

Búsqueda

Grupo de Planificación y Aprendizaje (PLG)
Departamento de Informática
Escuela Politécnica Superior
Universidad Carlos III de Madrid

22 de diciembre de 2008

Búsqueda de dos agentes

Búsqueda

Grupo de Planificación y Aprendizaje (PLG)
Departamento de Informática
Escuela Politécnica Superior
Universidad Carlos III de Madrid

22 de diciembre de 2008

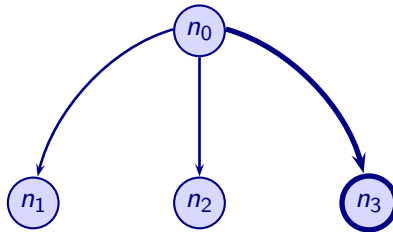
En Esta Sección:

- 1 Búsqueda de dos agentes
 - Caracterización del problema
 - Algoritmo Minimax
 - Algoritmo Alfa-Beta
 - Otras técnicas

Caracterización del problema

- **Suma nula:** lo que gana uno, lo pierde el otro
- **Dos agentes** (se puede generalizar, Max^n)
- **Información completa:** se conoce en cada momento el estado completo del juego
- **Deterministas** o de **información perfecta:** no entra en juego el azar (se puede generalizar)
- **Alternados:** las decisiones de cada agente se toman de forma alternada

Resolución con función de evaluación perfecta



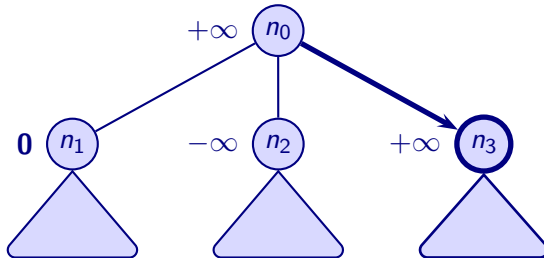
$$f^*(n) =$$

2**-3****8**

Problema

Normalmente, no se conoce dicha función

Resolución con búsqueda completa



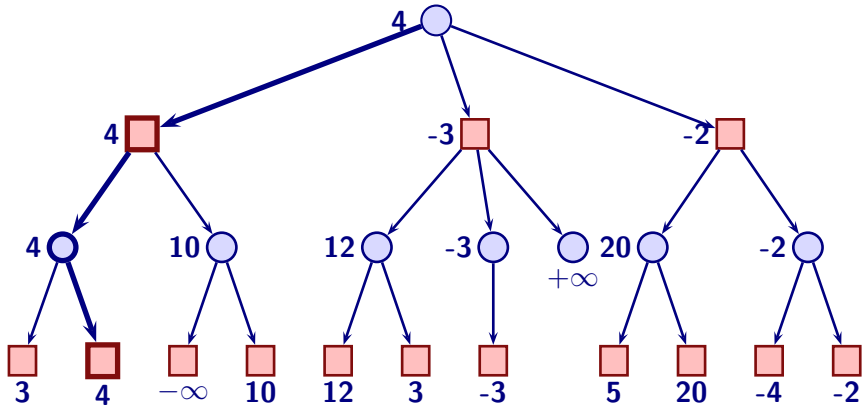
Observaciones

- Nodos hoja: ganar (∞), perder ($-\infty$) o empatar (0)
- **Problema: es intratable; no se puede realizar en un tiempo razonable**

Algoritmo Minimax

v_i ○ Nodos MAX (juega el Minimax)

v_i □ Nodos MIN (juega el oponente)



Cálculo del valor de cada nodo

Minimax

$$f(n) = \begin{cases} +\infty & \text{si } n \text{ es una situación ganadora} \\ -\infty & \text{si } n \text{ es una situación perdedora} \\ 0 & \text{si } n \text{ es una situación de empate} \\ f_{ev}(n) & \text{si } p = \text{Profundidad-máxima} \\ \max_{S_i \in S(n)} f(S_i) & \text{si } n \text{ es nodo MAX y } p < p_{max} \\ \min_{S_i \in S(n)} f(S_i) & \text{si } n \text{ es nodo MIN y } p < p_{max} \end{cases}$$

Negamax

$$f(n) = \begin{cases} f_{ev}(n) & \text{si } d = 0 \\ \max(-f(S_1), \dots, -f(S_k)) & \text{si } d > 0 \end{cases}$$

Algoritmo minimax

Procedimiento Minimax (Situación Profundidad)

SI Profundidad = p_{max}

ENTONCES devolver evaluación (Situación)

SI NO SI ganadora (Situación)

ENTONCES devolver $+\infty$

SI NO SI perdedora (Situación)

ENTONCES devolver $-\infty$

SI NO SI empate (Situación)

ENTONCES devolver 0

SI NO

S = sucesores (Situación)

