Relatório do Projeto de Laboratórios de Algoritmia I - LCC

Jogo Reversi

Ivo Lima – A90214

Mário Silva – A55884

Índice

Introdução

Num jogo clássico de Reversi ou Othello cada jogador tem como principal objetivo chegar a uma situação de fim de jogo com um número de peças na sua posse superior ao número de peças na posse do adversário. O momento de fim de jogo é estabelecido quando não existirem jogadas válidas para ambos os jogadores ou quando o tabuleiro estiver completamente preenchido, vencendo o jogo quem possuir mais peças em cima do mesmo. Assim, um bom jogador de Reversi deve formalizar estratégias e métodos de criar desvantagem ao seu adversário ao longo de todo o jogo, de modo a consagrar-se vencedor.

Estratégias

O desenvolvimento de um programa (bot) capaz de reagir e formalizar jogadas automaticamente num jogo como o Reversi obriga à implementação de estratégias que vão de encontro ao objetivo do programa (jogador virtual) – se ele é um jogador, tem como objetivo vencer.

Assim, utilizamos no nosso projeto uma função clássica de jogos de tabuleiro, a Alpha-Beta Pruning, que é uma variante de outra função clássica deste tipo de jogos, a MiniMax. Esta última caracteriza-se por realizar uma busca (exaustiva, como uma árvore de casos) de todos os estados (finais ou até determinado número de jogadas consecutivas - nível) de jogo possíveis, devolvendo um valor heurístico (quanto maior, mais preciosa é a jogada). Tendo em conta que o bot tem de maximizar sempre o seu valor heurístico, e o adversário irá minimizá-lo, compara-se os valores heurísticos de todas as jogadas possíveis e determina-se qual o melhor movimento para o bot.

A variante Alpha-Beta Pruning é uma otimização da função MiniMax, uma vez que converte uma busca exaustiva numa busca inteligente, sendo capaz de determinar quando não há necessidade de procurar em zonas da árvore de casos onde não se irá encontrar a solução.

A segunda estratégia utilizada foi a análise de posições de risco/valiosas do tabuleiro, atribuindo-lhes valores que são decrementados/ incrementados à heurística de jogadas realizadas nessas posições. Existem determinadas regiões do tabuleiro que são cruciais para obter uma boa pontuação no fim do jogo, permitindo controlar o decorrer do jogo. Essas posições são os cantos e as bordas do tabuleiro, com exceção das posições das bordas que dão acesso aos cantos. Por outro lado, existem posições do tabuleiro que são de risco, uma vez que dão acesso às posições vantajosas, sendo que uma jogada numa dessas posições pode comprometer o jogo para quem cometer esse pé em falso.

Implementação

Alpha-Beta Pruning

O número de níveis da árvore avaliados, em cada nível do bot, é dado por (NívelBot x (NívelBot - 1)) + 1 vezes.

Bot Nível 1

Para um nível mais básico, as ramificações exploradas na árvore de casos irão garantir que o bot compare os resultados de todos os movimentos que possa realizar naquele estado do tabuleiro, no sentido de tentar colocar-se em vantagem. Assim, apenas é avaliado um nível de ramificações ((1x (1 - 1) + 1) = 1), isto é, uma jogada (a do próprio bot). Neste nível as potencialidades do algoritmo não são efetivamente aproveitadas, uma vez que o primeiro nível da árvore é sempre totalmente avaliado.

Nível 2

Neste nível é feita uma busca em 3 níveis da árvore ((2x (2 - 1) + 1) = 3), ou seja, 3 jogadas (bot, adversário, bot). Neste nível as potencialidades do algoritmo já se traduzem na diminuição do esforço de computação.

Nível 3

Neste nível é feita uma busca em 7 níveis da árvore ((3x (3 - 1) + 1) = 7), ou seja, 7 jogadas (bot, adversário, bot, adversário, bot adversário, bot). Neste nível o algoritmo permite antecipar um número significativo de jogadas à frente com um tempo de computação reduzido, permitindo jogar fluentemente contra o computador.

Análise de posições de risco/valiosas

O tabuleiro está dividido por várias regiões consoante Figura 1. Para cada posição, o valor da heurística de jogar nessa posição é alterado consoante o nível do bot, como demonstrado na Tabela 1.

A utilização desta estratégia vai depender do número de casa vazias no tabuleiro, pois numa situação em que o tabuleiro tem poucos lugares vazios, o algoritmo Alpha-Beta Pruning é capaz de calcular todas as jogadas até à situação de fim de jogo. Assim, a análise de posições de risco/valiosas só é realizada se o número de casas vazias for superior ao número de jogadas que serão avaliadas pelo algoritmo Alpha-Beta Pruning.

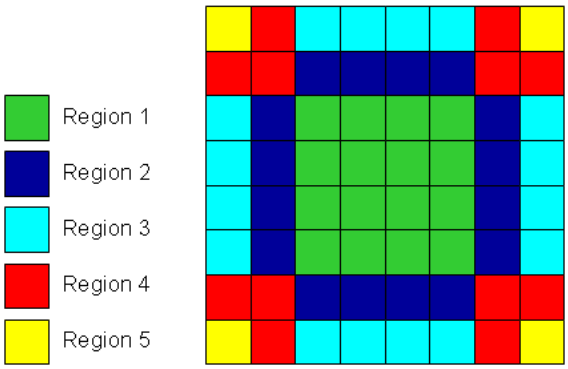

Figura 1 - Regiões do tabuleiro de Reversi. *(Retirado de http://mnemstudio.org/game-reversi-example-2.htm)*

Nível Bot	Região 1 (sem alteração)	Região 2 - (NxN)	Região 3 + (NxN)	Região 4 - (NxNx2)	Região 5 + (NxNx2)
1	+ 0	- 1	+ 1	- 2	+ 2
2	+ 0	- 4	+ 4	- 8	+ 8
3	+ 0	- 9	+ 9	- 18	+ 18

Tabela 1 - Alterações à heurística consoante região do tabuleiro e nível do bot (N = nível do bot).