

Tema 3. Búsqueda en juegos y Búsqueda para problemas de satisfacción de restricciones.

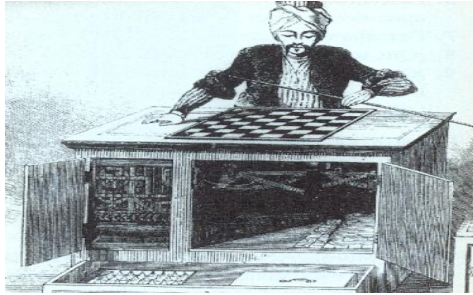
Parte1: Búsqueda en juegos

Búsqueda en juegos

- Búsqueda en juegos con adversario
 - Juegos como problemas de búsqueda
 - Estrategias básicas de búsqueda en juegos
 - Técnicas complementarias

Juegos

- XVIII (1760), **Wolfgang Kempelen**: ajedrecista mecánico



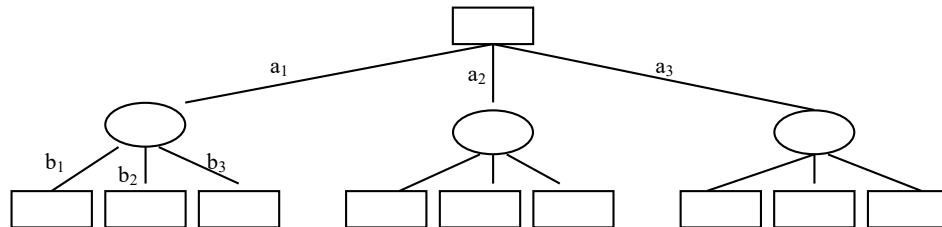
- XIX, **Charles Babbage**: máquina analítica, ajedrez
- '50 Alan Turing, ajedrez
<http://www.chessgames.com/perl/chessgame?gid=1356927>
- 1963, **Arthur Samuel**, damas
- 1997, **Deep Blue**, ajedrez, gana al campeón mundial de ajedrez
- Ajedrez
 - Cada jugada ▶ 35 posibles movimientos
 - Cada partida ▶ aproximadamente 100 jugadas
 - **Búsqueda exhaustiva imposible**

Juegos como problemas de búsqueda

- Imposible generar todo el árbol de búsqueda
 - Generar hasta un determinado nivel de profundidad
 - Aplicar alguna **función de evaluación $f(N)$**
 - Devuelve un valor numérico que indica cómo de bueno es un estado
 - **MAX maximizará esta función y MIN minimizará** dicha función.
 - En algunos casos la función nos puede devolver valores como PIERDE, GANA o EMPATA, siempre referidos a MAX
 - **Objetivo del análisis del árbol: determinar valor del nodo raíz (inicio de la jugada). A este valor se le denomina **valor MiniMax****

Juegos como problemas de búsqueda

- **Estado (N)**: configuración del juego en un momento dado
- **Árbol de juego**. Cada arista de ese árbol indica un posible movimiento. **Una rama completa contempla una posible jugada**



- **En cada nivel se van alternando los jugadores**
- **Factor de ramificación (B)**: número de posibles movimientos que se pueden realizar

Estrategia exhaustiva: MiniMax

- Genera todos los nodos del árbol hasta la profundidad deseada
- Evalúa cada nodo hoja
- Asigna un valor al nodo raíz
 - o si la decisión la toma el jugador MIN, asociar a ese nodo el mínimo de los valores de sus hijos, y el máximo en caso de MAX
- Cuando decimos que aplicamos el algoritmo lo que realizamos es calcular el valor de $V(N)$

Algoritmo minimax.