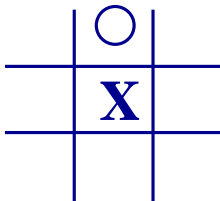
L = lista de llamadas al MINIMAX ($S_i \in S$, Profundidad + 1)

SI nivel-max (Profundidad)

ENTONCES devolver max (L)

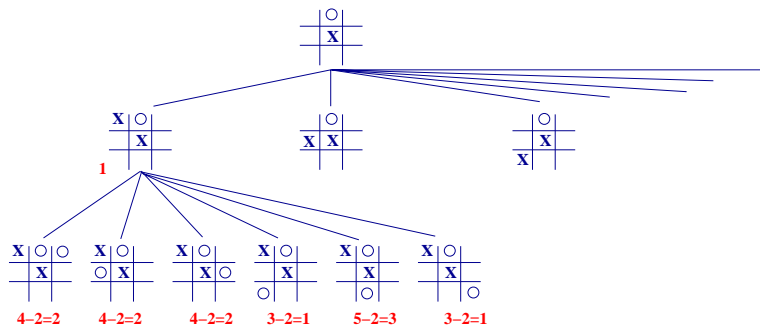
SI NO devolver min (L)

Tic-tac-toe – Minimax

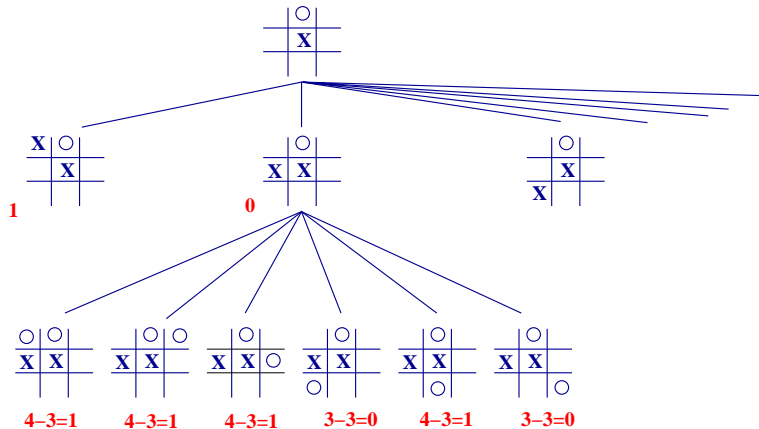


- Función de evaluación $f(n)$: número de posibilidades de hacer tres en raya del jugador menos número de posibilidades de hacer tres en raya del oponente
- Si colocamos la primera ficha en:
 - el centro: 4 posibilidades para hacer 3 en raya ($f(n) = 4 - 4 = 0$)
 - en una esquina: 3 posibilidades ($f(n) = 3 - 5 = -2$)
 - en el centro de un lateral: 2 posibilidades ($f(n) = 2 - 6 = -4$)

