

INTRODUCCION A LA INTELIGENCIA ARTIFICIAL

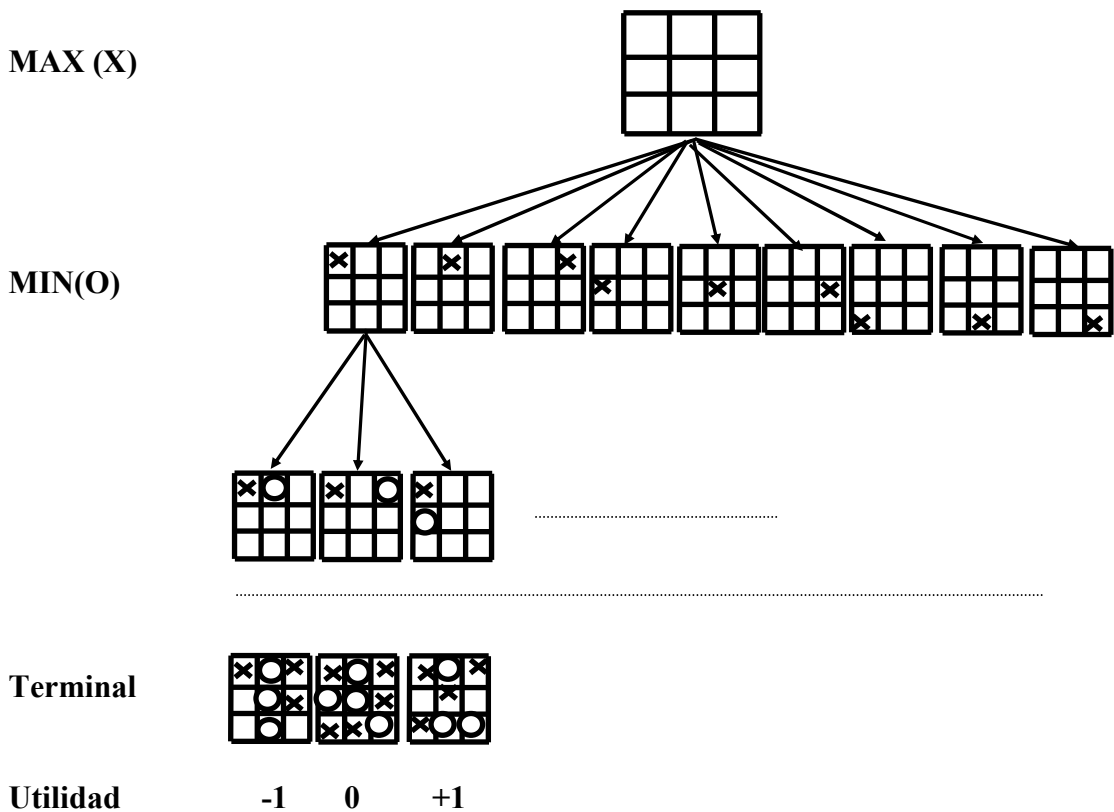
MÓDULO 5- JUEGOS COMO PROBLEMA DE BÚSQUEDA

Referencias: Inteligencia Artificial – Russell and Norvig Cap.5.
Artificial Intellingence Nils Nilsson Ch.3

Se trata el problema de la búsqueda de una solución cuando hay un oponente que responde con su propia estrategia, se eligen los juegos porque el mundo se puede describir con pocas reglas y es fácil de representar, se dicen de información perfecta porque los jugadores perciben el mundo en forma completa, el ajedrez es un juego típico, se asemejan mas al mundo real que un problema de búsqueda simple, el oponente introduce la incertidumbre, y se debe manejar el problema de contingencia

5.1 ALGORITMO MIN-MAX

Se aplica para juegos con dos participantes, uno llamdo MAX y otro MIN, vamos a ilustrar su funcionamiento con el juego fel Ta-Te-Ti, se debe tener en cuenta que MAX juega primero y la estrategia que se desarrolla es para que MAX gane.



Para definir en formalmente el juego se deberá establecer:

Estado Inicial: posición del tablero y una indicación de quien debe jugar

Operadores: que normalmente constituyen las reglas del juego

Prueba terminal: Como termina y quien gana

Función de Utilidad: si no se quiere/puede evaluar todo el espacio de soluciones se deberá disponer de una función de utilidad que evalúa la bondad de cada jugada.