V(N) Entrada: nodo N

Salida: valor minimax de dicho nodo

Si N es nodo terminal entonces devolver $f(N)$

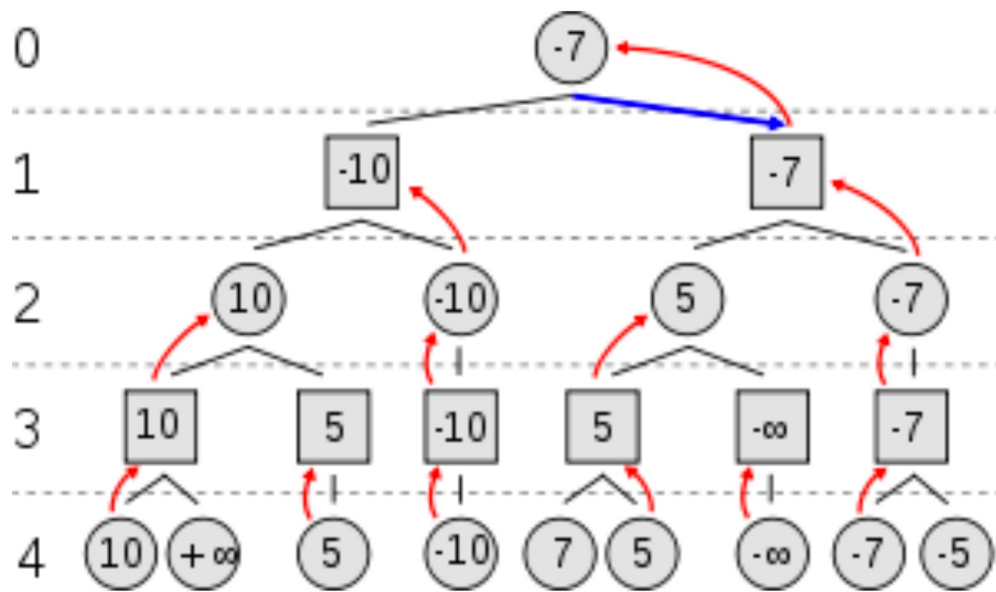
Sino generar todos los sucesores de N: N_1, N_2, \dots, N_b

FinSi Si N es MAX entonces devolver $\max(V(N_1), V(N_2), \dots, V(N_b))$

FinSi Si N es MIN entonces devolver $\min(V(N_1), V(N_2), \dots, V(N_b))$

FinSi

Falgoritmo



T3. Estrategias de búsqueda avanzada y juegos

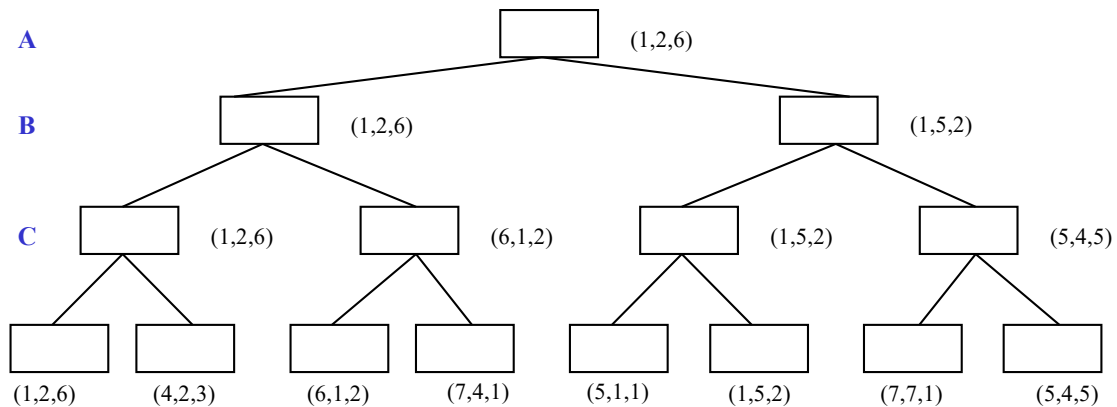
Para repasar: Minimax

<https://www.youtube.com/watch?v=QJm7EKDRuc>

T3. Estrategias de búsqueda avanzada y juegos

Juegos multijugador

- Minimax extendido a juegos de más de dos jugadores
- Se sustituye el valor de un nodo por un vector de valores (tantos valores como jugadores)
- Hoja->utilidad del estado desde el punto de vista de cada jugador



Árbol de juegos con tres jugadores

Estrategia de poda: α - β

La poda alfa beta es una técnica de búsqueda que reduce el número de nodos evaluados en un árbol de juego por el Minimax.

α es el valor de la mejor opción hasta el momento a lo largo del camino para MAX, esto implicará por lo tanto la elección del valor más alto

β es el valor de la mejor opción hasta el momento a lo largo del camino para MIN, esto implicará por lo tanto la elección del valor más bajo.

