

Búsqueda Heurística IV

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

Búsqueda para juegos

- 2 jugadores
- Perfecta información:
 - cada jugador conoce *toda* la información del contrario
 - no hay elementos aleatorios
- Ejemplos: ajedrez, damas, othello
- Juegos con información incompleta: poker, bridge

Programas para juegos

- 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”
- Importancia económica directa
- Estrategia:
 - búsqueda en un espacio de billones de nodos
 - profundización selectiva

Búsqueda Heurística

3

Othello

Fichas con dos caras: blanca y negra

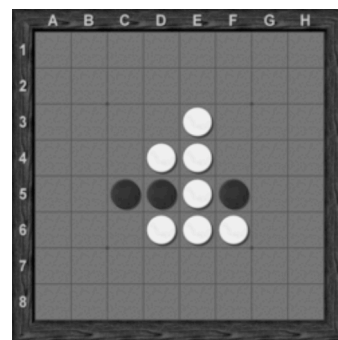
2 jugadores: blanco y negro

Mover: nueva ficha en posición en línea recta con otra ficha del mismo color que entremedio tenga fichas de color opuesto

Las fichas “entremedio” cambian de color (y por tanto de jugador)

Turnos alternativos

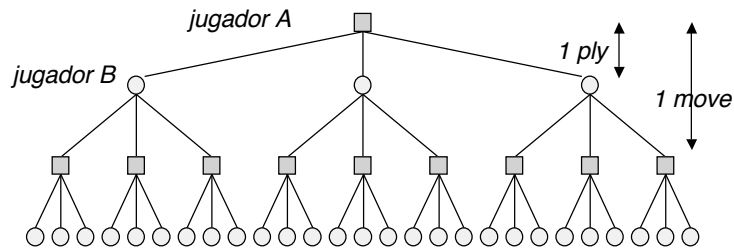
Ganar: más fichas de un color cuando se llena el tablero o cuando no hay movimiento posible



Búsqueda Heurística

4

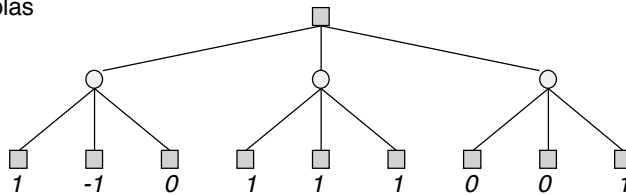
Árbol de juegos



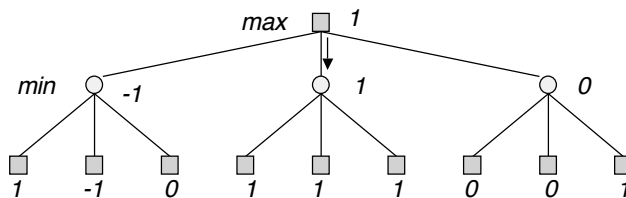
- Alternan jugadores por niveles
- Sucesores de un nodo: todas los movimientos legales que ese jugador puede hacer

MiniMax

1 gana ◻
 -1 gana ○
 0 tablas



Cuál es el mejor movimiento?
 Supongo que el oponente elige el movimiento que más le conviene.



MiniMax

- Idea: propagar hacia atrás el valor de los hijos
- Nodos terminales: valor 1, -1 (gana A o gana B)
- Dos tipos de nodos:
 - max**: gana con 1
 - min**: gana con -1
- Propagación hacia atrás:
 - **max**: el máximo de los valores de los hijos
 - **min**: el mínimo de los valores de los hijos

Algoritmo MiniMax

Algoritmo MiniMax: evaluar nodo n de un árbol de juegos

1. Expandir el árbol bajo n
2. Evaluar los nodos terminales.
3. Seleccionar un nodo sin valor que todos sus hijos han sido valuados. Si no existe, retornar el valor de n . Stop.
4. Si el nodo es max, asignar el máximo de sus hijos.
Si el nodo es min asignar el mínimo de sus hijos. Ir a 3.

Pregunta

El algoritmo anterior tiene un problema:

¿cuál?

