Algoritmo MiniMax com podas α-β utilizado no Gomoku

Fabio Moreira Marcello Klingelfus Junior

22 de Agosto de 2017

- **Heurística e utilidade**

  A função heurística $H(T)$ deste trabalho pode ser definida como o somatório de todos os encadeamentos ao quadrado que pertecem ao computador, multiplicado por um fator de bloqueio de jogada (detalhado adiante), subtraindo o somatório de todos os encadeamentos ao quadrado do adversário. A Equação 1 define matematicamente essa função.

$$H(T) = \left(\sum_{i=1}^{N} (EC)^2 \right) * TEB - \sum_{j=1}^{N} (EA)^2 \tag{1}$$

  Onde:

  - T: É uma determinada configuração do tabuleiro de *gomoku*
  - EC: Encadeamentos do computador
  - EA: Encadeamentos do adversário
  - TEB: Total de encadeamentos bloqueados, definido como a soma das peças bloqueadas do adversário ao adicionar uma peça pertencente ao computador.
  - N: Total de encadeamentos do computador/adversário

  Em uma partida normal, o adversário e o computador adicionam peças ao tabuleiro até que ou alguém faça uma sequência de cinco peças ou o tabuleiro seja totalmente preenchido, resultando em um empate. Para que se consiga atingir o objetivo do jogo, é necessário que as peças sejam agrupadas até que formem uma sequência de cinco peças. Para isso, é essencial que o computador entenda que peças agrupadas valem mais que peças "soltas" pelo tabuleiro. O objetivo de elevar os encadeamentos ao quadrado é justamente dizer ao computador que quanto maior o encadeamento, maiores serão as chances de atingir o objetivo buscado. O computador não joga sozinho, assim, também

é fundamental que ele compreenda que quanto mais encadeamentos grandes o adversário possuir, menores são as chances do computador de atingir seu objetivo, esse é o propósito da segunda parte da equação. O problema com essa heurística é que o adversário pode enganar o computador. Basta adicionar dois encadeamentos na mesma direção deixando um "buraco" entre eles. Com isso, o computador não verá essa jogada como uma ameaça e dará mais valor para outras jogadas, o fator TEB corrige essa deficiência.

A função de utilidade $U(T)$ pode ser definida matematicamente através da Equação 2.

$$H(T) = \left(\sum_{i=1}^{N}(EC)^2\right) * TEB - \left(\sum_{j=1}^{N}(EA)^2\right) * FJ \tag{2}$$

O que difere as equações é o fator FJ (Fim de Jogo). Ao detectar o fim de jogo, seu valor é alterado de 1 para 100.

- **Otimizações e estratégias**

  O tabuleiro foi projetado como uma matriz de tamanho 15×15. As peças são objetos que possuem informações sobre: proprietário (computador, adversário ou vazio); seus vizinhos imediatos na vertical, horizontal e diagonal e; sua posição no tabuleiro. Há também duas listas que guardam todas as peças utilizadas pelo computador e adversário. Com esses dados, é possível otimizar o tempo de processamento evitando percorrer toda a matriz sempre que precisar recuperar informações sobre encadeamentos. Para o algoritmo MiniMax, a árvore de estados será construída recursivamente ao longo da execução desse algoritmo. O tamanho do tabuleiro pode comprometer o desempenho do algoritmo. Assim, para a construção dos possíveis estados, será considerado um pedaço desse tabuleiro. Se o adversário adicionar uma peça fora desse espaço, o algoritmo passará a considerar um espaço que englobe essa peça.

- **Detecção de fim de jogo e sequência de quatro peças**

  Ambas as detecções são feitas por um mesmo algoritmo. O algoritmo seqN (presente no código fonte) busca, a partir de um ponto aleatório, todos os encadeamentos presentes no jogo relacionados a um único jogador. Ao invés de percorrer todo o tabuleiro, o algoritmo vai "empilhando" peças na mesma direção. Quando o encadeamento chega ao fim, sua profundidade é verificada e o encadeamento é guardado em uma lista. A posição da lista indica o tamanho do encadeamento, assim, ao encontrar uma profundidade de tamanho 3, o valor presente na posição 3 da lista é incrementado. Ao término do algoritmo, é só acessar essas posições e descobrir se há uma sequência de cinco peças indicando fim de jogo ou acessar as outras posições da lista para executar a heurística.