El valor MiniMax de un nodo estará siempre acotado

$$\alpha \leq V(N) \leq \beta$$

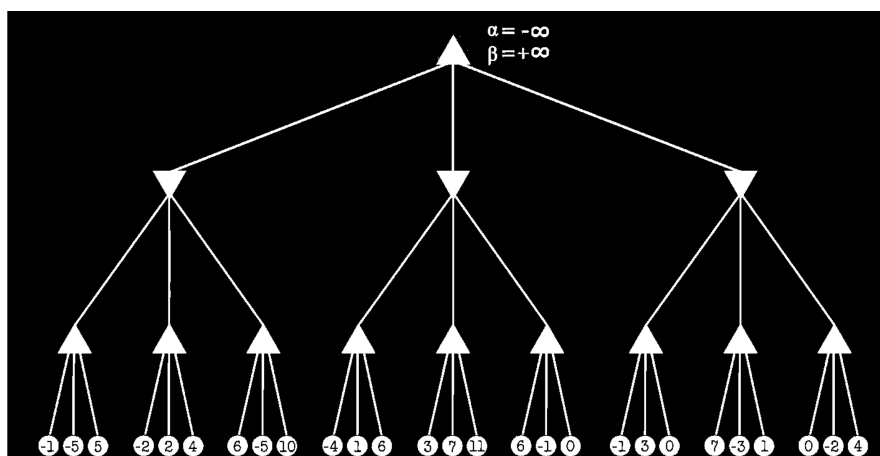
Al principio inicializamos $\alpha = -\infty$ y $\beta = \infty$ al no tener ninguna evidencia.

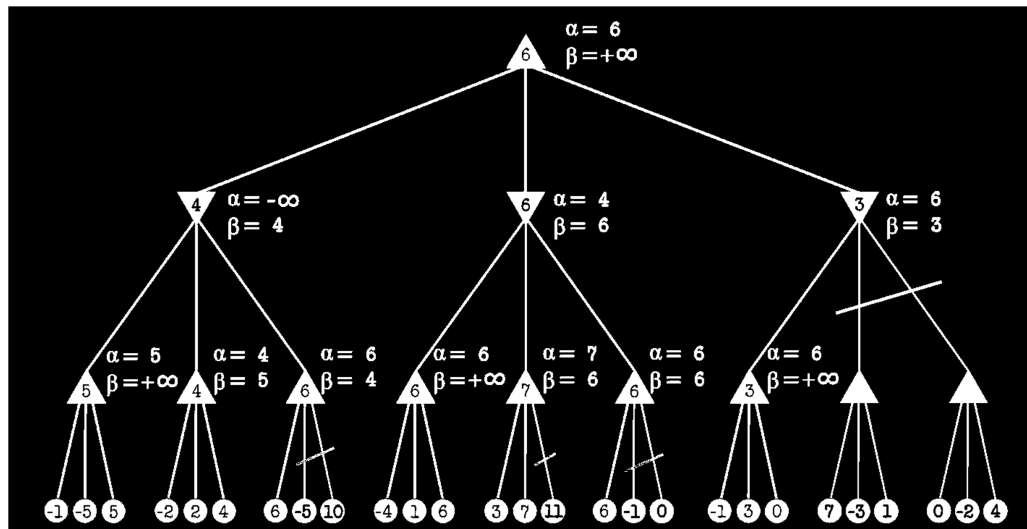
Esta búsqueda alfa-beta va actualizando el valor de los parámetros según se recorre el árbol. El método realizará la poda de las ramas restantes cuando el valor actual que se está examinando sea peor que el valor actual de α o β para MAX o MIN, respectivamente.

Algoritmo $\alpha - \beta$. $V(N, \alpha, \beta)$
Entrada: Nodo N , valores α y β .
Salida: Valor minimax de dicho nodo.

Si N es nodo hoja **entonces** devolver $f(N)$.
sino
 Si N es nodo MAX **entonces**
 Para $k = 1$ hasta b **hacer**
 $\alpha = \max[\alpha, V(N_k, \alpha, \beta)]$
 Si $\alpha \geq \beta$ **entonces** devolver β **FinSi**.
 Si $k = b$ **entonces** devolver α **FinSi**.
 FinPara.
 sino
 Para $k = 1$ hasta b **hacer**
 $\beta = \min[\beta, V(N_k, \alpha, \beta)]$
 Si $\alpha \geq \beta$ **entonces** devolver α **FinSi**.
 Si $k = b$ **entonces** devolver β **FinSi**.
 FinPara.
 FinSi
FinSi

