

Informe Alpha-Beta Pruning

Pilar Bocage (268398)

Josefina Ferrari (268395)

Grupo M7A

2024

En el artículo titulado “Alpha-Beta Pruning”, escrito por Carl Felstiner y publicado el 9 de mayo de 2019, se aborda el tema de cómo los algoritmos de juegos de computadora pueden ser optimizados para mejorar su eficiencia.

Las computadoras construyen un árbol con las diferentes posibilidades dentro del juego, para luego trabajar hacia atrás para encontrar el movimiento que le ofrecerá el mejor resultado. Se identifica que el mayor desafío es encontrar la forma de evitar buscar el mejor camino en ciertas partes del árbol de juego en las que es evidente que no se encontrará la solución óptima. Por lo que en el artículo se investiga diferentes maneras para reducir la cantidad de nodos que tendrá que evaluar la computadora en un árbol de juego.

Se describe en primer lugar un algoritmo llamado Minimax el cual consiste en trabajar hacia arriba desde las hojas de un árbol de juego. El objetivo del juego para la computadora es obtener el valor máximo posible encontrado en el último nivel del árbol, mientras que para el humano es conseguir el menor valor posible.

El algoritmo recorre recursivamente el árbol de decisiones de un juego, evaluando los posibles movimientos de ambos jugadores. Cuando es el turno de la computadora, busca el máximo valor posible dentro de los movimientos siguientes. Cuando es el turno del humano, busca el mínimo valor. Así la computadora tomando en cuenta lo que probablemente decida el humano en su turno, logra obtener el mejor movimiento que puede realizar para maximizar sus posibilidades de ganar.

La mayor desventaja que se identifica en este algoritmo es que el algoritmo debe atravesar el árbol de juego, lo que podría requerir de mucho tiempo y un gran consumo para la computadora. Además, otra desventaja destacada es que se supone que el humano siempre tomará la mejor decisión de juego cuando se le presente la oportunidad, lo que no siempre es así. En esos casos, el resultado para la computadora será al menos tan bueno como si el humano estuviera jugando perfectamente.

Para solucionar el mayor conflicto, se encuentra una versión expandida del algoritmo minimax, al que se lo denomina Alpha-Beta Pruning. En este algoritmo, se agregan dos variables, alpha y beta, alpha es el máximo valor que se puede obtener al nivel actual o

menor y beta es el mínimo valor que se puede llegar al nivel actual o menor. Pruning refiere a poda, está se da durante la exploración, cuando se determina que un nodo no contribuirá a la solución óptima (cuando $\alpha \geq \beta$), se evita seguir explorando sus subárboles.

Cuando $\alpha \geq \beta$, significa que la configuración actual del árbol de búsqueda ya encontró una solución igual o mejor en otro lugar del árbol. α representa el mejor valor encontrado hasta el momento para la computadora y β el mejor valor encontrado hasta el momento para el humano. Por lo que cuando se da esa condición entre los valores, se está indicando que ya se han proporcionado resultados iguales o mejores. Esto evita la exploración adicional de subárboles, ya que se sabe que estos no cambiarán la decisión final del algoritmo.

Esta nueva versión propuesta implica un menor esfuerzo por parte de la computadora. Aunque aún se pueden realizar mejoras, la mayor diferencia en gasto de recursos se mejora con esta expansión.

Además, en el artículo estudiado se realiza un análisis de Hex que destaca la importancia de la teoría de juegos en la informática, mostrando cómo conceptos matemáticos pueden aplicarse para mejorar la eficiencia de los algoritmos. Se demuestran varios lemas que establecen propiedades fundamentales del juego, como el hecho de tener una pieza adicional de tu propio color en el tablero no puede perjudicarte, o que ciertos movimientos de apertura no son estrategias ganadoras. Por lo que no sólo implica reducir el número de nodos evaluados, sino que también entender las estrategias ganadoras fundamentales del juego, lo cual es crucial para desarrollar sistemas de juego más inteligentes y eficaces.

