Algoritmo Minimax y Alpha-Beta

Nombre: Edison huiñaizaca

¿Qué es el algoritmo Minimax?

El algoritmo minimax es el principal algoritmo de búsqueda entre adversarios. Es el algoritmo más importante devido aque , los demás algoritmos suelen ser variaciones u optimizaciones suyas. En teoría el algoritmo minimax es un método de decisión para minimizar la pérdida máxima esperada en juegos con adversario. En conclucion, podríamos decir que este algoritmo se encarga de elegir el mejor movimiento para el jugador suponiendo que su rival escogerá el peor movimiento para el jugador.

Este algoritmo se suele usa en juegos como tic-tac-toe, go, ajedrez, Isola, damas y muchos otros juegos de dos jugadores.

Ejemplo

En la siguiente ilustración podemos ver un ejemplo del funcionamiento del Minimax.

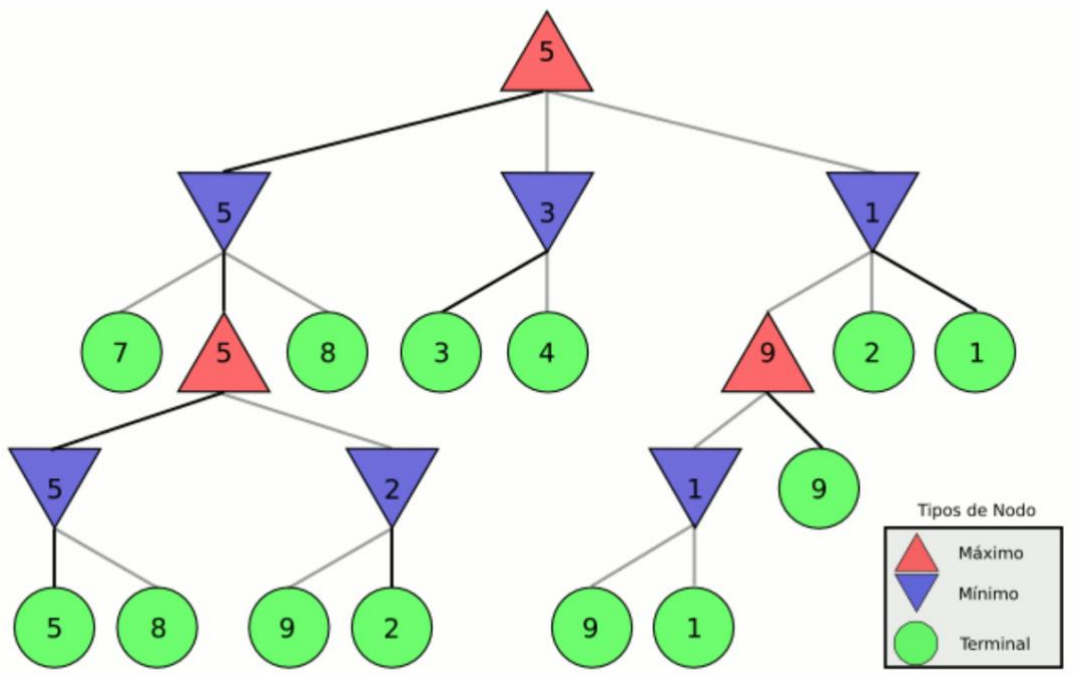


Ilustración 2: Ejemplo del algoritmo Minimax

1. Inicialmente se genera el árbol de juego.
2. Mediante una función de utilidad se les da un valor a los nodos terminales, los verdes en el caso del ejemplo, la cual tiene un rango de 1 a 9.
3. Una vez evaluados los nodos terminales podremos ascender a valores superiores otorgándoles un valor. Los nodos del nivel 5 ya disponen de un valor, pues son terminales.
4. Por ello pasamos a calcular los valores de los nodos de nivel 4. Como son de tipo MIN tendrán el menor valor de sus hijos, por ello se seleccionan los valores 5 (entre 5 y 8), 2 (entre 2 y 9) y 1 (entre 9 y 1).
5. Una vez calculados los valores de los nodos de nivel 4, pasaríamos a calcular los valores de nivel 3. Al ser de tipo MAX tendrán el mayor valor de sus hijos, por lo que se selecciona el 5 (entre 5 y 2) y el 9 (entre 1 y 9).

6. Ahora tocaría calcular los valores de nivel 2, de tipo MIN, por lo que se escogen los menores de nuevo: 5 (es el menor de 7,5 y 8), 3 (menor de 3 y 4) y 1 (menor de 1,9 y 2).
7. Finalmente, el nodo raíz es MAX, por lo que escogerá el mayor que es 5. Esto nos permitirá tomar la decisión de cuál es el movimiento adecuado.

¿Qué es el algoritmo poda alfa beta?

