UNIVERSIDAD DE GRANADA

ETSIIT
Escuela Técnica Superior de Ingenierías Informática y de Telecomunicación

Práctica 3: Búsqueda con Adversario/Juegos: Conecta-4 BOOM

Inteligencia Artificial

Grupo C1

Autor:
Lugli, Valentino Glauco · YB0819879

1. Análisis del Problema

El problema presentado en esta práctica es una alteración del muy estudiado juego en Inteligencia Artificial de Conecta 4, la mayor diferencia recae ahora en que en esta variante del juego se tiene que cada jugador puede colocar dos fichas y cada cuatro turnos se coloca una ficha "bomba" que quita esa ficha más las fichas del jugador contraria de esa misma fila, adicionalmente es ahora un tablero de 7×7.

Al seguir siendo una variante de Conecta 4, se sabe que este es un juego que puede resolverse por medio de la búsqueda con adversario, ya que son solo dos jugadores. Se ha decidido realizarla con el Algoritmo de Poda Alfa-Beta.

El problema básico es que se tiene que evitar que el jugador contrario, MIN, conecte 4 fichas ya sea horizontal, vertical o diagonalmente mientras que "nosotros", MAX, debemos de realizar dicho cuatro en raya; por lo tanto se debe razonar en que posición hacer jugadas y además aprovechar las oportunidades que brindan la versión modificada del juego como lo que es la ficha bomba.

Para el tablero, se ha investigado y los consejos típicos es comenzar en el centro del tablero, esto es así ya que desde la columna central se tiene la mayor libertad de actuación, es donde más posibilidades hay de colocar fichas que se alineen, esto es opuesto a jugar cerca del borde del tablero ya que ahí se restrige el número de posibles jugadas.

Luego, en la partida usual, siempre es mejor de igual manera mantenerse en el centro del tablero dentro de lo posible y de lo que mejor convenga, claro está que tener fichas alineadas posee un valor añadido estén dónde estén, pero en el centro se provee de una mayor gama de posibilidades.

Aún así, el hecho de poseer fichas separadas entre sí por 4 o menos casillas con espacios vacíos entre ellas presentan oportunidades valiosas para futuras jugadas y naturalmente, es de máxima prioridad tener primero dos y luego tres fichas alineadas ya que eso determinará fácilmente el ganador de esa partida.

Junto a esto, la ficha bomba trae otra oportunidad de realizar jugadas que normalmente no se podrían hacer, ya que esta ficha permite que se puedan colocar otras fichas en posición y una vez se realiza el *BOOM*, las fichas caen y se produce un cuatro en raya.

Por lo tanto, para este problema un conteo de las fichas MAX y MIN, además, otorgarles una valoración adicional dependiendo de la ubicación de las fichas y si estas están conectadas o al menos están alineadas sin tener fichas del contrario entre ellas.

Dicho todo esto, se debe ahora discretizar este conocimiento en una heurística que sea factible para el problema; esto se discute en la sección siguiente.

2. Descripción de la Solución Planteada

2.1. Algoritmo de Búsqueda

Se decidió utilizar el algoritmo de Poda Alfa-Beta por ser una versión más rápida del clásico algoritmo MiniMax, ya que no explorará aquellas ramas que se sabe no podrán dar un mejor valor del que se posee actualmente, se hace uso de las variables apropiadamente llamadas `alpha` y `beta` para

controlar el valor heurístico que se obtiene de cada nodo hoja. A continuación se presenta en pseudocódigo la implementación realizada:

El objeto `tablero` contiene la partida actual donde el `jugadorActual` le toca jugar, dependiendo de quién sea se determinará si es un nodo MAX o MIN si este valor coincide con `MAX`, la cual en la primera llamada desde `Think` se inicializa al jugador que la ha llamado. El resto de las variables tienen un nombre que indican que es lo que realizan.

En esencia, si es un nodo hoja, se realiza la heurística, en otro caso se entra en un bucle que va iterando por todas las posibles configuraciones del tablero, estas las manda recursivamente hasta llegar al caso base, una vez se devuelve la recursión dependiendo de quién es `jugadorActual`, se actualiza `alfa` o `beta` y de `alfa` se almacena la acción que logró esa mejora.