5.2 ALGORITMO MINIMAX

1- Genere todo el árbol hasta alcanzar los nodos terminales (o una profundidad especificada)

2- Obtenga el valor de utilidad a cada nodo terminal

3- Retrotraiga los valores de utilidad, nivel por nivel, desde los estados terminales, hasta llegar al nodo inicial de la siguiente manera. A los nodos en que le toca jugar a MAX (nivel MAX), se le asigna como valor de utilidad, el **máximo** valor de la función de utilidad de todos los nodos hijos (la mejor jugada de MAX). A los nodos en que juega MIN (nivel MIN) se le asigna el **mínimo** valor de la función de los nodos hijos (la mejor jugada de MIN).

4- Seleccione la jugada que tiene valor de utilidad más alto.

La complejidad en tiempo del algoritmo MINIMAX es $O(b^m)$ donde b es el factor de ramificación y m la profundidad donde se encuentra el nodo terminal más próximo, si bien la búsqueda se hace primero en profundidad, se deben explorar todos los nodos terminales, salvo que se ponga un límite de profundidad, con lo cual la decisión es imperfecta y no se puede garantizar el resultado.

5.3 PODA ALFA-BETA

El procedimiento es idéntico al Minimax pero evita expandir todos los nodos cuando el valor retornado por un nodo hace que este sea imposible de seleccionar.

Se supone que se alcanzó un primer nodo terminal o un límite de profundidad (la búsqueda siempre se hace primero en profundidad), entonces ya está en condiciones de retrotraer valores Min y Max **parciales**, para el caso de valores Max parciales se los llama Alfa y para el caso de los valores Min se los llama Beta.

Los valores que adopta Alfa son un límite inferior para los valores de Max y los valores de Beta son un límite superior para la elección de Min.

i) Los valores Alfa de los nodos Max son crecientes.

ii) Los valores Beta de los nodos Min son decrecientes.

Se pueden entonces escribir las siguientes reglas que permiten discontinuar la búsqueda:

1- Se puede discontinuar la búsqueda debajo de un nodo Min que tiene un valor Beta menor o igual al valor Alfa de cualquiera de sus Ancestros. El valor final retornado por este nodo es su valor Beta. Este valor puede no ser el mismo que el del algoritmo MinMAX pero la selección de la mejor movida será idéntica.

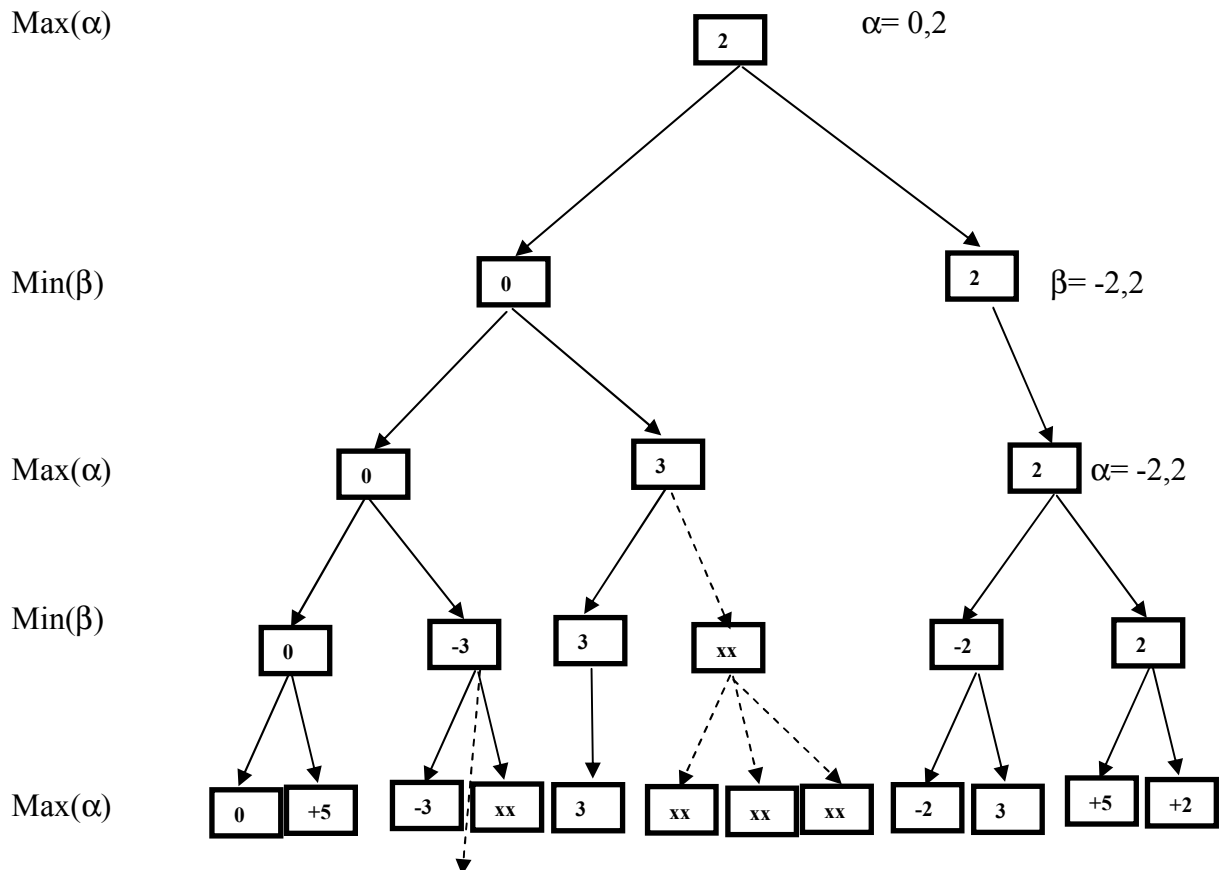
2- Se puede discontinuar la búsqueda debajo de un nodo Max que tiene un valor Alfa mayor o igual que el valor Beta de cualquiera de sus Ancestros Min. El valor final retornado de este nodo Max puede ser su valor Alfa