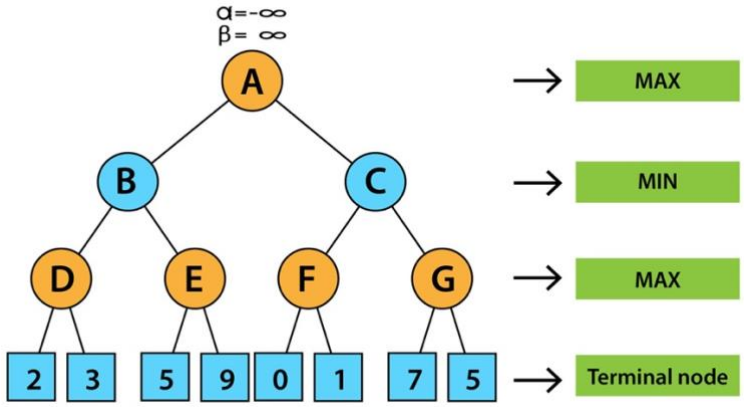
La poda alfa beta es una técnica de búsqueda que reduce el número de nodos evaluados en un árbol de juego por el algoritmo Minimax. Y es una técnica muy utilizada en programas de juegos entre adversarios como el ajedrez, el tres en raya o el Go.

El problema de la búsqueda Minimax es que el número de estados a explorar es exponencial al número de movimientos. Partiendo de este hecho, la técnica de poda alfa-beta trata de eliminar partes grandes del árbol, aplicándolo a un árbol Minimax estándar, de forma que se devuelva el mismo movimiento que devolvería este, gracias a que la poda de dichas ramas no influye en la decisión final.

Entre los pioneros en el uso de esta técnica encontramos a Arthur Samuel, D.J Edwards y T.P. Hart, Alan Kotok, Alexander Brudno, Donald Knuth y Ronald W. Moore

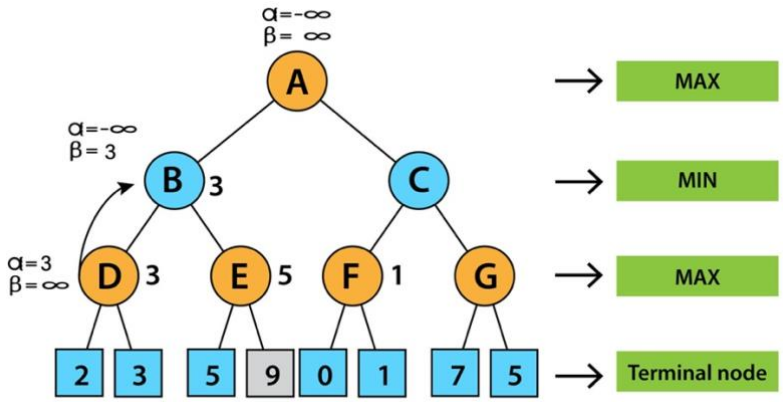
Ejemplo del algoritmo poda alfa beta

Primero comenzaremos con el movimiento inicial. Inicialmente definiremos los valores alfa y beta como el peor de los casos, es decir, $\alpha = -\infty$ y $\beta = +\infty$. Podaremos el nodo solo cuando alfa sea mayor o igual que beta.

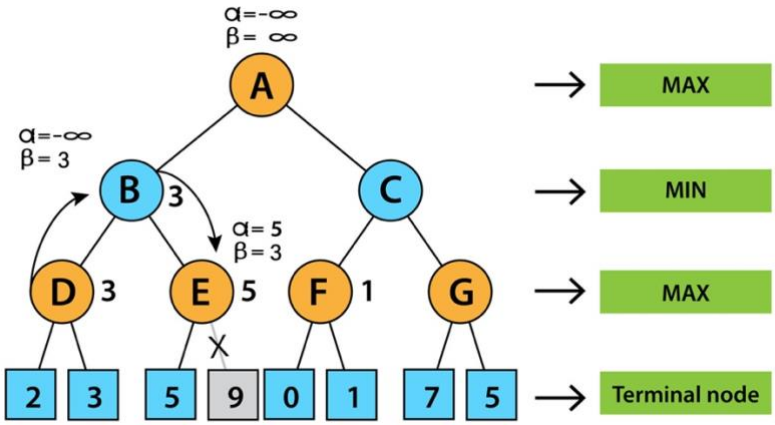


2. Dado que el valor inicial de alfa es menor que beta, no lo podamos. Ahora es el turno de MAX. Entonces, en el nodo D, se calculará el valor de alfa. El valor de alfa en el nodo D será máximo (2, 3). Entonces, el valor de alfa en el nodo D será 3.
3. Ahora el próximo movimiento será en el nodo B y su turno para MIN ahora. Entonces, en el nodo B, el valor de alfa beta será min (3, ∞). Entonces, en el

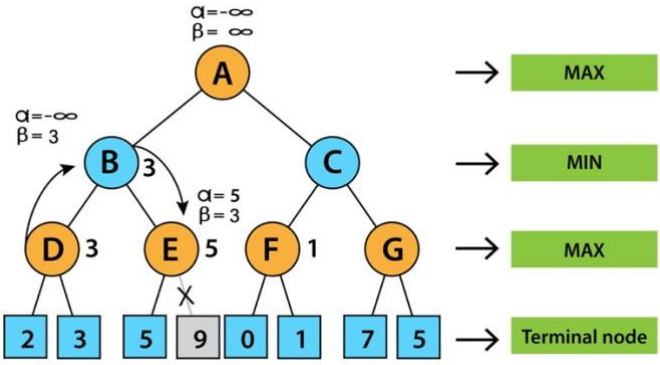
nodo B, los valores serán alfa = - ∞ y beta será 3.



