

Universidad Del Valle de Guatemala
Departamento de Computación
Inteligencia Artificial
Grupo 6

Diana Lucía Fernández Villatoro - 21747
Jennifer Michelle Toxcón Ordoñez - 21276
Daniel Esteban Morales Urizar - 21785
Brandon Rolando Sicay Cumes - 21757

Laboratorio 6

Task 1 - Teoría

- 1. En un juego de suma cero para dos jugadores, ¿cómo funciona el algoritmo minimax para determinar la estrategia óptima para cada jugador? ¿Puede explicarnos el concepto de "valor minimax" y su importancia en este contexto?**

Teniendo en cuenta que el algoritmo minimax comienza en el nivel de profundidad de búsqueda específico, para cada nivel se consideran todas las jugadas disponibles que el jugador actual y el oponente tengan disponibles, para estas se evalúa cada posible secuencia de movimientos hasta cierto nivel de probabilidad mediante una función de evaluación que asigna un valor a cada posición del juego.

Posteriormente se selecciona la jugada que maximiza el valor de la posición para el jugador actual mientras se asume que el jugador oponente también jugará de manera óptima. Este proceso se repite recursivamente para poder explorar las ramificaciones del árbol del juego hasta que se alcanza una posición terminal.

Por su parte, el valor minimax es un valor representativo de la evaluación de una posición específica desde el punto de vista del jugador actual, el cual se encuentra tratando de maximizar su ganancia y minimizar la ganancia del otro jugador. Ahora, su importancia para este concepto radica en que al tener en cuenta las posibles respuestas del oponente, el valor minimax ayuda a calcular las mejores secuencias de juego, alternando entre maximizar y minimizar el valor de las posiciones del árbol de búsqueda, tratando de, como se dijo anteriormente, maximizar la ganancia propia y minimizar la del oponente, tratando de llevar al jugador a la victoria del juego.

- 2. Compare y contraste el algoritmo minimax con la poda alfa-beta. ¿Cómo mejora la poda alfa-beta la eficiencia del algoritmo minimax, particularmente en árboles de caza grandes? Proporcione un ejemplo para ilustrar la diferencia en la complejidad computacional entre la poda minimax y alfa-beta.**

Específicamente la poda alfa-beta se usa dentro del algoritmo minimax para poder tener una mejor ejecución. En sí este algoritmo puede llegar a ser particularmente costoso en escenarios con un espacio de búsqueda grande, por lo que el aplicar la poda alfa-beta puede ayudar a reducir dicho costo computacional de una forma bastante significativa, esto mediante la eliminación de nodos que no resultan significativos para la decisión final que se tomará.

A pesar de que se eliminan ciertos nodos del árbol original, esta poda tiene la característica de que no afecta la calidad de las decisiones que se toman por el

algoritmo en cuestión, pues se eliminan ramas irrelevantes únicamente, lo que mejora la eficiencia específicamente en árboles de caza grandes al reducir su tamaño significativamente.

Tomando como ejemplo un árbol de búsqueda para un juego simple como lo es el *totito*, donde se tiene una profundidad máxima de tres niveles, donde, para efectos de simplificación, se tienen tres posibles movimientos para el jugador *MAX* y tres posibles respuestas para el jugador *MIN*. En el caso del algoritmo minimax se tienen que explorar todas las ramas del árbol para poder proporcionar la mejor jugada, y teniendo en cuenta que hay tres movimientos posibles para cada jugador, en un árbol con una profundidad de tres niveles, el número de nodos que se deben explorar serían $3^3 * 3^3 * 3^3$, lo cual es equivalente a 729 nodos a evaluar.

Por otra parte, si se aplica la poda alfa-beta se permite eliminar las ramas no prometedoras, reduciendo significativamente la cantidad de los 729 nodos que se deben evaluar. A pesar de que la reducción en el número de nodos que se evalúan varía dependiendo del juego y la configuración del árbol de búsqueda, la aplicación de esta poda mejora la eficiencia del algoritmo, contribuyendo a una búsqueda más rápida y eficiente, especialmente en escenarios con grandes espacios de búsqueda.

3. ¿Cuál es el papel de expectiminimax en juegos con incertidumbre, como aquellos que involucran nodos de azar o información oculta? ¿En qué se diferencia el expectiminimax del minimax en el manejo de resultados probabilísticos y cuáles son los desafíos clave que aborda?

Expectiminimax es una extensión del algoritmo minimax donde los nodos pueden representar eventos aleatorios o con información oculta donde, su objetivo no es únicamente maximizar su propia ganancia y minimizar la del jugador contrario, sino también abordar la incertidumbre que se asocia a los eventos aleatorios y todos los posibles movimientos del oponente y sus posibles resultados.

Debido a que los nodos representan eventos aleatorios el proceso de expectiminimax considera todas las posibles ramas que podrían surgir como resultado de eventos aleatorios o las decisiones del oponente, en donde a cada nodo se le calcula la expectativa, siendo esta la suma ponderada de los valores de los hijos, teniendo como las ponderaciones a las probabilidades de ocurrencia de distintas ramas. Teniendo todo esto calculado, el jugador actual elige la acción que maximiza su expectativa, considerando los posibles eventos aleatorios que puedan surgir y que el oponente actuará de la manera óptima.

La principal diferencia entre expectiminimax y minimax es en cómo se manejan los resultados probabilísticos en juegos con incertidumbre. Específicamente se tiene la característica de que el minimax asume que todos los movimientos del oponente y los resultados del juego son deterministas y conocidos, mientras que en expectiminimax se aborda la incertidumbre al incorporar nodos de probabilidad en el árbol de búsqueda.

Por su parte, entre los desafíos claves que se abordan está que el modelado de eventos aleatorios requiere una representación mucho más específica y adecuada a los posibles eventos aleatorios que puedan ocurrir. Así mismo se tiene que poder tomar

decisiones de calidad bajo cierta incertidumbre, estas siendo racionales y lógicas a pesar de no conocerse todos los resultados posibles de un juego, también se debe poder anticipar y reaccionar a las acciones del jugador contrario sin conocer sus estrategias o posibles jugadas completamente para adaptarse de mejor manera a la información oculta del juego