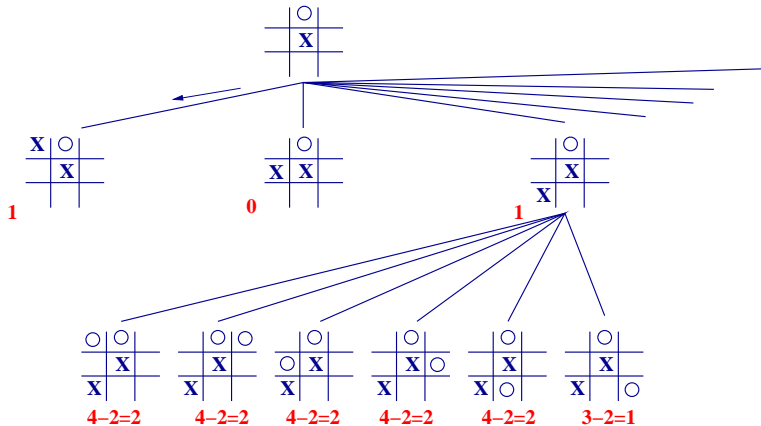
Tic-tac-toe – Minimax



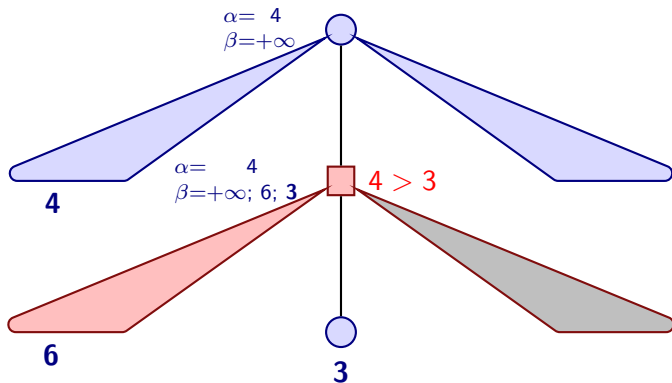
Tic-tac-toe – Minimax



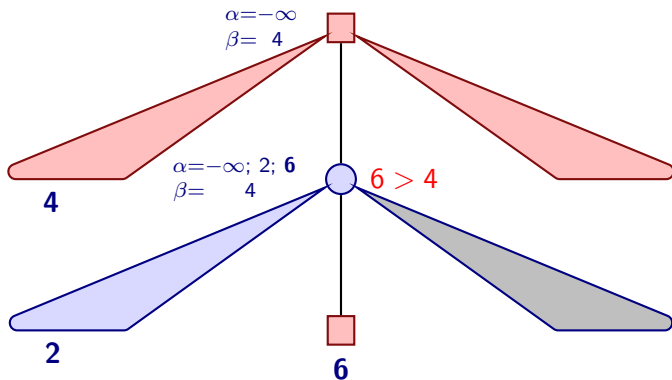
Tic-tac-toe – Minimax



Poda en los nodos MAX



Poda en los nodos MIN



Poda Alfa-Beta

- Nodos MAX: se utiliza el parámetro α para guardar el valor máximo de los sucesores encontrado hasta el momento.

$$\alpha_0 = -\infty$$

- Nodos MIN: se utiliza el parámetro β para guardar el valor mínimo de los sucesores encontrado hasta el momento.

$$\beta_0 = +\infty$$

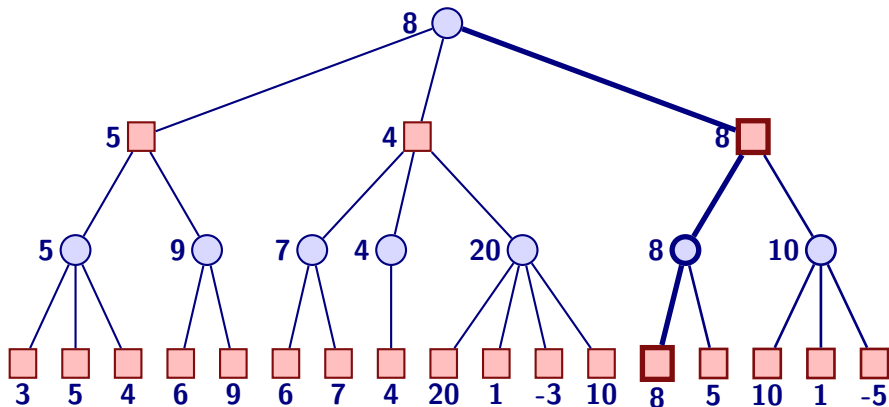
- Poda en los nodos MAX:

$$\alpha_p \geq \beta_{p-1}$$

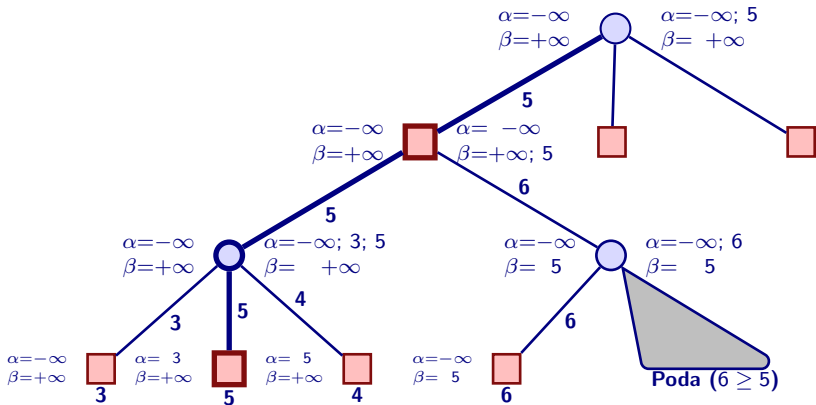
- Poda en los nodos MIN:

