



UNIVERSIDAD DE GRANADA

Escuela Técnica Superior de Ingeniería Informática y
Telecomunicaciones

PRÁCTICA 3: BÚSQUEDA CON ADVERSARIO (JUEGOS)

Doble Grado Ingeniería Informática y Matemáticas

Autores:
Javier Gómez López

Junio 2023



Este trabajo se distribuye bajo una licencia CC BY-NC-SA 4.0.

Eres libre de distribuir y adaptar el material siempre que reconozcas a los autores originales del documento, no lo utilices para fines comerciales y lo distribuyas bajo la misma licencia.

creativecommons.org/licenses/by-nc-sa/4.0/

Índice

1. Introducción	3
2. Análisis del problema	3
3. Soluciones planteadas	3
3.1. ScoringV1	3
3.2. ScoringV2	3
3.3. ScoringV3	4
4. Conclusiones	5

1. Introducción

En esta práctica, buscamos implementar alguna de las técnicas con adversario en un entorno de juegos. En nuestro caso, usaremos una versión algo modificada del popular juego Parchís. Esta será determinista y dispondremos de dados especiales que se podrán adquirir conforme nos vamos moviendo por el tablero. Estos dados, inspirados en un popular juego de carreras, darán a las fichas sobre las que se utilicen poderes especiales con los que podrán realizar acciones especiales.

2. Análisis del problema

En nuestro caso, buscamos la implementación de un algoritmo de Poda Alfa-Beta para buscar la mejor jugada posible siguiendo una heurística de diseño propio. Puesto que, dada la complejidad del juego, es computacionalmente imposible generar y manejar el árbol completo de posibilidades en cada instante de la partida, limitaremos la profundidad de nuestro árbol de decisiones a 6 nodos como máximo.

Hay que tener en cuenta que, con las reglas modificadas, habrá que tener en cuenta más factores que los que pudiésemos considerar en una partida de Parchís normal.

En general, a lo largo de las heurísticas que hemos ido planteando he implementando, hemos tenido en cuenta todas las casuísticas que definen una ventaja directa en ese momento de la partida, no aquellas que no sean medibles. Por ejemplo, es deseable tener una casilla en casa, pero no aporta nada tener fichas de un mismo color a una distancia pequeña.

3. Soluciones planteadas

A la hora de afrontar esta práctica, hemos seguido un enfoque progresivo. Puesto que la dificultad de los Ninjas iba en aumento, hemos ido haciendo distintas heurísticas por niveles, siendo la de niveles inferiores parte de las de niveles superiores. Por tanto, hemos ido añadiendo más factores de consideración a nuestra heurística, y en algunas ocasiones hemos modificado algún peso de la versión anterior.

Cabe destacar, que en algún momento un cambio ha supuesto mayor pérdida que beneficio: un cambio que nos hiciese ganar contra el Ninja 2 podía hacernos perder con el Ninja 1, al cual ya habíamos ganado previamente.

Por ello, tras introducir cambios en nuestra heurística, comprobábamos que no nos hacían perder lo ganado con Ninjas anteriores.

3.1. ScoringV1

El objetivo de esta heurística es una toma de contacto, queremos enfrentar el Ninja 1 que es el más sencillo. Por ello, esta heurística solo tiene un solo factor en cuenta: la distancia a la meta de todas mis fichas. Cuanto más lejos estoy de la meta, peor es esa situación para mi ficha. Por tanto, en esta heurística recorro todas mis fichas y asigno el valor al nodo de menos la suma, es decir,

$$p = - \left(\sum_{i=1}^2 \sum_{j=1}^3 d_{ij} \right)$$

donde

- p es la puntuación de mi nodo.
- d_{ij} es la distancia a la meta de la ficha de color i y de id j , con $i = \{1, 2, 3, 4\}$ y $j = \{1, 2, 3\}$.

Las ventajas de esta heurística es su simplicidad y permite hacer un primer acercamiento al problema más general. Además, nos permite ganar al Ninja 1.

Como desventajas, la principal es que no gana al resto de Ninjas. Esto se debe a que es demasiado simple, y no toma en cuenta más factores que si nos deberían de hacer tomar una decisión u otra, como el número de dados especiales y más factores que comentaremos más adelante.

