

Búsqueda con Adversarios

Jorge A. Baier

Departamento de Ciencia de la Computación
Pontificia Universidad Católica de Chile

Santiago, Chile



- Ya vimos algoritmos de búsqueda en donde hay un solo agente que modifica el mundo.
- Los algoritmos anteriores se pueden extender al caso de múltiples agentes cooperadores. (¿cómo?)



- Ya vimos algoritmos de búsqueda en donde hay un solo agente que modifica el mundo.
- Los algoritmos anteriores se pueden extender al caso de múltiples agentes cooperadores. (¿cómo?)
- Pero no se pueden extender cuando un agente actúa como adversario.
- Una situación particular: juegos. Ej: ajedrez, damas y go.



El Juego como Problema de Búsqueda

- El problema de búsqueda consiste en este caso en, dada una configuración del juego, encontrar la jugada que, a futuro, produzca un beneficio máximo, considerando que el otro agente tratará de hacer lo mismo.



El Juego como Problema de Búsqueda

- El problema de búsqueda consiste en este caso en, dada una configuración del juego, encontrar la jugada que, a futuro, produzca un beneficio máximo, considerando que el otro agente tratará de hacer lo mismo.
- El algoritmo más simple: Minimax.
- En este algoritmo identificamos dos jugadores: MAX y MIN. El objetivo es encontrar la mejor movida para MAX.
- Supondremos que MAX mueve inicialmente y que luego se turnan para jugar.



- **Estado inicial:** Es una configuración inicial del juego más una indicación de quién tiene la próxima movida.
- **Operadores:** Jugadas legales se pueden hacer en el juego.
- **Condición Terminal:** Determina cuándo el juego se acabó.
- **Función de Utilidad:** Da un valor numérico a una configuración final de un juego. En el gato o ajedrez, esta función toma valores en $\{0, 1, -1\}$ (empate, gano, pierdo), pero en otros juegos puede corresponder al puntaje obtenido por el jugador.
- **Función de evaluación:** es una contraparte de la función heurística. La función de evaluación califica a los estados según qué tan buenos son para MAX.



El algoritmo Minimax I

El algoritmo Minimax es el siguiente:

- Generar el árbol de búsqueda hasta un nivel de profundidad $2k$. (MAX va en la raíz)
- Aplicar la *función de evaluación* a cada hoja del árbol.
- Repetir mientras $k \geq 0$
 - Calcular la utilidad de los nodos de nivel $2k - 1$ como la *mínima* utilidad entre la de sus hijos.
 - Calcular la utilidad de los padres de los nodos de nivel $2k - 2$ como la *máxima* utilidad entre la de sus hijos.
 - $k = k - 1$
- La mejor jugada corresponde al hijo de mayor utilidad en el nodo inicial.



El Gato: un ejemplo sencillo I

- Para ejemplificar el algoritmo, consideremos el juego del gato. En este juego podemos usar la siguiente función de evaluación para un tablero t :

$$E(t) = N_A(t) - N_C(t)$$

donde $N_A(t)$ es el número de filas, columnas o diagonales abiertas para MAX (donde aún puede ganar) y $N_C(t)$ es el número de filas, columnas o diagonales abiertas para MIN.

- Si t es un tablero ganado por MAX, $E(t) = \infty$ y si es un tablero perdido, $E(t) = -\infty$.



