

LICENCIATURA EM ENGENHARIA INFORMÁTICA
INTRODUÇÃO À INTELIGÊNCIA ARTIFICIAL
AULAS TEÓRICAS: EXERCÍCIOS

FICHA VI B: ALGORITMOS PARA JOGOS - CONTINUAÇÃO

Soluções

1.

a) Cada peça que se deixa cair só pode ser colocada em 4 posições distintas porque, devido ao facto do tabuleiro ser vertical, apenas pode ficar numa das 4 colunas por cima das peças que já lá estão (as posições "no meio" do tabuleiro não podem ser atingidas enquanto não houver peças por baixo). Portanto, o factor de ramificação é um pouco inferior a 4 (seria exactamente 4 se não fossem as 3 jogadas finais, quando há menos de 4 posições livres.).

b) Na 1ª jogada pode escolher-se uma de 4 posições. Na 2ª também. Na 3ª também... Há sempre 4 jogadas possíveis (por isso é que o factor de ramificação é 4). Como há 16 jogadas, teremos cerca de 4^{16} nós.

c) Estrutura estado
tabuleiro [4,4] // Os elementos desta matriz serão 0 para casas vazias, 1
 // para peças do jogador 1 e 2 para peças do jogador 2

d) Há 4 operadores possíveis:

Empilhar uma nova peça na coluna 1
Empilhar uma nova peça na coluna 2
Empilhar uma nova peça na coluna 3
Empilhar uma nova peça na coluna 4

Qualquer deles deve receber como parâmetro o jogador que faz a jogada, de modo a empilhar a peça de código (1 ou 2) correspondente. Qualquer deles só é aplicável se o número total de peças na respectiva coluna for <4 .

e) Cada operador pode ser aplicado se o número de peças na respectiva coluna for inferior a 4.

f) De acordo com b) o limite superior para o número de estados é $4^{16}=4.294.967.296$. A geração de todos eles pelo MiniMax não é possível dado os limites reais de memória e tempo de processamento previsível. Sendo assim recorrer-se-ia ao *Alpha-Beta Pruning* com limite de profundidade sendo então necessária uma função de avaliação de cada estado

terminal gerado em cada expansão até uma profundidade pré-definida. Funções possíveis seriam poderia do tipo:

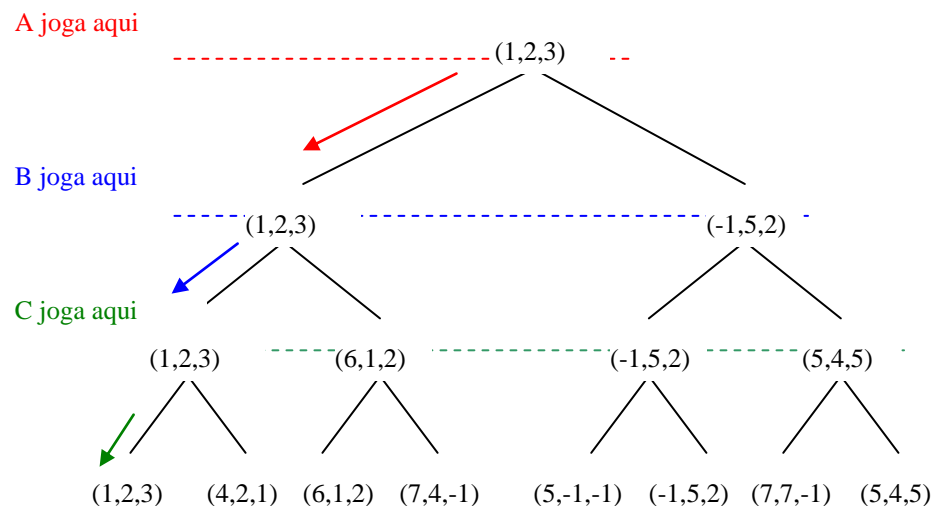
(nº de peças em linha e coluna de MAX - nº de peças em linha e coluna para MIN)

ou

(nº de linhas+colunas+diagonais ainda possíveis para MAX - nº de linhas+colunas+diagonais ainda possíveis para MIN)

Para saber o nº de linhas, colunas ou diagonais ainda possíveis para MAX ou MIN, basta ver se essas linhas, colunas e diagonais ainda não contêm nenhum símbolo do oponente (i.e., para MAX, ver se não têm símbolos de MIN, e vice-versa)

2. a) A árvore completa-se atendendo a que cada jogador A, B, ou C, escolherá, no nó em que é ele a decidir, a jogada que lhe proporcionar maior pontuação. Portanto, começando pelos nós terminais e progredindo em direcção à raiz obtemos:

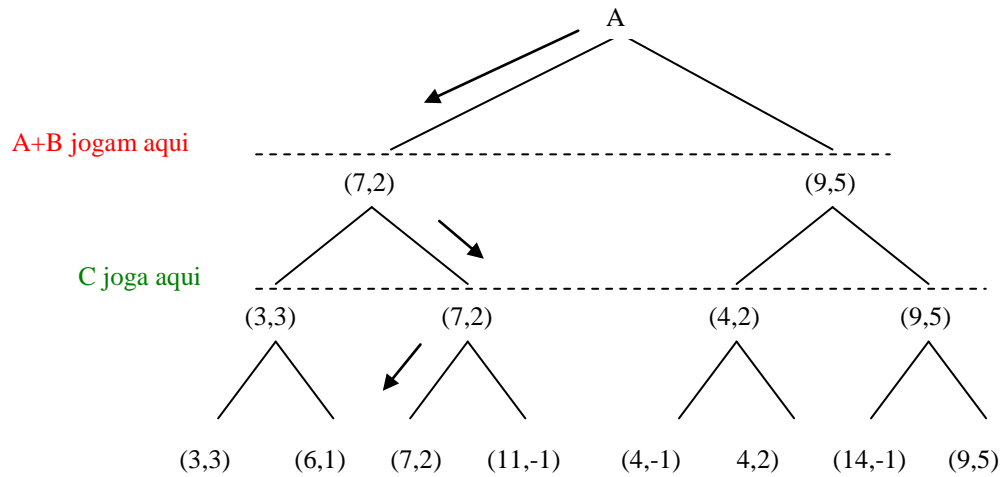


b) O jogador A opta pela jogada que o conduz ao nó de valorização (1,2,3) (assinalada na figura)

c) C ganha o jogo com pontuação 3 (ver figura).

d) Se A e B formam uma aliança, uma interpretação possível é que então quando forem A ou B a jogar deverão optar pelo nó para o qual a diferença $(A+B)-C$ for maior (função de avaliação). Quando for C a jogar, deverá optar pelos nós em que $C-(A+B)$ for maior. A valorização nas folhas deverá ser substituída pela soma de $(A+B, C)$:

A+B jogam aqui -----



Se se considerar (apenas) que o resultado é tanto melhor quanto maior a pontuação de A+B, independentemente da de C, (ou de C, independentemente da de A+B) então a sequência será a da figura seguinte:

