

Inteligencia Artificial

Curso 2018-19

Práctica 3. Métodos de Búsqueda con Adversarios (Juegos): DESCONECTA-4 BOOM

Autor¹: Víctor García Carrera, victorgarcia@correo.ugr.es

Síntesis.

A continuación se detalla el trabajo realizado y el proceso seguido para implementar la práctica 3 de Inteligencia Artificial de búsqueda con adversarios. Se nos proporciona un entorno de simulación con diversas opciones de ejecución para probar nuestro agente deliberativo. Éste se basa en una búsqueda MINIMAX con una heurística propia desarrollada para escoger el mejor movimiento posible (búsqueda con información). Podemos probar a jugar contra él tanto nosotros como otros agentes (ninjas) implementados.

Búsqueda MINIMAX

Debemos comenzar desarrollando el método que implemente la búsqueda MINIMAX. Este algoritmo describe un método de evaluación recursivo de los nodos del árbol de búsqueda, desde un nodo raíz MAX, hasta la profundidad escogida, alternando en profundidad nodos MAX y MIN, donde cada uno busca maximizar o minimizar el peso de sus sucesores (respectivamente), representando a los 2 oponentes donde cada uno, en su movimiento, busca maximizar su heurística, escoger su mejor movimiento. En nuestro caso la profundidad máxima es 4 (según se estableció en el guión de la práctica). Implementamos la función recursiva `EvaluaMinimax()` siguiendo el algoritmo descrito en las diapositivas de teoría, de forma que almacena en una variable pasada como parámetro la acción correspondiente al sucesor elegido por MINIMAX, y retorna el peso de esta acción (del estado que genera). Cabe mencionar que había que generar, para un estado dado, todos sus sucesores (movimientos) posibles, y según fuera MAX o MIN, buscar entre todos ellos aquel que maximiza o minimiza la heurística.

Heurística

La clave para desarrollar un agente con un comportamiento y actuación más o menos inteligente depende fundamentalmente de la heurística implementada para evaluar si un estado es favorable o no. A continuación describo la heurística implementada, posibles mejoras y heurísticas complementarias.

¹ Como autor declaro que los contenidos del presente documento son originales y elaborados por mi. De no cumplir con este compromiso, soy consciente de que, de acuerdo con la “[Normativa de evaluación y de calificaciones de los estudiantes de la Universidad de Granada](#)” esto “conllevará la calificación numérica de cero ... independientemente del resto de calificaciones que el estudiante hubiera obtenido ...”

Heurística implementada

La idea clave que subyace tanto en esta como en las siguientes heurísticas que describo es la de pensar en la forma de jugar (de ganar) en el CONECTA-4, juego bien conocido el cual podemos extrapolar directamente a nuestro juego DESCONECTA-4 BOOM, de forma que aquellas heurísticas buenas buscarán justo lo contrario que las del CONECTA. Por ello, la heurística implementada se basa en que, en el CONECTA-4, buscamos conseguir 4 en línea, por lo que nosotros buscaremos evitar tener 4 en línea. Así mismo, evitamos tener 3 en línea, para evitar con algo más de margen esta situación. De esta forma jugamos “a la defensiva”, buscamos siempre el siguiente movimiento que REDUZCA AL MÍNIMO el número de nuestras conexiones de 3 y 4 fichas. Esta heurística es simple pero potente, y con una buena gestión (incluso con las conexiones de 2 fichas) con pesos proporcionales que den mucha mayor importancia a las conexiones más grandes, puede dar muy buenos resultados.

Posibles mejoras y Heurísticas valoradas

En cuanto a mejoras posibles, mencionar la de contar también las líneas de 2 fichas para evitarlas (en manera proporcional a las líneas de 3 y 4-Muerte fichas). También podría implementarse una “actitud agresiva”, utilizando el mismo concepto de contar líneas de 2-3 fichas, pero con las fichas del rival, para intentar forzar al rival a aumentar al máximo sus fichas en línea o conectadas.

Siguiendo la línea ofensiva, podríamos intentar dotar al agente de “combos”, de manera que en determinadas situaciones, que podrían depender de parámetros como si hay 3 fichas en línea, la ocupación en determinadas filas y columnas adyacentes... tratara de centrar su esfuerzo en una zona del tablero o en determinadas acciones (un subplan).

Un concepto muy interesante a destacar era el de implementar una **HEURÍSTICA QUE EVOLUCIONA LARGO DEL TIEMPO**. Utilizando parámetros del juego como el número de jugada, número de casillas vacías u otros, se podría particionar la heurística de forma que reflejara una actitud cambiante a medida que se desarrolla el juego. De esta forma se implementa una heurística que enfoca los diversos **OBJETIVOS** principales del juego en sus distintas **ETAPAS** (EARLY y por ejemplo LATE).

Otra heurística muy útil es la de **BUSCAR LAS ESQUINAS**, pues de esta forma tiendes a posicionarte en casillas con un numero probablemente menor de conexiones (en los lados, menos conexiones posibles). Muy útil en el **EARLY**.

Anexo.

El código se encuentra en jugador.cpp y jugador.hpp