Para repasar: Una muy buena animación explicando el funcionamiento de la poda alfa – Beta:
<https://www.youtube.com/watch?v=I0y-TGehf-4>





T3. Estrategias de búsqueda avanzada y juegos

Técnicas complementarias

- Uso de movimientos de libro
- Espera del reposo
- Técnica de bajada progresiva
- Poda heurística
- Continuación heurística

Uso de movimientos de libro

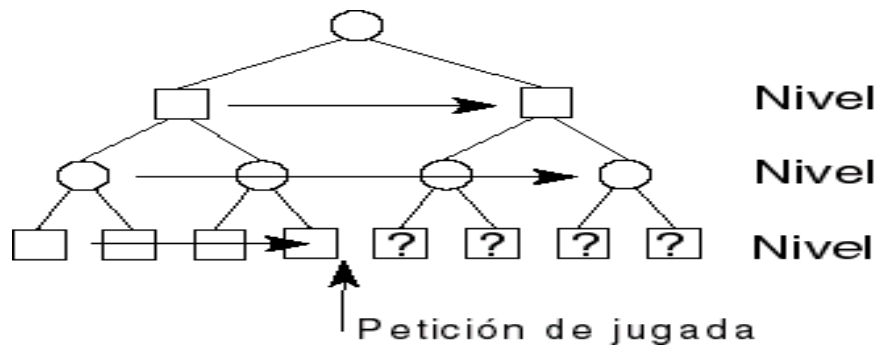
- Imposible seleccionar un movimiento consultando la configuración actual del juego en un catálogo y extrayendo el movimiento correcto.
- Razonable para algunas partes de ciertos juegos.
 - En ajedrez, tanto la secuencia de apertura como los finales están muy estudiados.
- El rendimiento del programa puede mejorarse si se le proporciona una lista de movimientos (movimientos de libro) que deberían realizarse en dichos casos.
- Se usa el libro en las aperturas y los finales combinado con el procedimiento MiniMax para la parte central de la partida
- Conocimiento + búsqueda

Espera del reposo

- Busca evitar el efecto horizonte
- Condición adicional de corte de recursión en minimax sería alcanzar una situación estable
- Si se evalúa un nodo y éste cambia su valor de manera drástica después de explorar un nivel más, la búsqueda debe continuar

Técnica de bajada progresiva

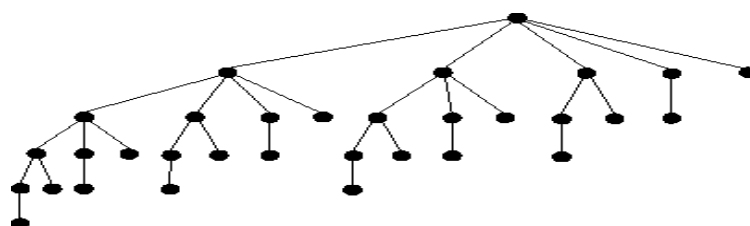
- **Restricciones de tiempo:** algoritmos presentados anteriormente no adecuados.
- **Técnica de bajada progresiva**
 - Recorrer nodos por niveles
 - Al llegar la petición de jugada, devolver la solución del último nivel que se haya completado.



Poda heurística

- Objetivo: **reducir B** desarrollando únicamente los **mejores movimientos** de cada nivel.
- $g(N)$: Función adicional de evaluación
 - De bajo coste.
 - Versión simplificada de $f(N)$.
 - Reordenación de nodos: el primer nodo de un nivel es el de mayor $g(N)$.
- Factor de ramificación:

$$\text{Factor}(\text{Nodo}) = \text{Factor}(\text{Padre}(\text{Nodo})) - \text{Rango}(\text{Nodo})$$



Continuación heurística

- Intento de evitar el **efecto horizonte**.
 - Provocado por la limitación en profundidad: solo se puede tener conocimiento hasta la profundidad seleccionada.
- Secuencia:
 - Desarrollar en anchura hasta un determinado nivel.
 - Seleccionar un subconjunto de nodos terminales para desarrollar búsquedas más profundas.
 - Selección dada por un conjunto de heurísticas directamente relacionadas con el juego.

Tema 3. Estrategias de búsqueda.

Bibliografía

- Stuart Russell, Peter Norving. **“Artificial Intelligence: A Modern Approach, Global Edition”** Ed. Pearson. Prentice Hall. 2021.