Universidad Simón Bolívar

Departamento de Computación y Tecnología de la Información

CI5437 Inteligencia Artificial I

29 de mayo de 2014

**OTHELLO**

Prof. Blai Bonet Elaborado por:

Gamar Azuaje 10-10051

Wilmer Bandres 10-10055

Juan A. Escalante 10-10227

Un árbol de juego, es un árbol dirigido en donde cada nodo representa un estado de algún juego en particular, y cada arco define un movimiento desde el estado anterior a un nuevo estado.

Existen juegos de dos personas, en donde cada uno tiene un turno y el objetivo de cada persona es jugar lo mejor posible. Este tipo de juegos se pueden representar con un árbol llamado “And-Or”, que está constituido en cada nivel del árbol solamente tiene nodos tipo “And” o solamente tiene nodos tipo “Or”.

La computación entra en juego, a la hora de buscar jugadas óptimas (o menos dañinas), o buenas jugadas para los dos jugadores. En el caso de un juego de ajedrez, puede ser que se busque hacer el jaque mate lo más rápido posible, o simplemente asegurar un jaque mate en una jugada futura. En el juego de damas, se puede ser ganar comiendo todas las fichas a la larga, o ganar lo más rápido posible. La intuición se presta a que buscar una solución óptima es mucho más costoso, computacionalmente hablando, que buscar una solución que se acerque al óptimo. Cuanto más complejo sea el juego (mientras más jugadas posibles en un turno se tenga), el espacio de búsqueda tiende a ser mucho más grande, que en juegos con una cantidad de movimientos posibles más limitada.

Existen diversos algoritmos que pueden ser utilizados, para la búsqueda de las mejores jugadas en un estado específico de un juego, entre los que se encuentran:

* Minimax y Negamax: son algoritmos que buscan la mejor jugada, similar a la forma de backtracking, esto es, prueban todas las posibles jugadas para un jugador en un estado dado, teniendo en cuenta que la mejor jugada para un jugador, se basa en escoger la peor jugada para su contrario.
* Poda alpha-beta con Minimax: es un algoritmo similar a la anterior, solo que recorta el espacio de búsqueda, comparando el mejor resultado (buscando la mejor jugada) que se ha obtenido hasta ahora con las demás posibles jugadas y recortando el espacio de búsqueda cuando se sabe que por alguna rama del árbol de juego es imposible llegar a un mejor escenario, lo que permite recortar esa rama y ahorrar tiempo.
* Scout: es un algoritmo que busca recortar el espacio de búsqueda haciendo uso de una función auxiliar llamada TEST, y recordando los parientes de un nodo del árbol para su beneficio.
* Negascout: es un algoritmo que busca un equilibrio entre poda alpha-beta y scout, uniendo lo mejor de ambos mundo y que en la práctica es el que resulta mejor.

A continuación, se presentan resultados obtenidos utilizando los algoritmos mencionados, utilizados para la búsqueda de una solución óptima de un juego de Othello de 6 x 6, luego de haber hecho varios pasos por la variante principal del juego (y tomando en cuenta que la profundidad de la raíz del árbol de juego es la profundidad 0):

* MiniMax:

|  |  |  |  |
| --- | --- | --- | --- |
| Profundidad | Nodos generados | Nodos generados por segundo | Tiempo |
| 30 | 5 | 3595200 | 0.00 |
| 28 | 13 | 3697200 | 0.00 |
| 26 | 174 | 3595300 | 0.00 |
| 24 | 4353 | 3275200 | 0.00 |
| 22 | 73445 | 3672250 | 0.02 |
| 20 | 3327475 | 3081000 | 1.08 |
| 18 | 85601844 | 3142510 | 27.24 |
| 16 | No termina | No termina | No termina |
| 14 | No termina | No termina | No termina |
| 12 | No termina | No termina | No termina |
| 10 | No termina | No termina | No termina |

* NegaMax:

|  |  |  |  |
| --- | --- | --- | --- |
| Profundidad | Nodos generados | Nodos generados por segundo | Tiempo |
| 30 | 5 | 3235290 | 0.00 |
| 28 | 13 | 3395255 | 0.00 |
| 26 | 174 | 3244987 | 0.00 |
| 24 | 4353 | 3234892 | 0.00 |
| 22 | 73445 | 2448170 | 0.03 |
| 20 | 3327475 | 3081000 | 1.08 |
| 18 | 85601844 | 3176320 | 27.24 |
| 16 | No termina | No termina | No termina |
| 14 | No termina | No termina | No termina |
| 12 | No termina | No termina | No termina |
| 10 | No termina | No termina | No termina |

* Poda alpha-beta con minimax:

|  |  |  |  |
| --- | --- | --- | --- |
| Profundidad | Nodos generados | Nodos generados por segundo | Tiempo |
| 30 | 5 | 3069855 | 0.00 |
| 28 | 13 | 3423975 | 0.00 |
| 26 | 80 | 3242525 | 0.00 |
| 24 | 4353 | 3543457 | 0.00 |
| 22 | 73445 | 3672250 | 0.00. |
| 20 | 3327475 | 3081000 | 0.04 |
| 18 | 260470 | 26047 | 0.1 |
| 16 | 2074971 | 2357920 | 0.88 |
| 14 | 43574642 | 1899500 | 22.94 |
| 12 | 415909955 | 1958330 | 212.38 |
| 10 | 6181062210 | 2256460 | 2739.28 |

* Scout:

|  |  |  |  |
| --- | --- | --- | --- |
| Profundidad | Nodos generados | Nodos generados por segundo | Tiempo |
| 30 | 2 | 2930498 | 0.00 |
| 28 | 9 | 2904549 | 0.00 |
| 26 | 67 | 2939483 | 0.00 |
| 24 | 1550 | 2824729 | 0.00 |
| 22 | 3832 | 2321824 | 0.00 |
| 20 | 74170 | 2472330 | 0.03 |
| 18 | 304674 | 2343650 | 0.13 |
| 16 | 1387259 | 2391830 | 0.58 |
| 14 | 42564855 | 2189550 | 19.44 |
| 12 | 323787340 | 2051360 | 157.84 |
| 10 | 5173234283 | 2351440 | 2200.03 |