsqueda local
 - Vecindad
 - Función de evaluación
 - Criterios de salto

Parámetros

- Curso 22 horas
- Nivel: postgraduado
 - es autocontenido
 - se supone un cierto conocimiento del tema
- Evaluación: examen
 - de conceptos generales
 - no es muy duro, pero ...
 - ... se suspende
- Transparencias:
www.iiia.csic.es/~pedro/
- Clases :
 - jueves de 15 a 17, IIIA
- Calendario:
 - Octubre: 26
 - Noviembre: 2, 9, 16, 23, 30
 - Diciembre: 14, 21
 - Enero: 18
 - Febrero: 1, 8 (EXAMEN)

Búsqueda Heurística

a confirmar en Navidad

3

¿Qué es *búsqueda*?

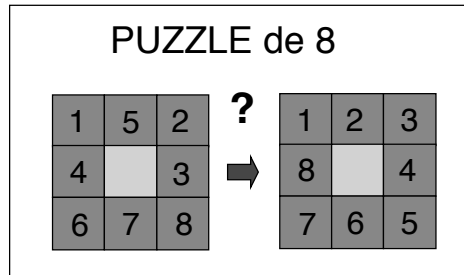
- Búsqueda: método computacional para resolver problemas
- ¿Qué problemas?
 - *single-agent path-finding*: único agente encuentra camino
 - *two-player games*: juegos de dos jugadores
 - *constraint satisfaction*: satisfacción de restricciones
- Características:
 - problemas difíciles: *NP-completos*
 - la solución se calcula por *enumeración*
 - se supone que *requieren* inteligencia → Inteligencia Artificial

Búsqueda Heurística

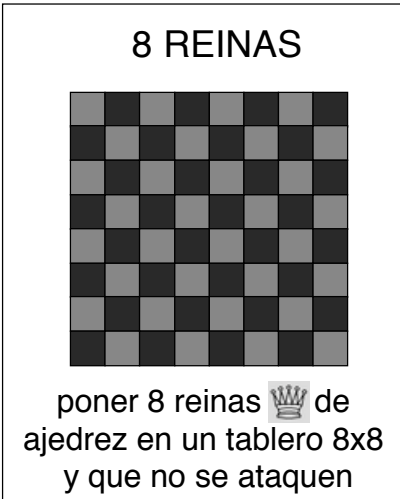
4

Ejemplos

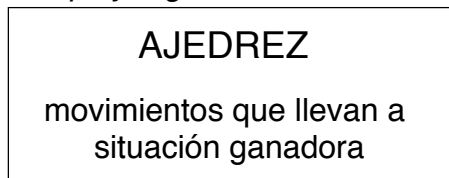
single-agent path-finding



constraint satisfaction



two-player games



Conceptos básicos

espacio del problema

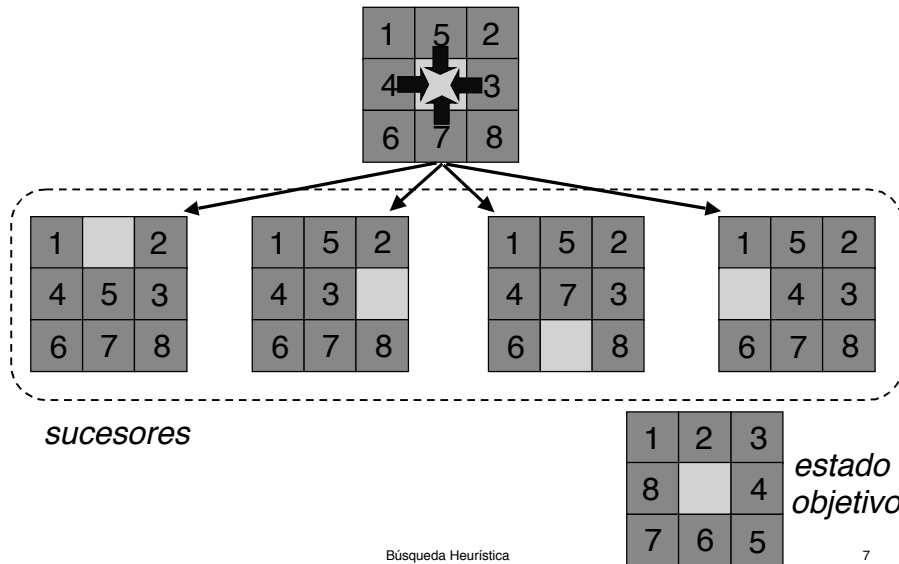
- Estado: una posible configuración de un problema
- Espacio de estados: todas las configuraciones
- Operadores:
 - acciones legales
 - generan estados *sucesores* de un estado