Una vez finaliza el ciclo, ya sea porque se exploraron todas las posibilidades o hubo una poda, ahora esta variable `accionLocal`, se le asigna a la variable que es la que se utilizará para decidir la acción de la partida. Si bien esta variable se actualiza en recursión la última vez será realizada en el nodo raíz garantizando que la acción será la mejor acción desde ese nodo.

Luego, dependiendo si es MAX o MIN, se devuelve el valor que posee de las cotas a la llamada superior o bien a `Think`.

```
funcion alfaBeta(Enviroment: tablero, entero: jugadorActual, accion: &mejorAccion,
   entero: profundidadActual, entero: maxProfundidad, doble: alfa, doble: beta,
   entero: MAX ) : doble
inicio:
    si(profundidadActual == profundidadMax ∨ tablero.JuegoTerminado())entonces
        retornar Valoracion(tablero, MAX)
    sino
        entero: sigAccion <- -1, accionLocal <- -1
        bool posible[8]
        doble valorActual
        Enviroment: sigTablero

        tablero.possible_actions(posible)

        mientras(sigAccion < 8 ∧ alfa < beta) hacer
            sigTablero <- tablero.GenerateNextMove(sigAccion)

            si(posible[sigAccion] ∧ sigAccion < 8) entonces
                valorActual <- alfaBeta(sigTablero, sigTablero.JugadorActivo(),
                   mejorAccion, profundidadActual+1, maxProfundidad, alfa, beta,
                   MAX)

                si(jugadorActual == MAX ∧ valorActual > alfa) entonces
                    accionLocal <- sigAccion
                    alfa <- valorActual

                fsi

                si(jugadorActual != MAX ∧ valorActual < beta) entonces
                    beta <- valorActual
                fsi
            fsi
```

```
        fmientras

        mejorAccion <- (accion)accionLocal

        si(tablero.JugadorActual == MAX) entonces
            retornar alfa
        sino
            retornar beta
        fsi
    fsi
ffunc
```

La función llamada en los nodos hoja, `Valoracion`, tiene implementada la heurística que será presentada a continuación.

2.2. Heurística

La heurística tiene dos partes, un análisis del tablero y una rápida "mirada" futura de la ficha bomba.

Primero que todo, dentro de `Valoracion` se tiene una valoración si el tablero es de una partida terminada. Si es así y ha ganado MAX se le asigna una valoración de 9999 y si gana MIN de -9999. Independientemente de que haya ganado alguien, se realiza el siguiente paso, el cual es un análisis de la "calidad" de la jugada desde el punto de vista de MAX, esto se realizó de esta manera porque aunque se prediga que MIN gane, la partida puede que sea favorable si MIN no realiza las acciones que ha predicho MAX.

1	2	3	4	3	2	1
1	2	3	4	3	2	1
1	2	3	4	3	2	1
1	2	3	4	3	2	1
1	2	3	4	3	2	1
1	2	3	4	3	2	1
1	2	3	4	3	2	1

Figura 1: Valoración de una ficha dependiendo de su posición en el tablero

En la función `checkGrid` se realiza el análisis del tablero para la calidad de la jugada, en primer lugar consiste en almacenar las fichas de los jugadores MAX y MIN que se encuentran en el tablero en una estructura `map` utilizando un registro sencillo llamado `token` que almacena su localización en el tablero y una valoración heurística de su posición en el mismo.

Este valor está almacenado en una matriz constante que posee la disposición de valores como se muestra en la siguiente imagen, de esta manera se le da una mayor peso a las columnas centrales y este valor va disminuyendo hacia las columnas laterales, así la heurística prioriza las jugadas en el centro, aún así, esto no es toda la historia, pues como se sabe, para ganar se tienen que alinear fichas, que es lo siguiente que la heurística hace.

En una función que se llama desde `checkGrid`, denominada `checkConnectedPieces`, se analiza, para cada ficha de MAX si horizontalmente, verticalmente o diagonalmente tiene fichas a su alrededor, no solamente aquellas que sean contiguas sino aquellas que estén separadas por un espacio vacío.

