Implementação de uma Inteligência Artificial jogadora de Shogi através de Busca Competitiva

Eduardo Blüthgen Garcia

[1]Departamento de Informática – Universidade Estadual de Maringá (UEM)
Maringá – PR – Brazil

ra99164@uem.br

Abstract. *This report is about the implementation of a shogi playing Artificial Intelligence via Adversarial Search through the Minimax algorithm with the use of Alfa-Beta Pruning. Six different value estimation strategies were compared, and the Minimax was tested against random and greedy foes, proving itself as the winner in all tests.*

Resumo. *Este relatório trata da implementação de uma Inteligência Artificial jogadora de shogi utilizando-se Busca Competitiva através do algoritmo Minimax e Poda Alfa-Beta. Seis heurísticas de avaliação do valor de um estado foram comparadas, e o Minimax foi testado contra oponentes aleatórios e gulosos, se provando o ganhador em todos os testes.*

1. Descrição do Problema

Neste trabalho, para se implementar uma Inteligência Artificial agente capaz de jogar um jogo, e o escolhido foi o *Shogi*.

1.1. *Shogi*

O ***Shogi***, também conhecido como **Jogo dos Generais** ou **Xadrez Japonês**, é acreditado ser uma evolução do *chaturanga* indiano, assim como o xadrez europeu, e, assim como este, é um jogo de tabuleiro *zero-sum*, mas apresenta complexidade computacional muito maior, a maior complexidade de sua classe de jogo, devido a não apenas seu tabuleiro maior que o do xadrez e número maior de peças, mas principalmente ao fato de que peças capturadas não são removidas do jogo, podendo ser devolvidas ao tabuleiro. Outra diferença em relação ao xadrez nascida de sua evolução paralela é no nível de poder de cada peça, sendo que no xadrez as peças são em gerais mais poderosas, o que causa um jogo mais defensivo, ao oposto do *shogi*, que é muito mais agressivo devido ao menor poder de movimento de cada peça: perder uma peça é menos impactante e, portanto, jogadas arriscadas são mais incentivadas.

2. Plataforma

Os experimentos descritos neste relatório foram realizados em um *laptop* com processador Intel Core i5 de sétima geração, com 8GB de memória DDR4, e 2TB de disco, com sistema operacional Windows 10. O algoritmo foi implementado na linguagem de programação *Python* versão 3.7.

3. Busca Competitiva

A técnica de agentes jogadores Busca Competitiva é baseada na natureza de oposição de muitos jogos, nos quais os objetivos de cada jogador leva a frustração dos objetivos do oponente. A Busca Competitiva tenta escolher a melhor ação a ser feita considerando as ações futuras do oponente, assumindo que este sempre atuará de maneira ótima, sendo portanto uma estratégia de contingência, de minimizar os riscos e perdas. O algoritmo utilizado, o *minimax*, adota um esquema de pontuação de cada estado do jogo para definir os objetivos de cada jogador, com o primeiro jogador, Max, objetivando a maior pontuação possível, e o segundo, Min, a menor. Como jogos tendem a causar árvores de decisão enormes, é normalmente empregado um algoritmo de estimativa para decidir o valor dos estados intermediários, permitindo a limitação da profundidade máxima a ser explorada.

3.1. Implementação

Para este trabalho, foram implementados seis estratégias de estimativa diferente, ou três pares nas quais a primeira das duas considera o valor de todas as peças de cada jogador e a segunda promove a devolução de peças ao tabuleiro considerando apenas metade do valor das peças fora do tabuleiro. Para o primeiro par, cada peça vale um ponto. Para o segundo, são utilizados os valores baseados nos oficiais, com algumas peças mais importantes valendo dois pontos. Para o terceiro, foi decidido para cada peça um valor arbitrário baseado em sua utilidade no tabuleiro. Após alguns testes, foi constatada a necessidade da limitação do tempo de cada jogada, sendo utilizado o limite de 25 minutos.

4. Resultados

Para se avaliar o algoritmo implementado, este jogou partidas do jogo contra alguns oponentes: o próprio minimax, um agente aleatório, e um agente guloso se valendo das mesmas 6 estratégias de estimativa de valor. A execução do minimax contra si mesmo só foi possível com a árvore de decisão estando em uma ordem aleatória, demorando em cada uma das quatro execuções 123, 225, 50 e 128 turnos, levando respectivamente 01h20min, 09h42min, 40min e 13h27min, com ambos os jogadores ganhando duas vezes cada. Quando as árvores de busca estavam na mesma ordem, o tempo de execução foi demais para o prazo de entrega deste trabalho, e foi cancelado após dois dias de execução.

Nos demais casos, o algoritmo implementado ganhou todas as partidas. Com oponente aleatório, o tempo de execução e número de turnos foi variável, porém rápido, levando 20, 26, 52 e 50 turnos, ou 02min, 16min, 2h48min e 09min, respectivamente.

Para as execuções do oponente guloso, foram realizados diversos jogos, cujos resultados podem ser observados em gráficos.

Os primeiros dois gráficos representam os resultados das execuções do algoritmo minimax com cada uma das 6 estratégias contra um algoritmo guloso seguindo a estratégia 1. O primeiro é em relação ao número de turnos e o segundo o tempo em horas consumido.