Durante la búsqueda los valores Alfa y Beta son calculados de la siguiente forma:

a) El valor Alfa actual de un nodo Max es igual al mayor de los valores Beta finales (valor retornado) de sus sucesores.

b) El valor Beta actual de un nodo Min es igual al menor de los valores Alfa finales (valor retornado) de sus sucesores.

Cuando se utiliza la regla uno se dice que se realizo una poda Alfa del árbol de búsqueda y cuando se usa la regla 2 se hizo una poda Beta.



5.4 LA EFICIENCIA EN EL PROCEDIMIENTO ALFA-BETA

Para realizar la poda Alfa-Beta al menos parte del árbol de búsqueda debe generarse a la máxima profundidad. Los valores alfa y beta se deben basar en valores estáticos de nodos terminales, se usa entonces, algún tipo de búsqueda en profundidad

El valor retornado al nodo raíz es idéntico al valor estático de algún nodo terminal, si se encuentra ese nodo terminal primero, entonces la poda es máxima.

Si la profundidad de la solución es d y el factor de ramificación es b

El número nodos terminales es $N = b^d$

Si la generación de nodos es afortunadamente la mejor, primero el valor máximo para los nodos Max y mínimo para Min.(ni de casualidad)

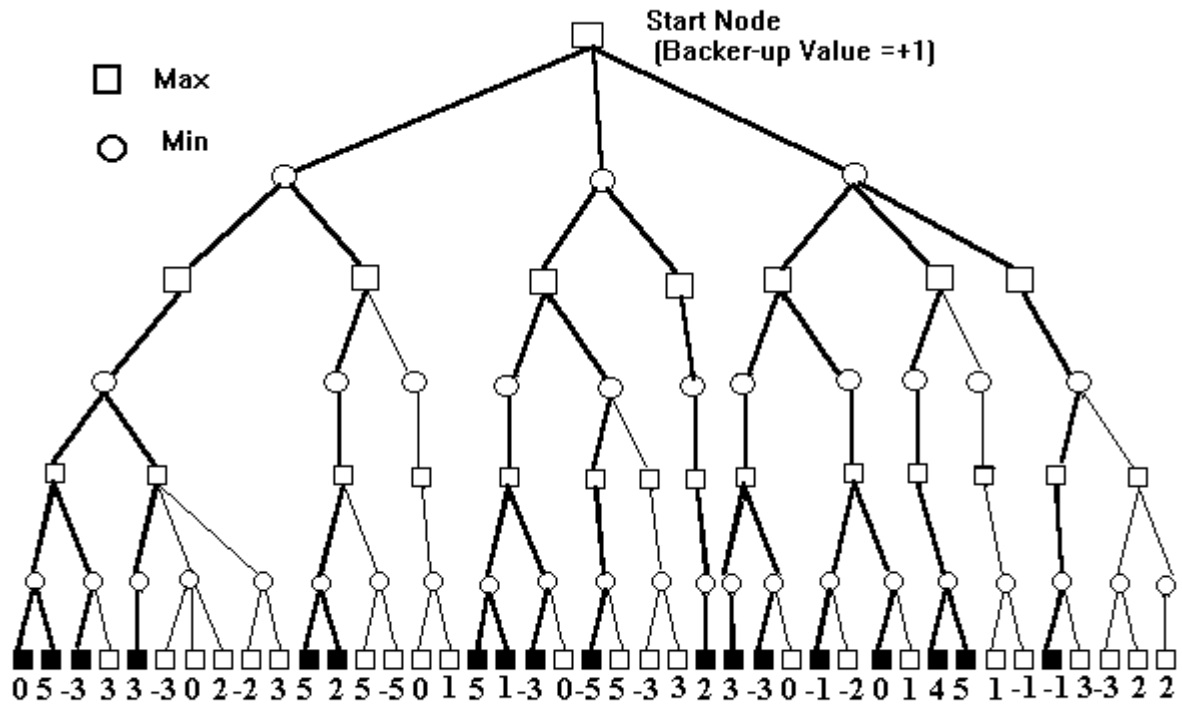
El número nodos terminales es $N = 2b^{d/2} - 1$ para d par

El número nodos terminales es $N = b^{(d+1)/2} + b^{d(d-1)/2}$ para d impar

La eficiencia de Alfa-Beta permite explorar árboles del doble de profundidad, si disponemos de un método de ordenamiento eficiente de valor de utilidad.

Otro ejemplo de poda Alfa-Beta

Definición: Tanto una - la poda alfa - como la otra - la poda beta - son métodos para limitar la búsqueda en los árboles de búsqueda del algoritmo MiniMax, basados en considerar racionalmente que es inútil explorar la totalidad de la estructura del árbol decisional, porque ni a "mí" ni a "mi contrincante" nos va a convenir introducirnos en algunas de las ramificaciones, pues están dominadas ya sea para mí (poda alfa), ya sea para el contrincante (poda beta).

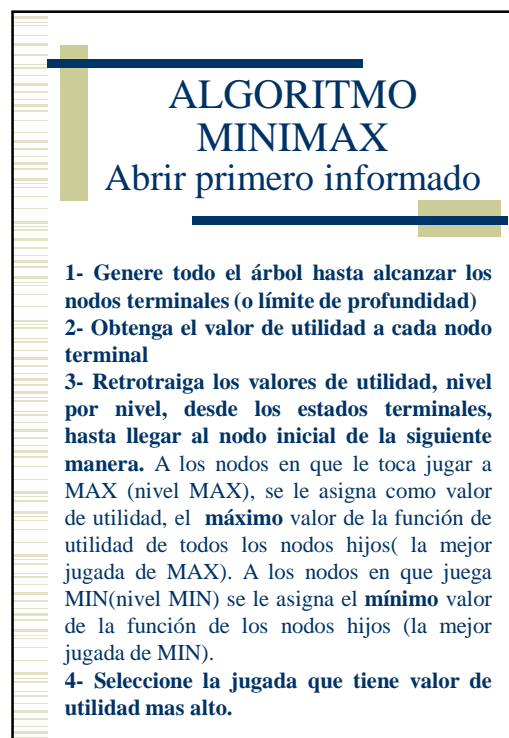
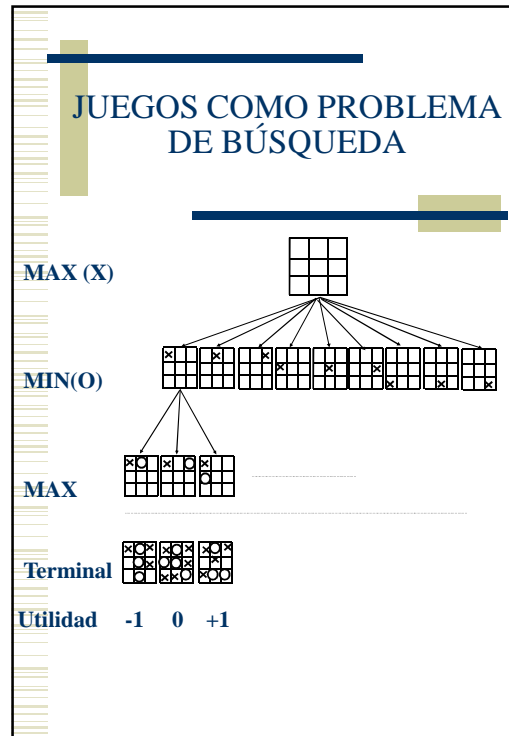


