

Memoria practica 4

Heurísticas generadas:

- Heurística 1 (Heurística del jugador perico): Al igual que las otras tres heurísticas lo primero que se contempla es si con el movimiento realizado se gana la partida. Si se gana la partida al valor de la heurística le damos la máxima puntuación para que el bot realice el movimiento, si por el contrario no es un movimiento ganador, le damos el valor mínimo a la heurística para partir de una puntuación en la que no obtenemos la victoria. En esta heurística primero calificamos la puntuación de nuestro jugador y a continuación la de nuestro oponente. De estos dos valores sacamos un valor único que será el valor final de nuestra heurística. Para puntuar el movimiento de nuestro jugador seguimos el siguiente criterio:

- Primero comprobamos el número de fichas debajo de las fichas que tenemos colocadas.

La puntuación que le damos a cada estado es la siguiente:

- 0: Si no hay ninguna ficha debajo.

- 7: Si hay 1 ficha debajo.

- 300: Si hay 2 fichas debajo.

- 1000: Si hay 3 fichas debajo.

- Después realizamos la misma comprobación con las fichas que hay a la derecha, a continuación con las que hay a la izquierda y por último comprobamos las que están colocadas en diagonal (Los pesos que se les da a cada estado son los mismos que en el primer caso)

Una vez obtenida la puntuación de nuestro jugador pasamos a obtener la puntuación del jugador contrario siguiendo el mismo criterio que con nuestro jugador, así como dándole los mismos pesos a las jugadas. Restamos el valor obtenido de nuestro jugador con el del jugador contrario y el resultado se corresponde con el valor de nuestra heurística.

- Heurística 2 (Heurística jugador perico2): Esta heurística es similar a la anterior. Se comprueba primero la puntuación de nuestro jugador con la jugada específica y a continuación se calcula el del jugador contrario. Ambas puntuaciones se restan para obtener el valor final de la heurística. En el caso de nuestro jugador, obtenemos la puntuación de la misma manera que en el caso anterior. En el caso del jugador contrario la evaluación es diferente. Seguimos el mismo análisis que en el caso anterior pero con pesos diferentes y menores que los que se dan en la puntuación de nuestro jugador. De esta manera obtenemos un jugador más ofensivo que el anterior. Los pesos que establecemos en el jugador rival son los siguientes:

- Comprobamos primero el número de fichas debajo de las que el rival tiene colocadas. La puntuación que le damos a cada estado es la siguiente:

- 0: Si no hay ninguna ficha debajo.

- 5: Si hay 1 ficha debajo.

- 190: Si hay 2 fichas debajo.

- 1000: Si hay 3 fichas debajo.

- Después realizamos la misma comprobación con las fichas que hay a la derecha, a continuación con las de la izquierda y por último comprobamos las diagonales.

Finalmente igual que en el caso anterior el resultado obtenido aparece cuando restamos la puntuación de nuestro jugador con la de nuestro oponente.

- Heurística 3 (Heurística jugador CR7HatTrycks): Esta última heurística sigue un planteamiento parecido a las anteriores, pero esta vez sin tener en cuenta las diagonales. El motivo de realizar un

jugador con menos comprobaciones que los anteriores era ver como se comportaban el resto de bots con nuestro jugador. El resultado obtenido con este es considerablemente peor que con los anteriores. Las comprobaciones que no realizabamos en este bot era las diagonales en direccion derecha del jugador contrario.

Heuristica sin terminar de implementar: Ademas de esto tratamos de realizar en python un algoritmo genetico que fuera capaz de establecer los diferentes pesos de las heurísticas para poder obtener una heurística con mejores resultados. Ademas estas heurísticas las generamos viendo las posibles situaciones que se podian dar con dos jugadas en adelante, es decir, despues de nosotros realizar un movimiento y que el rival realizase el suyo. Las heurísticas que utilizamos para este algoritmo genetico fueron las siguientes.

- Heuristica1: Comprobaba el numero de fichas consecutivas en diagonal del rival.
- Heuristica2: Comprobaba el numero de fichas consecutivas en horizaontal del rival.
- Heuristica3: Comprobaba el numero de fichas consecutivas en vertical del rival.
- Heuristica4: Comprobaba el numero de fichas consecutivas en diagonal de nuestro jugador.
- Heuristica5: Comprobaba el numero de fichas consecutivas en horizaontal de nuestro jugador.
- Heuristica6: Comprobaba el numero de fichas consecutivas en vertical de nuestro jugador.
- Heuristica7: Contaba el numero de fichas tanto del rival como nuestras en la columna 1.
- Heuristica8: Contaba el numero de fichas tanto del rival como nuestras en la columna 2.
- Heuristica9: Contaba el numero de fichas tanto del rival como nuestras en la columna 3.
- Heuristica10: Contaba el numero de fichas tanto del rival como nuestras en la columna 4.
- Heuristica11: Contaba el numero de fichas tanto del rival como nuestras en la columna 5.
- Heuristica12: Contaba el numero de fichas tanto del rival como nuestras en la columna 6.
- Heuristica13: Contaba el numero de fichas tanto del rival como nuestras en la columna 7.

El motivo de realizar heurísticas que contasen el numero de fichas de cada columna era darle a nuestro bot una caracteristica mas para poder evaluar el tablero. Aunque suponemos que con nuestro bot ideal la los valores de estas heurísticas iban a ser poco significativos.

El objetivo de nuestro algoritmo genetico era calcular los pesos de las diferentes heurísticas para obtener nuestro bot ideal. El problema que tuvimos finalmente es que la manera en que creabamos los hijos de los jugadores mas eficientes era demasiado aleatoria, por lo que perdiamos gran cantidad de las caracteristicas de los mismos. De esta manera al irse desarrollando nuestro jugador se volvia peor en iteraciones siguientes, mejor...