Un ejemplo

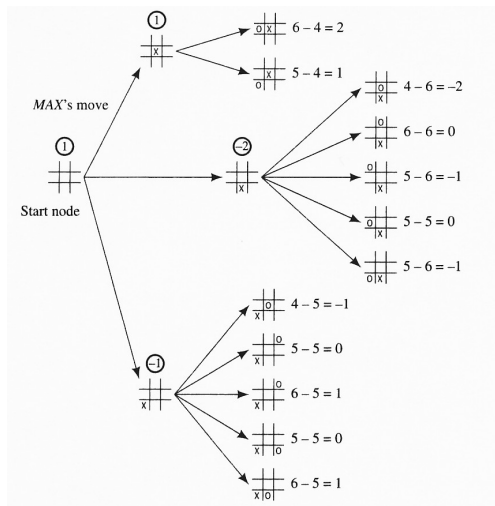


Figure: Primera etapa en la búsqueda del gato



Un ejemplo

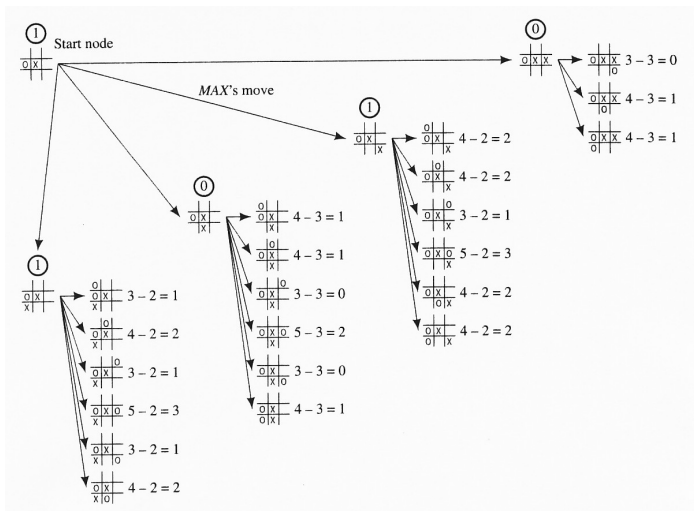


Figure: Segunda etapa en la búsqueda del gato

Un ejemplo

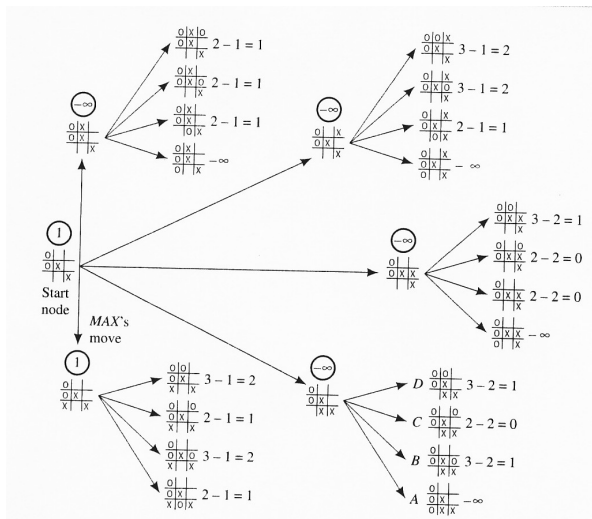


Figure: Última etapa en la búsqueda del gato



- El ejemplo que hemos considerado es bastante sencillo. En juegos más complejos será necesario considerar más de dos niveles en el árbol. Esto significa que para escoger la jugada correcta en juegos de mayor complejidad será necesario analizar un gran espacio de búsqueda.
- En general, en todos los juegos de este tipo, cuando podemos generar el espacio de búsqueda completo, tenemos una estrategia óptima. ¿por qué?



- De hecho, se ha estimado que el espacio de búsqueda del ajedrez tiene aproximadamente 10^{40} nodos. Para generar este espacio demoraríamos orden de 10^{21} siglos (para tener una idea de qué tanto tiempo es, la edad del universo se ha estimado en 10^8 años). Este cálculo está hecho bajo el supuesto que un suceso puede en un ciclo en un procesador de 3000MHz.
- Peor aún es el caso del Go, que tiene un factor de ramificación promedio de 361 (el del ajedrez es sólo 35).
- Una desventaja del algoritmo Minimax es que debemos generar el espacio de búsqueda antes de calcular la evaluación de los estados.



- Consideremos el árbol de juego de la última etapa del gato (figura ??). Supongamos que un nodo hoja es evaluado en cuanto es generado. Después de evaluar el nodo A no tiene sentido seguir generando ni evaluando los nodos B , C o D . El mismo tipo de poda se puede aplicar cuando las posiciones en la búsqueda no representan juegos ganados para MIN o MAX.
- Consideremos ahora la primera etapa del gato (figura ??). Supongamos que la búsqueda se realiza usando una estrategia DFS y que cada vez que una hoja es generada su se computa su evaluación. Supongamos, además, que también se calculan las evaluaciones para los nodos no- hoja, en cuanto es posible.



La poda Alfa-Beta II

- Al calcular el valor para el nodo A , sabemos que el valor del nodo inicial (Start Node, en la figura) está acotado inferiormente por -1 . A este valor se le conoce como *valor alfa*.
- ¿Qué pasa si estamos explorando el nodo C y ya sabemos que el valor para A es -1 ?

La respuesta es que no necesitamos seguir evaluando ningún sucesor de B pues, a lo más, B tendrá utilidad -1 .

Al calcular el valor de C sabemos que B tiene un límite superior de -1 . A este valor se le conoce por *valor beta*. Notemos que:

- Los valores alfa de los nodos MAX nunca pueden decrecer.
- Los valores beta de los nodos MIN nunca pueden crecer.