Entonces, en esta función además se sumarse al valor acumulado el valor de la ficha que está siendo "analizada", si tiene una ficha que es contigua se le suma un valor de 10 por cada una que se consigue, si no es contigua no se suma –aún así el valor de la otra ficha será tomado en cuenta cuando sea esta misma analizada–, si se consigue una ficha del jugador contrario finaliza el bucle pues no tiene sentido tomar en cuenta fichas que estén aisladas entre sí.

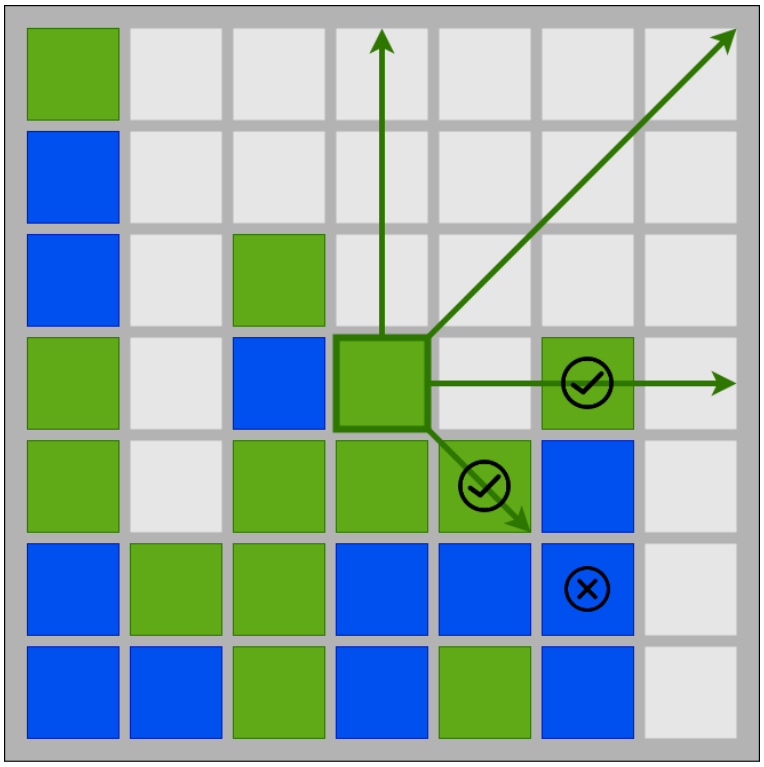

Figura 2: Visualización del análisis por cada ficha de un jugador

Cabe notar que se el análisis por cada ficha se realiza desde la posición de la misma hacia el final del tablero por la derecha, esto es así para que las fichas no se cuenten más de lo necesario, si es cierto que si existen tres fichas en línea también se estarán contando como que hay dos fichas en línea, teniendo una ficha que se solapa.

Por lo tanto, cada ficha se suma y se suma adicionalmente si tiene fichas contiguas.

Este análisis se realiza tanto para MAX como para MIN, y se devuelve la diferencia entre estas dos cantidades la que determina la calidad de la partida. Si MAX tiene más fichas alineadas que MIN, su valoración es mayor, ya que se han sumado más puntos y si la jugada ha sido más en el centro del tablero cada ficha propiamente tendrá más valor y aportará esto al total. Además si se tuvo un cuatro en raya, entonces la calidad aumentará mucho más.

La jugada es más positiva para MAX mientras más alto hacia los valores positivos sea el valor MiniMax y peor mientras más negativo sea.

Adicionalmente, si se posee una ficha bomba, se llama también a la función `checkBoomEffects` dónde el tablero se avanza un movimiento haciendo explotar la bomba y analizando si el tablero resultante es el final de una partida, si es así, si gana MAX, se le añaden otros 100 puntos de valoración y si pierde, -100.

Cabe destacar que la ficha bomba también está contenida en las fichas que se analizan como el resto de las fichas, ya que estas mismas cuentan para un cuatro en raya igualmente.

3. Fuentes Consultadas

- *How to Win at Connect 4* de *Seth Brown*. URL: `https://www.thesprucecrafts.com/how-to-win-at-connect-four-basic-strategy-tips-412539`
- *Quickest Ways to Win at Connect 4!* de *Keith Galli*, vídeo. URL: `https://youtu.be/XISUTn9EeoY`
- *Connect 4 tips to win* de *Gary DeVries*, vídeo. URL: `https://youtu.be/M3KLDctPjKI`