



Eletiva - Inteligência Artificial

Projeto de Agente para Atuar em Ambiente Multi-agente e Competitivo

São Paulo, 25 de outubro de 2022

Gustavo Marques Borges

Leonardo Fernandes Farah Elias

Rafael Oncins

Sumário

1) Objetivo do Projeto.....	3
2) Restrições do Projeto	3
3) Perguntas do Projeto	3
4) Referencias.....	5

1) Objetivo do Projeto

O objetivo do projeto é desenvolver um agente jogador de uma variante de Liga4 chamada *Connect4 PopOut*. O tabuleiro desta variante tem as mesmas dimensões que o Liga4 convencional, as peças continuam sendo as mesmas e o objetivo também. No entanto, durante cada jogada, o jogador pode ou adicionar um novo disco no topo ou remover um disco seu da base (o que chamamos de *pop out*). Ao fazer isto, todos os discos que estão na mesma coluna que o disco que foi removido irão descer uma linha, mudando a sua relação com os demais discos presentes no tabuleiro.

2) Restrições do Projeto

Ao implementar este agente você deve considerar algumas restrições:

- A duração de nenhum movimento deverá ultrapassar 10 segundos.
- O seu agente não deverá fazer nenhuma jogada inválida:
 - jogar um disco em uma coluna que não existe;
 - remover um disco que não existe;
 - remover um disco do oponente, ou;
 - jogar um disco em uma coluna que já está completa.
- O seu agente não pode perder nenhuma partida para o jogador aleatório.

Se o seu agente realizar qualquer um destes movimentos durante a competição então ele será desclassificado da competição.

3) Perguntas do Projeto

1. Que algoritmo deve ser utilizado para desenvolver um agente jogador de *Connect4 PopOut* vencedor? Deve-se utilizar uma implementação de Min-Max com poda alpha-beta? Se sim, qual a profundidade que deverá ser utilizada para evitar processamentos superiores a 5 segundos por jogada? Qual a função de utilidade que deve ser utilizada?

No código desenvolvido foi implementado o algoritmo Min-Max com poda alpha-beta, uma vez que esse algoritmo “poda” a árvore de busca, contribuindo para limitar as opções de tomada de decisão, tornando-as mais objetivas. Além disso, pode ser utilizada uma profundidade de até 7 níveis. A função utilidade que foi implementada é essencialmente estabelecer uma pontuação para a quantidade de fichas consecutivas ponderando um valor para um conjunto de 3, 2 ou 1 ficha. Também incluímos na pontuação o posicionamento das fichas do nosso jogador ao compará-las com um mapa (matriz) desenhado com o intuito de forçar o código a seguir a estratégia de dominar o centro. O valor final da nossa função utilidade é calculado como a diferença da pontuação obtida pelo nosso player e pelo adversário somado a pontuação obtida pelo posicionamento de nossas peças segundo ao mapa.

2. O seu jogador faz uso de alguma base de conhecimento? Se sim, como ela é utilizada durante o processo de tomada de decisão?

A única base de conhecimento utilizada pelo nosso player é o mapa criado para implementar a estratégia de dominar o centro. Ela é utilizada dentro da nossa função utilidade como citado na questão anterior. No entanto, ela tem um grande valor no início do jogo visto que enquanto não se tem peças alinhadas um mapa é o único método de garantir a pontuação. Portanto, ele guia o começo do jogo, implementando assim, a estratégia de dominar o centro.

3. Qual a sua expectativa com relação ao desempenho do seu agente? Você acredita que ele irá desempenhar bem na competição? Por que? Você executou testes contra outros jogadores? Qual foram os resultados?

Acredito que o desempenho de nosso agente irá ser adequado, justamente por termos testado com outros jogadores. Nós testamos nosso player contra o RandomPlayer, obtendo resultados positivos, no entanto ao colocá-lo contra o BarthPlayer (adaptado para jogar no modo pop out) obtivemos resultados negativos quando utilizamos profundidades maiores e chegamos a um empate utilizando a mesma profundidade. O desempenho que foi observado mostra que as estratégias implementadas foram simples em prol de manter o tempo de processamento adequando a competição, no entanto isso pode ser a causa de na competição não ter uma performance excelente.

4. Quais foram as principais referências utilizadas para a implementação do seu jogador?

Além dos materiais de aula foi feito uma pesquisa de funções utilidades para implementar, no entanto decidimos por utilizar uma função muito similar à do BarthPlayer, somente alterando a forma que a pontuação foi ponderada segundo o estudo da faculdade de Cornell ([link](#))

5. Existem diferenças significativas entre um jogador de *Connect4* e um jogador de *Connect4 PopOut* em termos de árvore de busca e função de avaliação? É possível utilizar o jogador implementado para o *Connect4 PopOut* em competições de *Connect4* sem muitas modificações?

Sim existem, já que em um alguns caso (como por exemplo em que todas as fichas da última linha forem suas) serão adicionados mais 7 nodos a arvore de busca. Como falamos anteriormente nós conseguimos adaptar facilmente o BarthPlayer para competir com jogadores de Pop Out, isso só mudando uma linha do código original.

4) Referencias

<http://www.cs.cornell.edu/boom/2001sp/Anvari/Anvari.htm>

https://en.wikipedia.org/wiki/Connect_Four

https://en.wikipedia.org/wiki/Evaluation_function

<https://stockfishchess.org/blog/2020/introducing-nnue-evaluation/>