

Tema 4. Búsqueda con adversario: juegos.

Interés.

- Laboratorios perfectos para investigar en técnicas de resolución de problemas.
- Es fácil medir el éxito o el fracaso.
- Fascinación para cierta gente.
- Aspecto comercial.
- Aplicaciones en ámbitos empresariales.

Juegos bipersonales con información perfecta.

Estas situaciones se estudian y resuelven utilizando la Teoría de Juegos.

¿Qué es un juego? Es cualquier situación de decisión, caracterizada por poseer una interdependencia estratégica, gobernada por un conjunto de reglas y con un resultado bien definido.

En un juego, cada jugador intenta conseguir el mayor beneficio para sus intereses. La solución de un juego permite indicar a cada jugador qué resultado puede esperar y cómo alcanzarlo.

Ejemplo de juego: el dilema del prisionero.

Ejemplo de juego: el juego de los palillos (cada jugador puede quitar 1, 2 o 3 palos).

Un juego de información perfecta es aquel en el que los jugadores tienen a su disposición toda la información de la situación del juego.

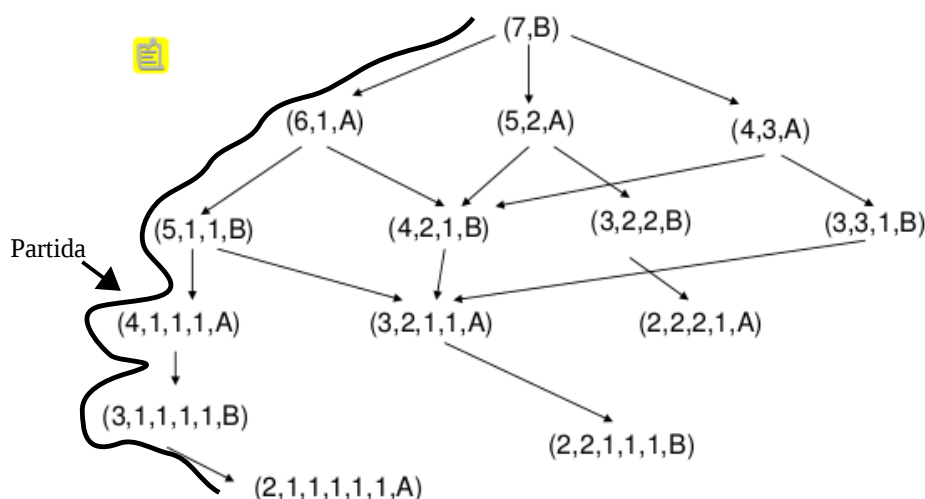
Árboles de exploración de juegos.

Agente deliberativo que tiene delante a otro jugador.

Un árbol del juego es una representación explícita de todas las formas de jugar a un juego. El árbol consiste en las opciones que tiene cada jugador. Cada camino es una PARTIDA.

Correspondencia entre árboles de juegos y árboles Y/O. Cuando no controlo algo, es un árbol Y/O.

Ejemplo simple.



Vas dividiendo un montón de monedas, y tienes que hacer divisiones de ellas.

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.



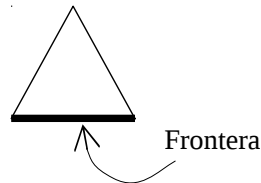
- ✓ ESCUELAS 40 ALREDEDOR DEL MUNDO
- ✓ TODOS LOS NIVELES Y OBJETIVOS
- ✓ AÑOS DE 80 EXPERIENCIA
- ✓ 97% DE RECOMENDACIÓN

Nuevo modelo de solución.

Nos fijamos en como juega un ser humano. Los juegos complejos no se pueden resolver ya que es imposible la exploración total hasta la terminación.

Nuevo objetivo: encontrar una buena jugada inmediata. La heurística en el proceso es muy importante.

Búsqueda parcial: ve lo que hay en la frontera. En la frontera usamos heurísticas. Propagan los valores de la heurística hacia arriba mediante métodos de propagación hacia atrás.



Ejemplo:

A.Samuel: las damas. Tiene valores que ejemplifican la posición de las piezas en la tabla $(x_1 \dots x_n)$. Heurística $\lambda_1 * x_1 + \dots + \lambda_n * x_n$ $\lambda_i = (1/n)$. El jugador automático jugará muy bien.

El modelo básico.

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

La regla minimax.

El valor $V(J)$ de un nodo J de la frontera de búsqueda es igual al de su evaluación estática; en otro caso:

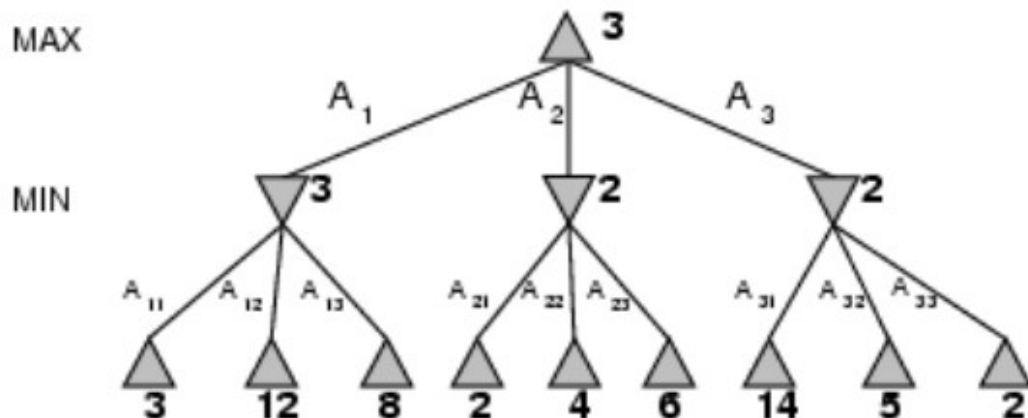
- Si J es un nodo **MAX**, entonces su valor $V(J)$ es igual al máximo de los valores de sus nodos sucesores.
- Si J es un nodo **MIN**, entonces su valor $V(J)$ es igual al mínimo de los valores de sus nodos sucesores.

Algoritmo minimax.

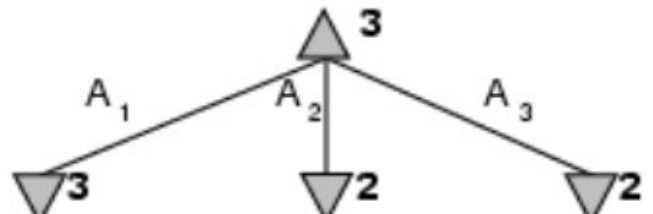
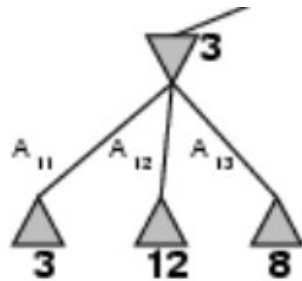
Para determinar el valor minimax, $V(J)$ de un nodo J , hacer lo siguiente:

- 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 $V(J) = AV(J)$.

Ejemplo.



Baja a los nodos hoja de cada nodo. En este ejemplo, busca el menor valor que hay en sus hijos. El 3.



En este caso, busca el máximo de los hijos, que es 3.

Algoritmo ALFA-BETA.

Cota alfa->

- Uso: MIN.
- Calcular: máximo de los nodos MAX que estan por debajo.
- Como se usa -> cota inferior.

Cota beta ->

- Uso: MAX.
- Calcular: mínimo de los nodos MIN que están por encima.
- Como se usa -> cota superior.