Búsqueda Adversaria en Juegos

Sistemas Inteligentes

Profesor: Pablo Schwarzenberg

Nombre: Mario Gonzalez Rivera

**Introducción**

El juego Othello consiste en un tablero de 6 filas por 6 columnas, en el cual se colocan fichas de dos colores distintos, correspondientes a cada jugador.

En su fase inicial, el juego tendrá dos fichas de cada color en el centro del tablero. La jugada es cuando un jugador puede saltar sobre la ficha del otro para así “quitar” las fichas al otro. Gana el que obtiene mas fichas en el tablero. El juego acaba cuando el tablero se completa o un jugador obtiene todas las fichas puestas en el tablero.

Sobre las técnicas utilizadas para resolver este juego computacionalmente, principalmente se utiliza el árbol del juego para este fin. La búsqueda continua hasta llegar a un nodo hoja del árbol, es decir, hasta una jugada terminal. También se implementa el algoritmo Minimax, el cual, calcula la utilidad e intenta maximizar esta o minimizarla.

**Análisis del juego**

El juego corresponde es de tipo suma cero, ya que la una acción que hace un jugar provoca un efecto negativo para otro jugador, porque, el jugador al “saltar” sobre las fichas del otro, estas cambian su color, perjudicando al otro.

También el juego es de información perfecta, ya que, el estado de este es completamente visible para los dos jugadores, es decir, no hay nada de información oculta.

El juego es determinista, ya que, no existe ningún componente aleatorio como: Dados, cartas al azar etc.

**Minimax aplicado al problema**

Ya que este juego es de información perfecta, se puede utilizar el algoritmo Minimax para resolverlo, en esta tarea se aplicó este algoritmo.

Este algoritmo utiliza la utilidad de cada jugador para determinar la jugada que se realizará

Las jugadas posibles en este problema se generaron buscando las fichas adyacentes de cada ficha del jugador, cuando se encontraba una ficha del jugador contrario adyacente, seguía buscando en esa dirección hasta llegar a una ficha vacía. Luego esas jugadas se guardaban en un arreglo de coordenadas posibles.

La utilidad se calculó buscando cada ficha de un jugador en el tablero, es decir, si un jugador poseía 3 fichas en el tablero, este tendría 3 de utilidad.

Entonces, el programa generará las jugadas posibles y evaluará cada jugada en la función de utilidad, para que luego se elija la jugada que maximice la utilidad en el caso de el jugador 1 o minimizar en el caso del agente.

**Dificultad**

La dificultad se controlará a través de el valor de profundidad del árbol, ya que, un numero más bajo producirá que el agente no podría elegir la mejor estrategia para minimizar la utilidad del jugador.