Al estudiar el Tic-Tac-Toe en tres dimensiones, se ilustra cómo los avances en algoritmos de optimización pueden resolver problemas complejos de manera eficiente. Aunque tradicionalmente se juega en una configuración 4x4x4, el enfoque inicial en una versión 3x3x3 demostró ser más accesible computacionalmente. Mediante mejoras al algoritmo minimax, se identificó que el movimiento ganador en un tablero vacío es ocupar el centro, destacando cómo una estrategia bien definida puede reducir

drásticamente el número de posiciones evaluadas. Este estudio no sólo demuestra la aplicabilidad de la teoría de juegos en contextos tridimensionales, sino que también subraya la eficiencia alcanzada al adaptar y optimizar algoritmos para resolver problemas específicos en juegos computarizados.

Un aspecto interesante estudiado en el artículo es el impacto de la aleatoriedad en juegos como Tic-Tac-Toe y Hex. En el Tic-Tac-Toe bidimensional, se observó que la aleatoriedad tiene un efecto insignificante cuando se activan los movimientos forzados, ya que estos limitan las ramificaciones del árbol de búsqueda. Sin embargo, sin movimientos forzados, la aleatoriedad redujo el número de nodos evaluados en un 6-10% en promedio sobre 1000 simulaciones, especialmente cuando el tablero está menos lleno y el árbol de búsqueda es más grande. En el caso de Tic-Tac-Toe tridimensional, la aleatoriedad tuvo un efecto aún menos significativo debido a la alta cantidad de movimientos forzados. En contraste, en el juego Hex, la aleatoriedad mejoró el algoritmo de manera considerable, reduciendo en un 58% y 88% la cantidad de nodos evaluados en tableros de 3x3 y 4x4 respectivamente. Se espera que en tablero más grandes, con árboles de búsqueda más extensos, la aleatoriedad tenga un efecto todavía más notable en la reducción del tiempo de ejecución del algoritmo.

El estudio del impacto de la aleatoriedad no sólo destaca la importancia de considerar este atributo en la optimización de algoritmos, sino que también resalta cómo está puede influir en la eficiencia del proceso de toma de decisiones de la computadora en diferentes contextos de juego. La idea de la aleatoriedad nos hace reflexionar sobre cómo podría aplicarse en otros campos más allá de los juegos, donde la exploración no determinista podría ofrecer soluciones más eficientes y creativas.

El algoritmo minimax y sus mejoras pueden aplicarse fácilmente a otros juegos de tablero finitos similares a Tic-Tac-Toe y Hex, como Connect Four. Esto se logra mediante la implementación de funciones específicas para verificar las condiciones de victoria y ajustando el código para diferentes tamaños de tablero y reglas. Además, se podría mejorar el algoritmo al incluir una función de evaluación para asignar valores a nodos no terminales y ajustar la profundidad de búsqueda en áreas del árbol de juego donde los movimientos tienen un impacto significativo. Estas adaptaciones permiten

que los algoritmos sean más eficientes y se apliquen a una variedad más amplia de juegos.

Comprender las estrategias ganadoras fundamentales y cómo se aplican en diferentes escenarios no solo mejora la eficiencia de los algoritmos, sino que también enriquece nuestra comprensión de la inteligencia artificial y la toma de decisiones estratégicas.

En conclusión, el artículo “Alpha-Beta Pruning” de Carl Felstiner ofrece una exploración detallada y técnica sobre cómo optimizar algoritmos de juegos de computadora para mejorar su eficiencia. A través de un análisis riguroso de los algoritmos Minimax y Minimaxab, el artículo presenta métodos para reducir la cantidad de nodos evaluados en un árbol de juego, destacando cómo estas técnicas pueden ahorrar tiempo y recursos computacionales. Además, al estudiar juegos como Hex y Tic-Tac-Toe en tres dimensiones, el artículo demuestra la aplicabilidad y adaptabilidad de estos algoritmos a diferentes contextos de juego, subrayando la importancia de entender las estrategias ganadoras fundamentales y el impacto de la aleatoriedad en la optimización del rendimiento. El análisis crítico del artículo permite apreciar no sólo los avances en la eficiencia computacional, sino que también la profundidad teórica que sustenta estas mejoras, proporcionando una valiosa contribución al campo de los algoritmos de juegos.