

Importante

Puedo eliminar la publi de este documento con 1 coin

¿Cómo consigo coins? → Plan Turbo: barato
→ Planes pro: más coins

pierdo espacio



Necesito concentración

ali ali oohh
esto con 1 coin me
lo quito yo...

WUOLAH

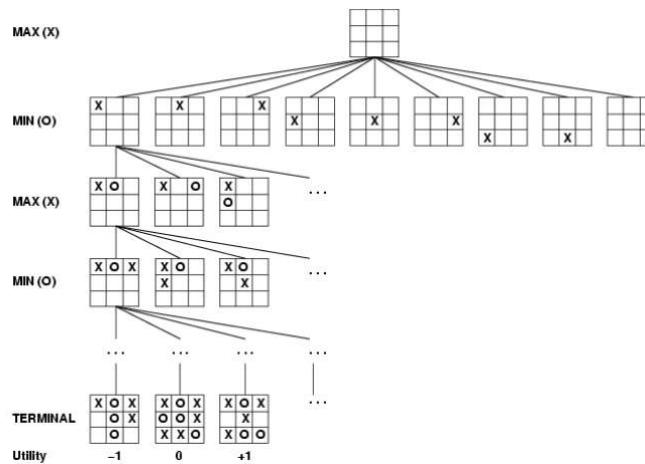
Tema 4. Búsqueda con adversarios: juegos

Árboles de exploración de juegos

Un árbol del juego es una representación explícita de todas las formas de jugar a un juego.

Notación min-max

- **MAX**: primer jugador
- **MIN**: segundo jugador
- Nodos MAX y nodos MIN
- Los nodos terminales se etiquetan con V, D o E desde el punto de vista de MAX



Algoritmo STATUS

- Si J es un nodo MAX no terminal, entonces STATUS(J)=
 - V si alguno de los sucesores de J tiene STATUS V
 - D si todos los sucesores de J tienen STATUS D
 - E en otro caso
- Si J es un nodo MIN no terminal, entonces STATUS(J)=
 - V si todos los sucesores de J tienen STATUS V
 - D si alguno de los sucesores de J tiene STATUS D
 - E en otro caso

Nuevo modelo de solución

Los juegos complejos no se pueden resolver ya que es imposible calcular todos los casos posibles. Por lo que el objetivo será encontrar una buena jugada. En este proceso, la heurística será muy importante.

WUOLAH

El modelo básico

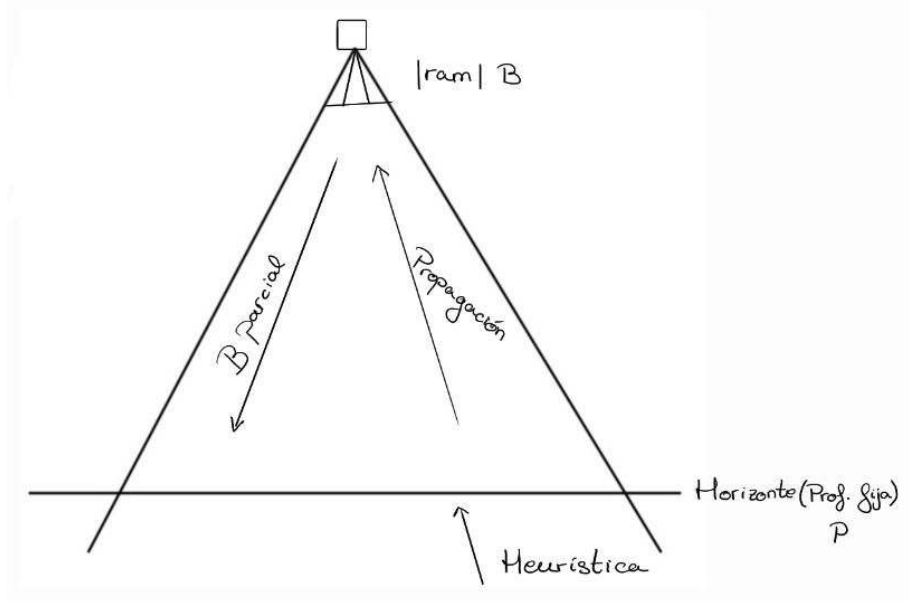
- Arquitectura percepción/planificación/actuación
- Búsqueda con horizonte
- Uso de heurísticas

En distintos juegos necesitamos anticipar movimientos futuros para tomar buenas decisiones. Elementos claves para ello:

- **Horizonte: profundidad.** Se refiere a cuántos movimientos hacia adelante va a analizar el algoritmo antes de tomar una decisión.
- **Heurística.** Como no se puede llegar hasta el final del juego (por tiempo o recursos), se usa una función heurística para estimar qué tan buena es una posición en el juego. Por ejemplo, en el ajedrez, +1 por cada peón tuyo, -1 por cada del rival, etc.
- **Búsqueda parcial.** Solo se exploran ciertas ramas del árbol de juego hasta cierta profundidad. No todo el árbol se genera completamente.
- **Propagación.** Una vez que se evalúan las posiciones con la heurística, se propagan los valores hacia arriba del árbol usando estrategias como Minimax.

