

Práctica 3: Búsqueda con Adversario

El Parchís (BOOM!)



**UNIVERSIDAD
DE GRANADA**

Antonio Javier Rodríguez Romero
Inteligencia Artificial
3º DGIIM

1. Análisis del problema.

Se nos pide diseñar un algoritmo de Poda Alfa-Beta que, junto con una heurística, sea capaz de vencer a una serie de bots, los “ninjas”.

El diseño e implementación del algoritmo en sí es bastante rápido, basta deducir que los nodos que queremos maximizar serán aquellos en los que sea el turno de nuestro jugador y los que queremos minimizar son los del rival.

Sin embargo, el buen funcionamiento de este dependerá casi enteramente de la heurística que usemos para la poda, luego el problema principalmente será el diseño de este.

2. Solución planteada.

a. Aproximación inicial.

Para empezar con el diseño de esta heurística, tendremos que resaltar qué aspectos de un estado determinan la buena o mala posición de un jugador en la partida.

Además, podemos tomar como referencia la heurística que se nos provee de prueba, y nos daremos cuenta de que es bastante acertada en todo lo que implementa.

Estos puntos a tener en cuenta desde el primer momento serán:

- Valor de la posición del jugador: obviamente tendremos que valorar la posición de nuestro jugador en el tablero.
- Valor de la posición del oponente: cuanto mejor sea esta, peor será la valoración global del estado.
- ¿Partida acabada?: tendremos que tener en cuenta antes de comprobar nada más si es un estado de *Game Over* o no, de forma que premiamos si es victoria nuestra o penalicemos si es del oponente.

Ahora desarrollaremos los dos primeros, pero la estructura de nuestra heurística, y que se mantendrá hasta el final, será:

```
double heuristica( const Parchis &actual, int player) {
    int opponent = (player + 1) % 2;

    if( actual.getWinner() == player) return gana;
    else if( actual.getWinner() == opponent) return pierde;
    else {

        double valor_p = valorJugador(actual,player);
        double valor_o = valorJugador(actual,opponent);
        return valor_p - valor_o;
    }
}
```

El siguiente paso será diseñar la valoración de la posición de un jugador.

b. Evolución de la solución

A lo largo de las distintas heurísticas que he diseñado para la resolución del problema, más que cambios he ido añadiendo aspectos que tener en cuenta a favor o en contra del jugador, además de cambiar los pesos de cada uno de estos aspectos para que se tengan más o menos en cuenta.

En un primer momento, los puntos que he detectado y teniendo en cuenta han sido:

- Posición de las fichas de cada color del jugador por separado: esto se debe a que no tendremos que llevar las fichas de los dos colores, sino solo de uno, de manera que tendremos más en cuenta la valoración del color que se encuentre en mejor posición. Dentro de este valoraremos:
 - Distancia a la meta de las fichas, de manera que a mayor distancia menor puntuación.
 - Piezas en casilla segura, de manera que premiamos que estén a salvo.
 - Piezas en la meta.
 - Piezas en casa, favorecemos sacarlas.
 - Si se ha comido alguna ficha
- En caso de que tenga el dado especial disponible, premiaremos en función de la energía que tenga el jugador, es decir, el objeto que pueda usar.

Sin embargo, esto no será suficiente para vencer a los “ninjas”.

En las siguientes implementaciones de la heurística se tomarían en cuenta también:

- Si se ha comido una pieza, quién y de qué color (Pasa a estar en la valoración del jugador, no de color).
- Piezas destruidas en el último movimiento, premiando si son rivales y penalizando si son propias.
- Formación de barreras.
- Pesos de comer ficha dinámicos en función de la cercanía de esta a meta.

Una de las pocas ideas que he tenido que descartar era que la proporción de contribución a la puntuación de cada color fuera variable proporcionalmente a las mismas puntuaciones de ambos. No resultó útil ya que llegaba un punto en el que intentaba avanzar con ambos colores a la vez para mantener más o menos las puntuaciones de ambos iguales. Es muy preferible, por lo menos en mi diseño, que una se tomará en una proporción bastante superior.

c. Solución final

Con diferentes ajustes de los pesos en cada heurística, a cada actualización la nueva versión vencía a las anteriores y, aunque se ha dado el caso en algún punto de que, supongo que debido también al comportamiento de cada ninja, había heurísticas que le ganaban al ninja 3 pero no al ninja 2, la última solución, teniendo en cuenta todos los apartados mencionados previamente, gana a todos los ninjas menos al 3 jugando como *Player 2*, además de dar muy buenos resultados en la competición entre heurísticas,

donde me he mantenido en el top 10 durante casi 2 semanas. Esta heurística es *heurística4()*.