Sea el árbol de búsqueda donde se ha estudiado anteriormente la función de evaluación y sus resultados aparecen transcritos en el séptimo nivel. Sin saber esas cifras, le toca decidir al jugador maximizante cuál jugada le es óptima (qué cifra ubicar en el cuadrado del primer nivel).

Estudiaremos la poda más sencilla de este árbol. Busque en el séptimo nivel la secuencia 3,-3 y 0 (cerca del centro, no confundir con otra que está a la izquierda) y anótela en un papel, con los círculos y cuadrados de más arriba enlazados con 3,-3 y 0.. Hay que llenar los dos círculos MIN del sexto nivel relacionados con dichas tres cifras. En el primer círculo lo único que cabe es ese "3". Ese "3" es un candidato a ser seleccionado por MAX (el cuadrado en blanco del quinto nivel). Veamos cual es el otro candidato. Durante una búsqueda primero en profundidad desde el quinto nivel MAX halla un "-3" que le encantaría a MIN. Para evitar la "tentación" de MIN eligiendo ese buen valor, MAX pierde todo interés en esa rama y no la elige: lo hace con "3" (dominante) en lugar del "-3" (dominado). Ha podado por poda alfa la ramita entre el círculo con "3" y el cuadrado con 0. Podar es sinónimo de omitir el cálculo que da 0. Obsérvese que en el mapa ese camino se dibujó pálido y el cuadrado de 0 se ha dejado vacío. Entendiendo este ejemplito, el lector puede reconocer 5 podas alfa más entre el sexto y el séptimo nivel.

Ahora una novedad. Se llaman podas beta las que decide MIN. Por ejemplo MIN (ubicado en los diferentes nodos circulares del cuarto nivel) no tiene interés en favorecer a MAX y podar tres ramas a la izquierda, entre el quinto y el séptimo nivel, cuando encuentra una primera evidencia de su irrelevancia para sus intereses. Ubicado MAX en el tercer nivel elegirá podar por poda alfa la rama que

acaba en -3 y 3 - así como otra rama que acaba en -3,2,2. Desde el segundo nivel habrá dos podas beta adicionales. Total el esfuerzo de cálculo se reduce fuertemente. Obsérvese que la poda es función del orden arbitrario en que se ubican los nodos superiores. La secuencia ganadora es la que contiene un 2 en cada nivel (casi al centro). Si por casualidad empezamos por la secuencia ganadora, la poda es bastante mayor. O sea que la eficiencia de la poda es función del orden de procesamiento de los nodos por una búsqueda primero en profundidad.



PODA ALFA-BETA

i) Los valores Alfa de los nodos Max son crecientes.
 ii) Los valores Beta de los nodos Min son decrecientes.

Reglas que para discontinuar la búsqueda:

1-discontinuar la búsqueda debajo de un nodo Min que tiene un valor Beta menor o igual al valor Alfa de cualquiera de sus Ancestros.
 2- Discontinuar la búsqueda debajo de un nodo Max que tiene un valor Alfa mayor o igual que el valor Beta de cualquiera de sus Ancestros

a) El valor Alfa actual de un nodo Max es igual al mayor de los valores Beta finales
 b) El valor Beta actual de un nodo Min es igual al menor de los valores Alfa finales

PODA ALFA-BETA

Profundizar primero Informado

Max(α)
Min(β)
Max(α)
Min(β)
Max(α)

profundidad d factor de ramificación b
 nodos terminales es $N = b^d$
 Para una poda óptima
 $N = 2b^{d/2} - 1$ para d par
 $N = b^{(d+1)/2} + b^{d(d-1)/2}$ para d impar