- Estados inicial y objetivo
explícito *puede ser implícito*

problema concreto

- Solución:
 - secuencia operadores de inicial a objetivo: *path-finding, games*
 - estado objetivo: *constraint satisfaction*

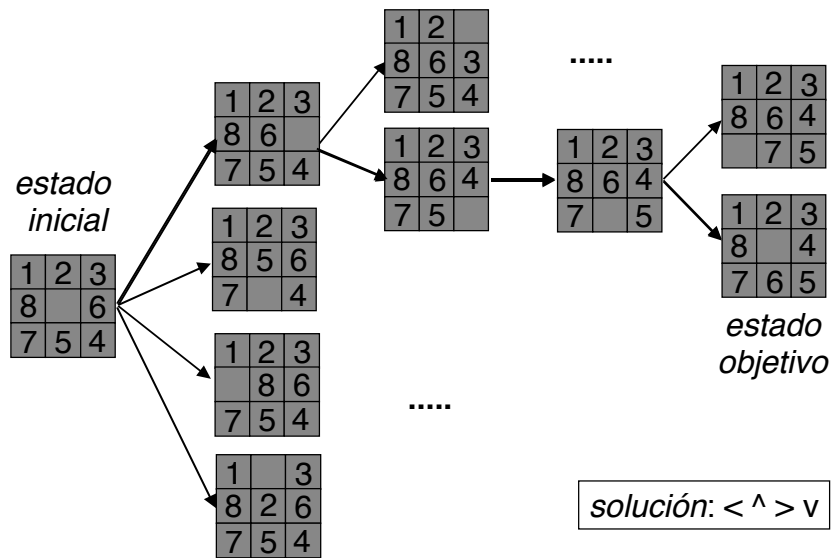
Ejemplo: puzzle de 8 (I)



Ejemplo: puzzle de 8 (II)

- Espacio de estados: posibles configuraciones
- Operadores:
 - arriba: ^
 - abajo: v
 - derecha: >
 - izquierda: <
- Estado inicial / objetivo: cualquiera / fijado
- Solución: secuencia mínima de movimientos para llegar al estado objetivo

Ejemplo: puzzle de 8 (III)



Búsqueda Heurística

9

Grafo dirigido

- Espacio de estados: grafo dirigido
- De estado inicial a objetivo: búsqueda en grafo
- ¿Por qué *NO* usar algoritmos de acceso en grafos?
- Porque el grafo no cabe en memoria: tamaños
 - puzzle-8: $9!$ damas: 10^{20} ajedrez: 10^{40}
- Hablamos de:
 - Grafo implícito: espacio de estados no desarrollado
 - Subgrafo explícito: subespacio desarrollado

Búsqueda Heurística

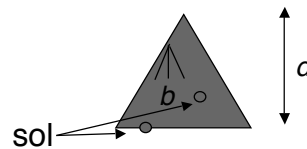
10

Sistemática y no sistemática

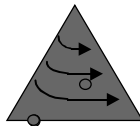
- **Sistemática:** si existe una solución la encuentra
 - recorrido ordenado del espacio de estados
 - si un estado no puede ser solución --> **poda**
- **No sistemática:** pueden no encontrar la solución
 - búsqueda local
 - descenso por gradiente
 - *simulated annealing*
 - algoritmos genéticos
 - búsqueda tabú
 - pueden visitar varias veces el mismo estado