Supone búsqueda en anchura:

4. Si el nodo es max, asignar el **máximo de sus hijos**.
Si el nodo es min asignar el **mínimo de sus hijos**.

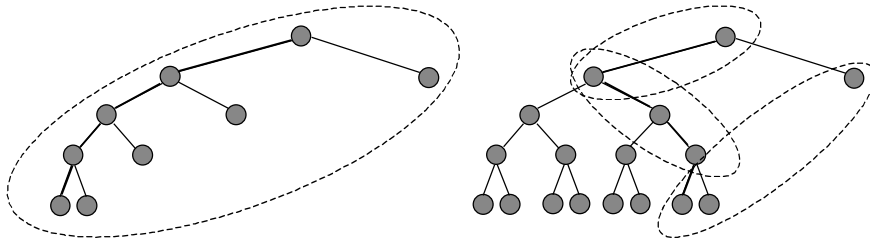
espacio exponencial 

MiniMax en DFS

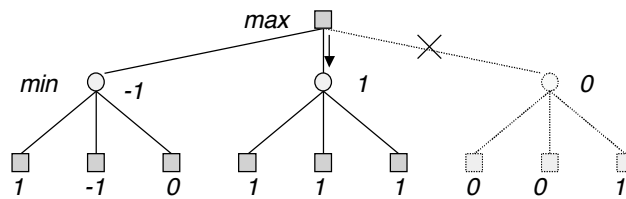
1. $L = \{n\}$
2. $x \leftarrow \text{extrae-primero}(L)$.
Si $x = n$ con valor asignado, retorna ese valor. Stop.
3. Si x tiene valor w , nodo p es padre de x con valor v
Si p es max, asignar $\max(v, w)$ a p .
Si p es min, asignar $\min(v, w)$ a p .
Eliminar x de L , ir a 2.
4. Si x no tiene valor y es un nodo terminal,
asignar $+1$ o -1 como valor, depende si gana para max o min
5. Si x no tiene valor y no es un nodo terminal,
asignar $-\infty$ o $+\infty$ como valor, depende si es max o min

MiniMax en DFS

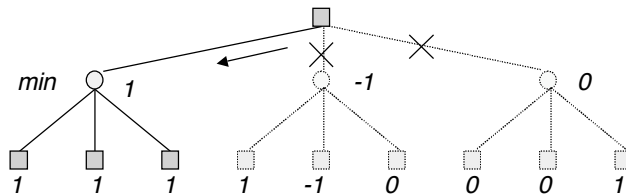
- Minimax ha de guardar:
 - sucesores de rama actual (como DFS)
 - nodos en la rama actual: para actualizar su valor



