Tema 4 - Búsqueda con adversario: juegos

- ▼ Juegos bipersonales con información perfecta
  - ▼ Características
    - Ambos jugadores conocen el estado completo del juego
    - No hay azar ni información oculta
    - Se resuelven usando árboles de exploración y teoría de juegos
  - ▼ Ejemplos
    - Tres en raya
    - Damas
    - Ajedrez
- ▼ Árboles de exploración de juegos
  - Representación explícita de todas las formas de jugar a un juego
  - Correspondencia entre árboles de juegos y árboles Y/O
  - ▼ Notación min-max

	MAX	MIN
Jugador	1	2
Objetivo	Maximizar puntuación	Minimizar puntuación rival

    - ▼ Nodos terminales (desde punto de vista de MAX)
      - V → victoria
      - D → derrota
      - E → empate
  - ▼ Algoritmo STATUS
    - ▼ Valor de STATUS(J) en un nodo MAX no terminal
      - V → alguno de sus sucesores tiene STATUS V
      - D → todos de sus sucesores tiene STATUS D

- E → cualquier otro caso

▼ Valor de STATUS(J) en un nodo MIN no terminal

- V → todos sus sucesores tienen STATUS V
- D → alguno de sus sucesores tiene STATUS D
- E → cualquier otro caso

▼ Nuevo modelo de solución

- Juegos complejos no se pueden resolver → imposible exploración total hasta terminación
- Nuevo objetivo → encontrar buena jugada inmediata
- Importancia de la heurística en el proceso

▼ Modelo básico

- Arquitectura percepción/planificación/actuación

▼ Búsqueda con horizonte

- Horizonte: profundidad
- Búsqueda parcial
- Uso de heurísticas
- Propagación

▼ Fórmula de complejidad → B^P

- B → factor de ramificación (cuántas opciones tiene cada jugador por turno)
- P → profundidad de búsqueda

▼ Heurísticas para la búsqueda en árboles de juegos

▼ Evaluación para atrás

- Propaga valores desde las hojas 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

▼ Anchura de la búsqueda

- Describe cuántas jugadas se consideran por turno (factor de ramificación)

▼ Algoritmo Minimax

▼ Valor minimax $V(J)$ de un nodo J

- Si es J un nodo terminal → $V(J) = f(J)$

▼ En otro caso

- Generar los k sucesores de J

▼ Calcular $V(J_k)$

- Si J es un nodo MAX → $V(J) = max[V(J), V(J_k)]$
- Si J es un nodo MIN → $V(J) = min[V(J), V(J_k)]$

▼ Algoritmo alfa-beta

▼ Poda alfa-beta

- Mismo resultado que el algoritmo minimax con menos esfuerzo computacional

▼ Cotas alfa y beta

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

▼ Algoritmo

▼ Valor $V(J, alfa, beta)$

▼ Si J es nodo terminal

- Devolver $V(J) = f(J)$

▼ En otro caso

- Sean $J_1, ..., J_k, ...J_b$ todos los sucesores de J

- ▼ Si J es un nodo MAX
  - $alfa = max[alfa, V(J_k, alfa, beta)]$
  - ▼ ¿$alfa >= beta$?
    - Sí → devolver beta
    - No → continuar
  - ▼ ¿k=b?
    - Sí → devolver alfa
    - No → k = k+1 y volver a empezar
- ▼ Si J es un nodo MIN
  - $beta = min[beta, V(J_k, alfa, beta)]$
  - ▼ ¿$alfa >= beta$?
    - Sí → devolver beta
    - No → continuar
  - ▼ ¿k=b?
    - Sí → devolver alfa
    - No → k = k+1 y volver a empezar

▼ Debilidades

▼ Complejidad cuando el factor de ramificación crece

- Mejor caso → B^(P/2)
- Peor caso → B^P

- Definición de una buena función heurística

▼ Juegos con aleatoriedad

▼ Búsqueda en Árboles Monte Carlo (MCTS)

▼ Características

- No requiere funciones heurísticas explícitas
- Realiza simulaciones aleatorias completas desde el estado actual
- Evalúa los nodos en base al promedio de sus resultados

- Funciona bien cuando el espacio de estados es muy grande

▼ Fases

- Selección
- Expansión
- Simulación
- Retropropagación

▼ Juegos con azar

▼ Características

- Se introducen nodos de azar → valores ponderados por probabilidad
- Se modelan como árboles de decisión con nodos de esperanza matemática

▼ Ejemplos

▼ AlphaZero (DeepMind)

- MCTS + redes neuronales
- Aprende desde cero sin datos previos

▼ ChessBench

- Entrenado con millones de partidas
- Usado para investigación en evaluación de movimientos