Búsqueda ciega

- Búsqueda en árbol:

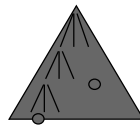


- Tres formas de generar y recorrer el árbol:



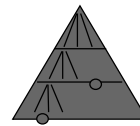
ANCHURA:
por niveles

- + sol más cercana raíz
- espacio exponencial



PROFUNDIDAD:
por ramas

- ¿sol cercana raíz?
- + espacio lineal



PROF. ITERATIVA: por ramas con nivel máximo

- + sol más cercana raíz
- + espacio lineal
- repite trabajo

Heurística



¡¡ EUREKA !!



Arquímedes en su baño

- Origen: *heuriskein* (encontrar en griego)
- Significado:
 - Procedimiento: criterio que puede resolver un problema pero que no hay garantía de que siempre lo resuelva.
 - Función: estimación del coste necesario para alcanzar una solución desde el estado actual.

Búsqueda Heurística

- Búsqueda *informada* con función heurística
- Heurística: función que asigna a cada estado una estimación del coste óptimo a la solución
- Compromiso: dos efectos
 - positivo: ahorra esfuerzo de búsqueda
 - negativo: coste de cálculo de la heurística en cada nodo
- efecto neto positivo si:

$\text{ahorro} > \text{coste}$

Ejemplo: puzzle de 8

estado actual

1	5	2
4		3
6	7	8

1	2	3
8		4
7	6	5

estado objetivo

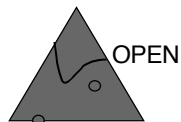
Heurísticas:

piezas fuera de sitio = 7

Σ distancia Manhattan = 3 + 1 + 1 + 3 + 1 + 1 + 2 = 12
(número pasos en cuadrícula)

Búsqueda informada: primero el mejor

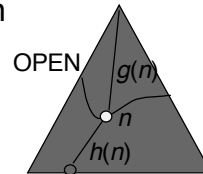
- Función $f(n)$: coste del nodo n
- Lista OPEN: nodos generados pero no testeados
- Primero el mejor: siguiente n con $\min f(x) \ x \in \text{OPEN}$



- Rendimiento: mucho mejor que la búsqueda ciega

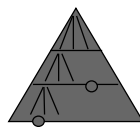
Algoritmo A*

- Heurística: $h(n)$ estima el coste de un camino óptimo de n a una solución
- $g(n)$: coste desde la raíz hasta n
- $h(n)$ admisible: si no sobreestima el coste óptimo
- A*: primero el mejor con $f(n) = g(n) + h(n)$, h admisible
 - + completo
 - + solución óptima
 - espacio: a menudo degenera en exponencial



Búsqueda heurística con memoria acotada

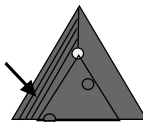
- Profundización iterativa A*:



+ $h(n)$ admisible

- *Branch and bound* (DFBnB)

UB = mejor $f(x)$ en



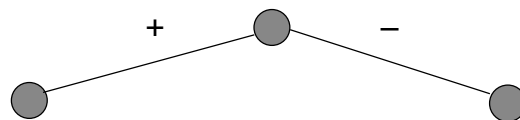
$g(n) + h(n) = LB$

si $UB \leq LB$, poda

+ ambos requieren memoria lineal
- se cambia espacio por tiempo

Búsqueda con discrepancias

Árbol de búsqueda + heurística: ordena los sucesores, izda. preferido a der.