Alfa-Beta



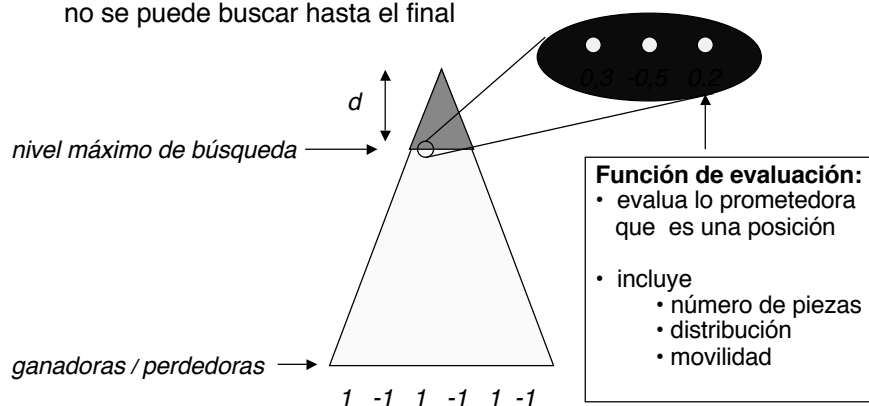
Con una buena ordenación de sucesores,



Poda muy importante: $b \approx 36 \rightarrow b \approx 6$

Función de evaluación

Árbol de juegos:
demasiado grande
no se puede buscar hasta el final



Función de evaluación para ajedrez

$$e(n) \rightarrow [-1, +1]$$

- sustituye la detección de nodos terminales
- es imperfecta

Ajedrez: sumar los valores de las piezas

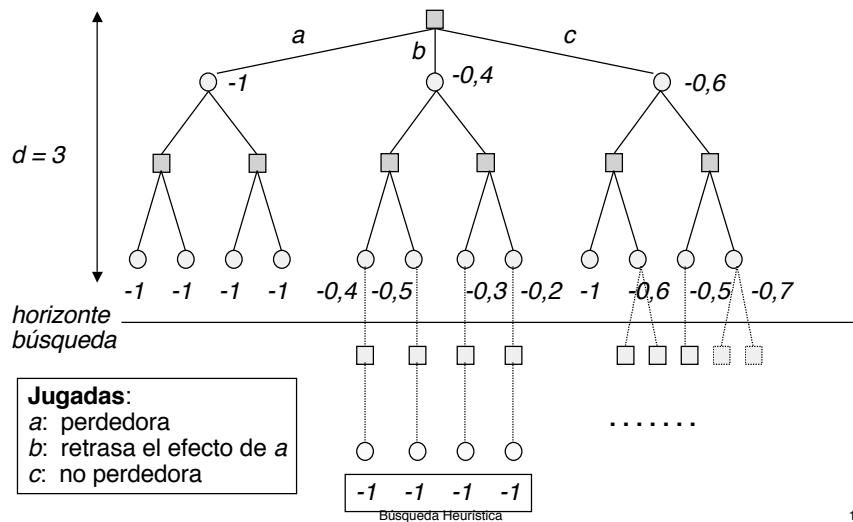
peón, caballo, alfil, torre, reina

1 3 3 5 10

B: suma de las blancas N: suma de las negras

$$e(n) = B - N / B + N$$

Efecto horizonte



15

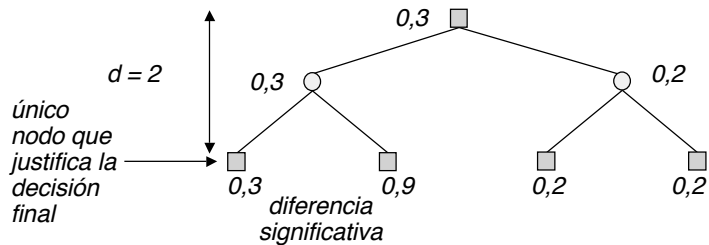
Efecto horizonte en ajedrez

- Efecto horizonte: una jugada que inicialmente parece buena puede resultar mala
 - el peligro no se ve, está tras el horizonte
 - la función de evaluación es incapaz de detectarlo
- Solución fuerza bruta: buscar más profundo
 - coste computacional
 - en ajedrez, hay muchos movimientos que pueden retrasar una jugada de peligro
- Aproximación selectiva: identificar los nodos en los que se debe profundizar más → extensiones singulares

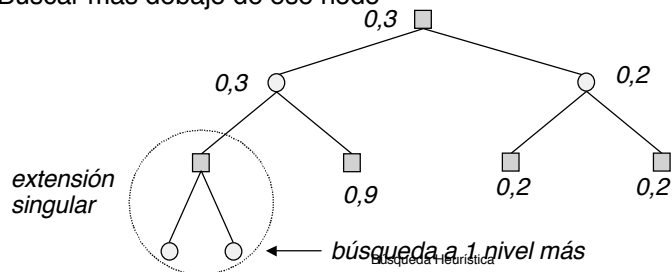
Búsqueda Heurística

16

Extensiones singulares



Buscar más debajo de ese nodo



17

Deep Blue

- Proyecto de IBM
- Alfa-beta paralelo:
 - *Hardware* especializado (2 millones posiciones / sg)
 - Extensiones singulares, quiescencia
 - Función de evaluación sofisticada
 - Biblioteca: aperturas y finales
- Resultados:
 - Venció a Kasparov en 1997
 - Ratio 2700, Kasparov 2800
 - *quantity had become quality*

Búsqueda Heurística

18

Algunas especulaciones

¿Por qué los programas son buenos en ciertos juegos y en otros no?

Hipótesis:

- Programa = búsqueda
- Rendimiento frente a humanos: aumenta con profundidad d
- Número de estados a visitar b^d en tiempo fijado

Depende de b factor de ramificación:

- Si b pequeño, se puede profundizar → buen rendimiento
- Si b grande, no se puede profundizar → rendimiento pobre

Algunas especulaciones

Factor de ramificación:

- damas: 4
 - othello: < 10
 - ajedrez: 36
 - go: 361
- } *el campeón mundial es un programa*

Otras ideas:

- aprendizaje (backgammon)
- dificultad cuando se intercambian muchas piezas

Resumen

- Juegos: objetivo de IA desde hace mucho tiempo
- 2 jugadores, perfecta información
- Árbol de juegos
- MiniMax: propaga valores de nodos terminales a internos
- Alfa-beta: genera poda
- Árbol demasiado grande: no se llega a nodos terminales
- Función de evaluación: sobre nodos a profundidad d
- Extensiones singulares
- Especulación: factor de ramificación