

Pau Ramos
Ilarion Tsekot

MiniMassimo

Disseny de l'heurística

Primer de tot, es mira si ha guanyat algun dels dos jugadors, si és així es retorna 1.000.000 o -1.000.000 de puntuació pel tauler depenent de qui guanyi. Si no és el cas, per cada fitxa del nostre jugador a la columna central es sumen 6 punts al tauler, seguidament, s'avalua en tauler mirant les finestres de cada posició en cada direcció (horitzontal, vertical, diagonal positiva i diagonal negativa) i se li suma la puntuació de cada finestra al tauler. Per calcular la puntuació d'una finestra, es compten els colors de cada casella dins l'array passat per paràmetre. Si dins de l'array hi han 4 fitxes del d'un mateix color, voldrà dir que un jugador ja haurà guanyat, si hi han 3 fitxes d'un mateix color i un espai buit, voldrà dir que el jugador s'està apropant a la victòria i per tant se li proporciona una puntuació alta, si hi han 2 fitxes i 2 buits, se li suma poca puntuació perquè s'està anant bé però no assegura la victòria. En canvi si l'oponent té 3 fitxes en una finestra i un espai buit, se li resta puntuació al tauler ja que s'està apropant a la victòria.

```
// Si totes les cel·les són del jugador, atorga la puntuació per guanyar.
if (conteigColor == 4) {
    puntuacio += 100000;
    // Si el jugador té 3 cel·les i una buida, atorga una puntuació alta.
} else if (conteigColor == 3 && conteigBuit == 1) {
    puntuacio += 100;
    // Si el jugador té 2 cel·les i dues buides, atorga una puntuació positiva però menor.
} else if (conteigColor == 2 && conteigBuit == 2) {
    puntuacio += 10;
}

// Disminueix la puntuació si l'oponent té 3 cel·les i una buida.
if (conteigOponent == 3 && conteigBuit == 1) {
    puntuacio -= 80;
}
```

Incidència de la poda alfa-beta

Per implementar la poda, es passa a la funció de minimax un paràmetre alpha i un paràmetre beta. La primera crida es fa inicialitzant alpha al valor més petit possible (Integer.MIN_VALUE) i beta al valor més elevat possible (Integer.MAX_VALUE). A cada crida de la funció, s'actualitza alpha o beta (depenent de si la fase és maximitzadora o minimitzadora) amb el valor heurístic més gran o més petit respectivament. Aleshores, si la beta és inferior a l'alfa i la poda està habilitada, trenca el bucle que genera nous nodes dins de la funció minimax.

```
int valorMoviment = minimax(nouTauler, profunditatMaxima - 1, false, color, Integer.MIN_VALUE, Integer.MAX_VALUE);
```

Crida inicial a la funció minimax.

```
int valor = minimax(nouTauler, profunditat - 1, true, color, alpha, beta);  
// Actualitza el valor mínim trobat fins ara.  
minValor = Math.min(minValor, valor);  
// Actualitza el límit beta (millor opció coneguda per al minimitzador).  
beta = Math.min(beta, valor);  
// Poda alfa: si el valor actual és pitjor que el límit alfa, s'atura l'exploració.  
if (poda && beta <= alpha) {  
    break; // Poda alfa  
}
```

Fragment de codi dins del bucle per crear moviments a la part minimitzadora.

```
int valor = minimax(nouTauler, profunditat - 1, false, color, alpha, beta);  
maxValor = Math.max(maxValor, valor); // Actualitza el valor màxim trobat fins ara. // Actualitza el límit alfa (millor opció coneguda per al maximitzador).  
alpha = Math.max(alpha, valor);  
// Poda beta: si el valor actual és millor que el límit beta, s'atura l'exploració.  
if (poda && beta <= alpha) {  
    break; // Poda beta  
}
```

Fragment de codi dins del bucle per crear moviments a la part maximitzadora.

Profunditat màxima	Nodes totals explorats	
	Sense poda	Amb poda
2	1.404	1.404
4	13.881	9.864
6	579.766	353.850
8	8.117.269	3.704.233

Avaluat contra el professor amb heurística false, sent jugador 1 i amb la mateixa profunditat màxima que el MiniMassimo.

La taula anterior mostra l'efecte de la poda sobre el nombre de nodes explorats en total durant l'execució d'un joc MiniMax, comparat amb el mateix joc sense cap poda. S'evidencia que, a mesura que augmenta la profunditat de cerca, l'efecte de la poda és més visible, especialment a les profunditats de cerca més altes com 6 i 8, que és quan la diferència en el nombre de nodes explorats es fa notòria. Aquesta optimització permet enfocar els recursos computacionals als nodes rellevants, mantenint la qualitat de la decisió final.

Incidència de l'ordre d'exploració dels fills en la poda alfa-beta

La ordenació que s'utilitza a l'algorime prioritza els moviments realitzats a la columna central ja que així són els que s'avaluen primer i afavoreix l'aparició de bons resultats a la heurística (així es poden més nodes).

Profunditat màxima	Nodes totals explorats			
	Sense poda		Amb poda	
	Ordenant	Sense ordenar	Ordenant	Sense ordenar
2	1.404	928	1.404	928
4	13.881	17.182	9.864	12.758
6	579.766	896.120	353.850	475.151
8	8.117.269	16.224.123	3.704.233	8.678.066

Avaluat contra el professor amb heurística false, sent jugador 1 i amb la mateixa profunditat màxima que el MiniMassimo.

Resultats

A continuació, una taula on s'exposen els resultats contra el MiniMax del professor amb les seves diferents heurístiques.

Profunditat màxima	MiniMassimo com a jugador 1		MiniMassimo com a jugador 2	
	H. Prof. 1 (true)	H. Prof. 2 (false)	H. Prof. 1 (true)	H. Prof. 2 (false)
2	Guanya	Guanya	Guanya	Perd
4	Guanya	Guanya	Guanya	Guanya
6	Guanya	Guanya	Guanya	Guanya
8	Guanya	Guanya	Guanya	Guanya