3.2. ScoringV2

Aquí ya las cosas se empiezan a poner interesantes. Poco a poco, nos vamos a introducir en heurísticas algo más complejas. En esta ocasión, además de usar el planteamiento de la versión anterior, consideramos otros

factores: el número de piezas en casa, el número de piezas en la emta y la cantidad de dados especiales que tengamos. Así, nuestra puntuación en esta ocasión sería:

$$p = - \left(\sum_{i=1}^2 \sum_{j=1}^3 d_{ij} * 2 \right) + 100 * s - 200 * c + 90 * g + 50 * sd$$

donde

- p es la puntuación de mi nodo.
- d_{ij} es la distancia a la meta de la ficha de color i y de id j , con $i = \{1, 2, 3, 4\}$ y $j = \{1, 2, 3\}$.
- s es el número de casillas en un lugar seguro.
- c es el número de casillas en casa.
- g es el número de casillas en la meta.
- sd es el número de dados especiales.

Las ventajas de esta heurística es que ya tiene en cuenta un mayor número de factores, lo cual nos permite ganar tanto al Ninja 1 como al Ninja 2. Comentar que los pesos se han hecho de manera totalmente arbitraria y mediante pruebas.

Las desventajas son menores que en la anterior. La principal es que no conseguimos ganar al Ninja 3. Esto se debe a que estamos siendo muy toscos con los pesos. Por ejemplo, valoramos todos los dados especiales por igual, cuando no tienen el mismo efecto. Esto lo arreglaremos con la última y definitiva heurística.

3.3. ScoringV3

Esta es la joya de la corona de esta práctica. Usamos los conceptos que hemos visto que funcionan en la anterior versión y los refinamos, dando un pasito más. En esta ocasión, hacemos un pequeño cambio de concepto: no queremos que ganen mis dos colores, solo queremos uno. Por tanto, calcularemos las puntuaciones de nuestros dos colores por separado, y priorizaremos la que sea mayor en ese momento. Además, cambiamos los pesos que le damos a los dados especiales, siendo este distinto según el tipo de dado especial que estemos teniendo en cuenta. Así, nuestra heurística definitiva se calcula de la siguiente manera:

$$p_i = - \left(\sum_{j=1}^3 d_{ij} \right) + 70 * s_i$$

donde

- p_i es la puntuación asociada al color i .
- d_{ij} es la distancia a la meta de la ficha de color i y de id j , con $i = \{1, 2, 3, 4\}$ y $j = \{1, 2, 3\}$.
- s_i es el número de fichas en un lugar seguro del color i .

y nuestra puntuación definitiva es

$$p = \begin{cases} 2 \cdot \max(p_{i_1}, p_{i_2}) + 0,75 \cdot \min(p_{i_1}, p_{i_2}) + \sum_{t=1}^n \omega(x) \cdot sp_t & \text{si } p_{i_1} \neq p_{i_2} \\ p_{i_1} + p_{i_2} + \sum_{t=1}^n \omega(x) \cdot sp_t & \text{si } p_{i_1} = p_{i_2} \end{cases}$$

donde

- p_{i_j} es la puntuación del color i y como solo tenemos dos colores por jugador, $j = \{1, 2\}$.
- sp_t es el dado especial t , que como solo podemos tener 2, $t = \{1, 2\}$.
- $\omega(x)$ es una aplicación $\omega : \mathbb{N} \rightarrow \mathbb{N}$ es una función peso tal que

$$\begin{aligned} \omega(101) &= 400 & \omega(102) &= 100 & \omega(103) &= 350 & \omega(104) &= 100 \\ \omega(105) &= 100 & \omega(106) &= 150 & \omega(108) &= 140 & \omega(109) &= 50 & \omega(110) &= 10 \end{aligned}$$

Esta heurística presenta un refinamiento respecto a la anterior. Damos una ponderación a los pesos, y beneficiamos al color que vaya mejor de nuestro equipo. Esto es válido para ganar al Ninja 3. Notar que los pesos y las ponderaciones son totalmente arbitrarias, pero funcionan en un entorno real. Los consideramos como números mágicos.

Como desventajas posibles, se podrían tener en cuenta más factores como las barreras, las fichas comidas y otros elementos, pero que no son necesarios para el caso que nos aplica.

4. Conclusiones

Siguiendo con las memorias de prácticas anteriores, quiero resaltar vuestra labor como docentes. La práctica me ha parecido muy instructiva, muy acorde a lo que hemos visto en teoría y en exámenes de prácticas de años anteriores.

De verdad, ojalá todos los profesores de la universidad fuesen como vosotros, porque entonces la UGR sería un referente a nivel mundial. Destacar vuestra dedicación, atención y continua disponibilidad, así como la cercanía con los alumnos.

Muchas gracias por estas prácticas.