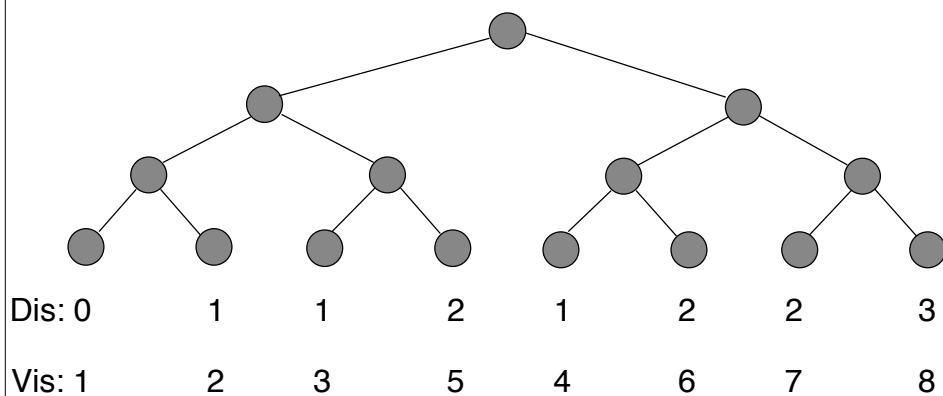


Discrepancia: ir en contra de la heurística

IDEA1: si la heurística casi siempre acierta, la solución estará en una rama con pocas discrepancias

Búsqueda con discrepancias

IDEA2: recorrer el árbol por orden creciente de discrepancias



Juegos

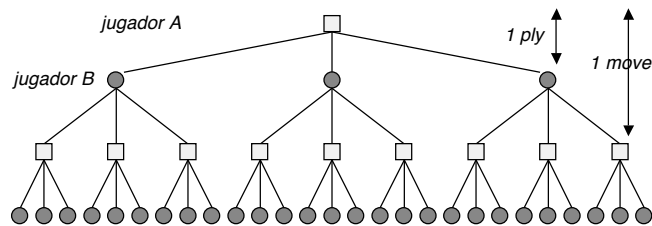
- 2 jugadores, información perfecta
- Programas campeones del mundo: damas, othello
- Programa con muy buen rendimiento: ajedrez
 - importante: desde 1956, ajedrez era un objetivo para IA
 - *Deep Blue* gana a Kasparov en 1997
 - “*quantity had become quality*”
- Estrategia:
 - búsqueda en un espacio de billones de nodos
 - profundización selectiva

Búsqueda Heurística

21

Búsqueda para juegos

- Árbol de juegos:



- Algoritmo minimax, asciende los valores de nodos terminales
- Mejora alfa-beta (ordenación nodos)
- Función evaluación, quiescencia, extensiones singulares

Búsqueda Heurística

22

Búsqueda en tiempo real

- Búsqueda clásica (*off-line search*):

Búsqueda de una solución completa	Ejecución de la solución
-----------------------------------	--------------------------

- No sirve para tareas con tiempo o información limitada
- IDEA: alternar búsqueda con ejecución (*on-line search*)

Búsqueda	Ejecución	Búsqueda	Ejecución	Búsqueda	Ejecución
----------	-----------	----------	-----------	----------	-----------

- Algoritmos: RTA*, LRTA*:
 - primera ejecución
 - al repetir la misma tarea: mejorar rendimiento
(óptimo rendimiento en el límite)

Aplicaciones de búsqueda en tiempo real

Age of Empires



Warcraft III



Agentes:

tiempo e información limitada
han de mejorar el rendimiento con la experiencia

Resumen

- Búsqueda: método computacional para resolver problemas
- ¿Qué problemas? Resolución por enumeración
- Espacio de estados, operadores, inicial, objetivo
- Búsqueda ciega
- Heurística: función que estima el coste al objetivo
- Búsqueda informada, A*
- Búsqueda en memoria acotada
- Búsqueda con discrepancias
- Búsqueda para juegos
- Búsqueda en tiempo real

¿Por qué estudiar búsqueda ...

- ...cuando hay otros temas más *de moda*?
- Varias razones:
 - muchos problemas no se pueden resolver de otra forma
 - área de investigación viva:
 - ver IJCAIs, AAAIs, ECAIs
 - ideas nuevas: discrepancias, *pattern databases*, *real time*, ...
 - proporciona un marco de referencia:
 - muchos problemas de IA se pueden conceptualizar como búsqueda
 - aunque se resuelvan en la práctica por otro método
 - explora ideas que se pueden exportar a otros campos