Fórmula de complejidad:

- **B** es el **factor de ramificación**: cuántas opciones tiene cada jugador en promedio por turno.
- **P** es la **profundidad** de búsqueda.
- Entonces B^P nos da cuántos nodos del árbol se generarían (en el peor de los casos). En otras palabras. Complejidad de alg = n° de llamadas a la heurística



Minimax

La regla minimax

- El valor $V(J)$ de un nodo J de la frontera de búsqueda es su valor estático.
- Si J es un nodo MAX, entonces su valor $V(J)$ es máximo de los valores de sus nodos sucesores
- Si J es un nodo MIN, entonces su valor $V(J)$ es el mínimo de los valores de sus nodos sucesores.

Algoritmo Minimax

Para determinar el valor minimax, $V(J)$ de un nodo J , hacer lo siguiente:

- Si J es un nodo terminal, devolver $V(J)=f(J)$; en otro caso
- Para $k=1,2,\dots,b$, hacer:
 - Generar J_k , el k -ésimo sucesor de J
 - Calcular $V(J_k)$
 - Si $k=1$, hacer $AV(J) \leftarrow V(J_1)$; en otro caso, para $k \geq 2$,
 - hacer $AV(J) \leftarrow \max\{AV(J), V(J_k)\}$ si J es un nodo MAX o
 - hacer $AV(J) \leftarrow \min\{AV(J), V(J_k)\}$ si J es un nodo MIN
- Devolver $V(J)=AV(J)$

Poda ALFA-BETA

Cotas alfa y beta

	Para nodos	Se calcula	es
Cota alfa	Nodos MIN	Máximo de los nodos MAX	Cota inferior
Cota beta	Nodos MAX	Mínimo de los nodos MIN	Cota superior

Algoritmo ALFA-BETA

Para calcular el valor $V(J, \alpha, \beta)$, hacer lo siguiente:

1. Si J es un nodo terminal, devolver $V(J)=f(J)$. En otro caso, sean $J_1, \dots, J_k, \dots, J_b$ los sucesores de J . Hacer $k \leftarrow 1$ y, si J es un nodo MAX ir al paso 2; si J es un nodo MIN ir al paso 5.
2. Hacer $\alpha \leftarrow \max(\alpha, V(J_k, \alpha, \beta))$.
3. Si $\alpha \geq \beta$ devolver β ; si no, continuar
4. Si $k=b$, devolver α ; si no, hacer $k \leftarrow k+1$ y volver al paso 2.
5. Hacer $\beta \leftarrow \min(\beta, V(J_k, \alpha, \beta))$.
6. Si $\beta \leq \alpha$ devolver α ; si no, continuar
7. Si $k=b$, devolver β ; si no, hacer $k \leftarrow k+1$ y volver al paso 5.

Complejidad

- Peor caso $\rightarrow B^P$
- Mejor caso $\rightarrow B^{(P/2)}$

Importante

Puedo eliminar la publi de este documento con 1 coin

¿Cómo consigo coins? → Plan Turbo: barato
→ Planes pro: más coins

pierdo
espacio



Necesito
concentración

ali ali ooh
esto con 1 coin me
lo quito yo...

WUOLAH

Heurísticas para la búsqueda en árboles de juegos

Componentes del diseño de la búsqueda que influyen al rendimiento de la búsqueda.

- **Evaluación hacia atrás.** Propaga valores desde las hojas del árbol hacia la raíz usando Minimax.
- **Profundidad de la búsqueda.** Limita cuántos turnos hacia adelante analiza el algoritmo.
- **Ordenación de la búsqueda.** Prioriza los movimientos más prometedores para mejorar la eficiencia. Esto para el algoritmo Minimax con poda alfa-beta hace que sea mucho más eficiente ya que se poda mucho más.
- **Anchura de la búsqueda.** Describe cuántas jugadas se consideran por turno (factor de ramificación). Cuanto mayor es la anchura, más lenta se vuelve la búsqueda.

WUOLAH