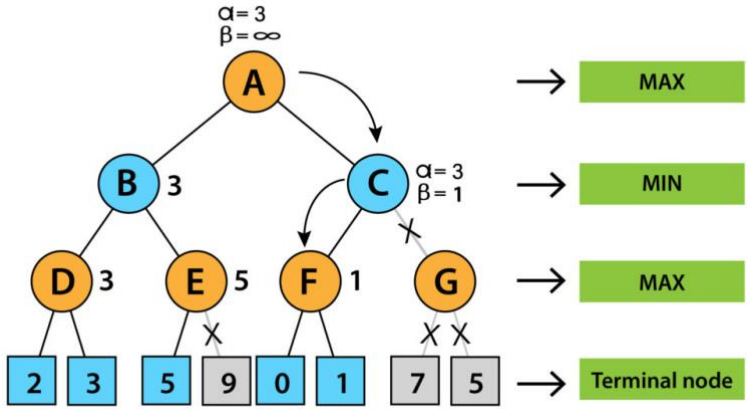
En el siguiente paso, los algoritmos atraviesan el siguiente sucesor del Nodo B, que es el nodo E, y también se pasarán los valores de α = -∞ y β = 3.
4. Ahora es el turno de MAX. Entonces, en el nodo E buscaremos MAX. El valor actual de alfa en E es - ∞ y se comparará con 5. Entonces, MAX (- ∞, 5) será 5. Entonces, en el nodo E, alpha = 5, Beta = 5. Ahora, como podemos ver ese alfa es mayor que beta, lo que satisface la condición de poda para que podamos podar el sucesor correcto del nodo E y el algoritmo no se atravesará y el valor en el nodo E será 5.



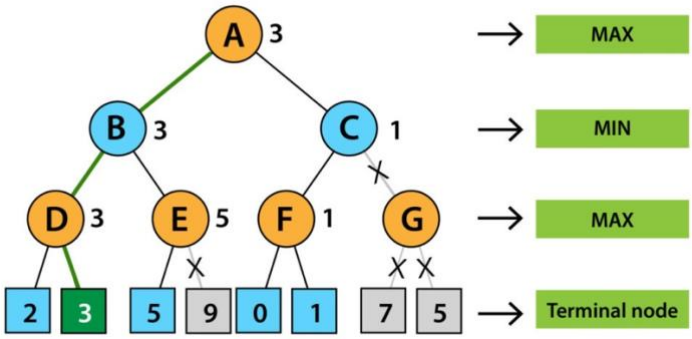
6. En el siguiente paso, el algoritmo vuelve al nodo A desde el nodo B. En el nodo A, alfa se cambiará al valor máximo como MAX (- ∞, 3). Así que ahora el valor de alfa y beta en el nodo A será (3, + ∞) respectivamente y se transferirá al nodo C. Estos mismos valores se transferirán al nodo F.
7. En el nodo F, el valor de alfa se comparará con la rama izquierda que es 0. Entonces, MAX (0, 3) será 3 y luego se comparará con el hijo derecho que es 1, y MAX (3,1) = 3 todavía α sigue siendo 3, pero el valor de nodo de F se convertirá en 1.



8. Ahora el nodo F devolverá el valor del nodo 1 a C y se comparará con el valor beta en C. Ahora es su turno para MIN. Entonces, MIN (+ ∞, 1) será 1. Ahora en el nodo C, α = 3 y β = 1 y alfa es mayor que beta, lo que nuevamente satisface la condición de poda. Entonces, el próximo sucesor del nodo C, es decir, G, se eliminará y el algoritmo no calculó todo el subárbol G.



Ahora, C devolverá el valor del nodo a A y el mejor valor de A será MAX (1, 3) será 3.



El árbol representado anteriormente es el árbol final que muestra los nodos que se calculan y los nodos que no se calculan. Entonces, para este ejemplo, el valor óptimo del maximizador será 3.

Bibliografia

https://www.mygreatlearning.com/blog/alpha-beta-pruning-in-ai/
https://es.wikipedia.org/wiki/Poda_alfa-beta
https://link.springer.com/chapter/10.1007/978-1-4757-1968-0_13

http://www.cs.us.es/~fsancho/?e=107
https://riuma.uma.es/xmlui/bitstream/handle/10630/12713/CristinaGomezGomezMemoriaTFG.pdf?sequence=1&isAllowed=y