$$\beta_p \leq \alpha_{p-1}$$

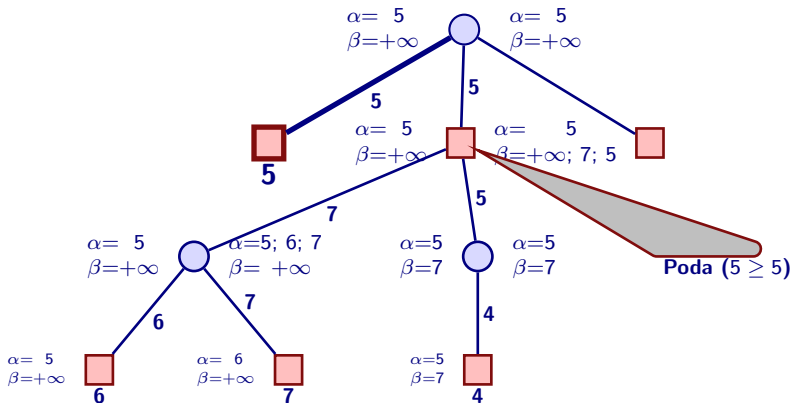
Alfa-Beta: Ejemplo



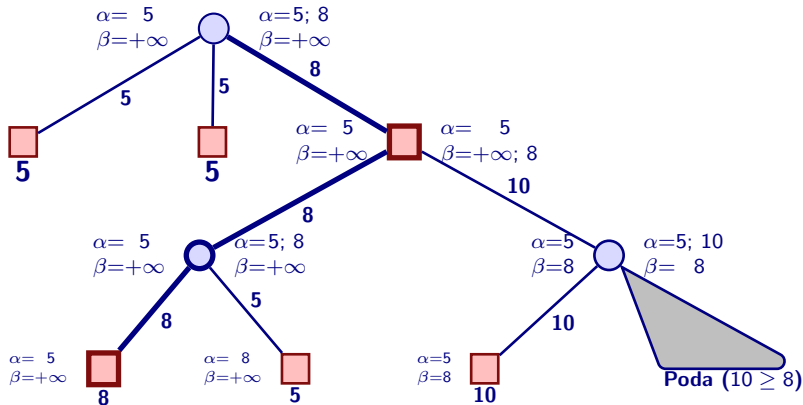
Alfa-Beta: Ejemplo



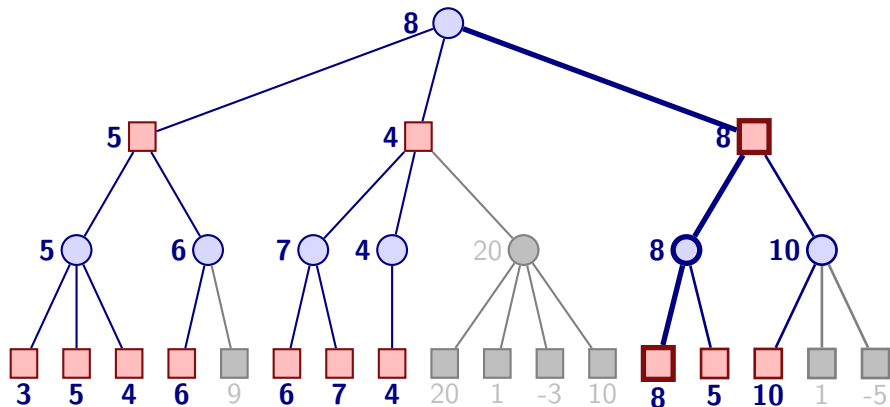
Alfa-Beta: Ejemplo



Alfa-Beta: Ejemplo



Alfa-Beta: Ejemplo



Algunos aspectos relacionados

- **Efecto horizonte:** se produce al fijar una profundidad máxima (horizonte)
- **Ordenación de nodos:** se ordenan los nodos hasta profundidad k de acuerdo a una función de evaluación más sencilla f'
- **Ventana inicial:** se puede comenzar por valores de alfa y beta diferentes del mínimo y máximo valor, respectivamente
- **Profundización iterativa**
- **Movimiento nulo:** suponer que no se realiza ninguna acción es lo peor que se puede hacer en algunos casos (posiciones “zugzwang”)
- **Movimiento asesino:** guardar estados y valores
- **Tablas de transposición:** para evitar la re-expansión de nodos

Algunos aspectos relacionados

- **Búsqueda secundaria:** realizar búsquedas parciales en los nodos en los que se detecten problemas
- **Extensiones selectivas:** donde se realiza la búsqueda secundaria
- **Cálculo del valor de cada nodo:** se puede utilizar la media, la suma ponderada, tener en cuenta las probabilidades, ...

Estado de la cuestión en juegos de dos jugadores

- Ajedrez: HiTech, Deep Thought-Blue, Fritz
- Damas: Chinook
- Othello: Logistello
- Backgammon: NeuroGammon
- Juegos resueltos: Tic-Tac-Toe, Cuatro en Raya, Qubic, Go-Moku, **Damas (oficialmente desde 2007)**, ...
- IA en juegos actuales: estrategia (ORTS), independencia de dominios (GGP), ...

Otras técnicas

- Otras técnicas equivalentes de suma nula: SSS^* , B^* , números conspiratorios, de prueba, y otros
- No-deterministas: \ast -Minimax
- Más de dos oponentes: Max^n
 - En lugar de mantener un valor (alfa o beta) mantienen n valores en un vector
 - Sólo es aplicable si la suma de todas las puntuaciones es un valor constante