La poda Alfa-Beta III

Dada estas restricciones podemos establecer las siguientes reglas para el podado del árbol de búsqueda:

- La búsqueda es abandonada bajo todo nodo MIN que tiene un valor beta menor o igual al valor alfa de alguno de sus antecesores MAX.
- La búsqueda es abandonada bajo todo nodo MAX que tiene un valor alfa mayor o igual al valor beta de alguno de sus antecesores MIN.

Durante la búsqueda, los valores alfa y beta se computan de la siguiente manera:

- El valor alfa de un nodo MAX es igual al mayor valor calculado en sus sucesores.
- El valor beta de un nodo MIN es el menor valor calculado en sus sucesores.



La poda Alfa-Beta IV

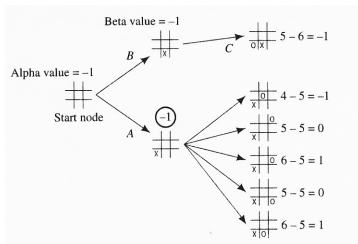


Figure: Primera etapa de la búsqueda del gato



La poda Alfa-Beta V

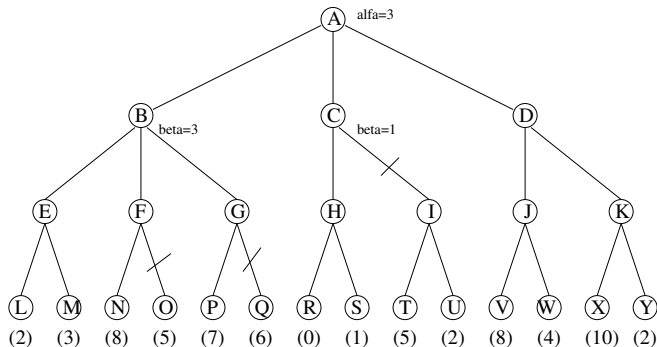


Figure: Poda alfa-beta antes de revisar el nodo *D*.



Alfa-Beta en pseudo código I

- Si P es la profundidad máxima a la que se quiere llegar, entonces llamando a $AB(s, -\infty, +\infty, 0, P)$ se obtiene el recorrido del árbol con poda alfa-beta.

```
AB(nodo,  $\alpha$ ,  $\beta$ , prof, limite)
1  if prof = limite
2    then return EVAL(nodo)
3  for each  $n_i \in \{n_1, n_2, \dots, n_k\} = \text{Sucesores}(\text{nodo})$ 
4    do if prof mod 2 = 0
5      then  $\alpha \leftarrow \max\{\alpha; AB(n_i, \alpha, \beta, \text{prof} + 1, P)\}$ 
6         if  $\alpha \geq \beta$ 
7           then return  $\beta$ 
8         if  $i = k$ 
9           then return  $\alpha$ 
10     else  $\beta \leftarrow \min\{\beta; AB(n_i, \alpha, \beta, \text{prof} + 1, P)\}$ 
11         if  $\beta \leq \alpha$ 
12           then return  $\alpha$ 
13         if  $i = k$ 
14           then return  $\beta$ 
```



- Desde los años 60's que se construyen aplicaciones que juegan ajedrez.
- Las primeras aplicaciones estaban bastante lejos del nivel de un campeón.

Un gran salto en el rendimiento se produjo con es Belle (1982).

La característica principal de Belle es que posee hardware especial para generar posiciones y calcular su función de evaluación.

- Deep Blue (1997) ha superado al ser humano en el ajedrez.



- El 1 de Octubre de 2002, Chessmaster 9000 ganó al campeón estadounidense de ajedrez.
- Actualmente, Chinook (1992) es el campeón mundial de ajedrez.
Chinook usa poda alfa-beta y bases de datos de juegos perfectos para tableros con menos de 7 piezas.
- En 2016, AlphaGo ganó a Lee Sedol, rankeado dan 9 en el juego.



Más de juegos III

- AlphaGo usó técnicas de aprendizaje automático para aprender funciones de evaluación y políticas, todo esto combinado con Monte Carlo Tree Search, un algoritmo de búsqueda aleatorizado.
- Bridge Baron (Great Games) y GIB (Mat Ginsberg) son los dos programas más conocidos de bridge. Éstos no han alcanzado superar el nivel de un humano.
- Programas para este juego están aún lejos de obtener un desempeño similar al humano.
- Poki (U. de Alberta) es uno de los programas más populares.



- Go-moku es similar al juego del gato. Se juega sobre un tablero de 15×15 . El objetivo es hacer una línea (vertical, diagonal u horizontal) de 5 fichas.
- Este juego está resuelto (1992), es decir, los programas tienen una estrategia óptima para jugar el juego cuando comienzan jugando.