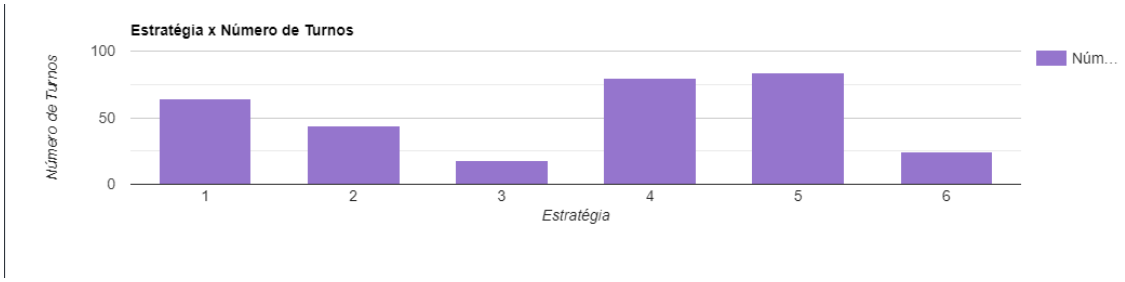

Figura 1. Turnos gastos contra algoritmo guloso com Estratégia 1

Figura 2. Tempo gasto contra algoritmo guloso com Estratégia 1

Os dois próximos gráficos representam os resultados das execuções do algoritmo minimax com cada uma das 6 estratégias contra um algoritmo guloso seguindo a mesma estratégia que o minimax. O primeiro é em relação ao número de turnos e o segundo o tempo em horas consumido. Nota-se que a Estratégia 1 é dramaticamente pior que as outras em relação ao tempo gasto.

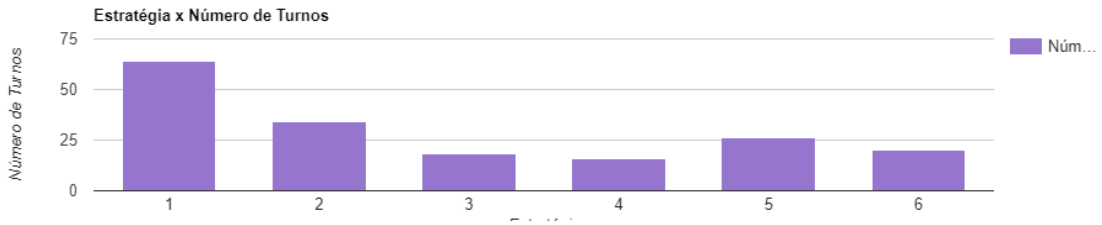

Figura 3. Turnos gastos contra algoritmo guloso com a mesma Estratégia

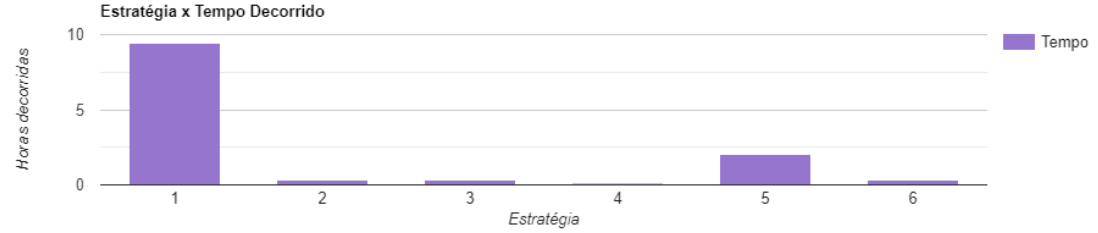

Figura 4. Tempo gasto contra algoritmo guloso com a mesma Estratégia

Em todos os gráficos é visível que as Estratégias pares, ou seja, as que consideram apenas metade dos pontos das peças fora do tabuleiro, apresentam resultado melhor que sua versão que considera o valor das peças igualmente, com a exceção da Estratégia 4 que foi pior que a 3 contra um oponente guloso que segue a Estratégia 1. É visível também que em todos os gráficos, as estratégias 3 e 6 se provam superiores às demais, alcançando a vitória com poucas ações e em pouco tempo.

5. Conclusão

O *shogi* é um jogo de alta complexidade computacional, e por isso é difícil de se implementar uma Inteligência Artificial que faça decisões de maneira rápida. A IA implementada através do minimax se provou superior às outras técnicas, em especial quando utilizando-se as heurísticas 3, inspirada pela pontuação oficial de cada peça, e a 6, utilizando uma pontuação arbitrária para representar o valor de cada peça, e incentivando se retornar peças ao tabuleiro. O tempo de espera por cada decisão ainda é relativamente alto, mas considerando a dificuldade do problema, uma IA que se prova inequívocamente capaz de ganhar de outros agentes mais simples é um sucesso.

Referências

Japan shogi league. `http://www.shogi.or.jp/`. Accessed on 2019-04-26.

Rules and manner of shogi. `http://81dojo.com/documents/Rules_and_Manners_of_Shogi`. Accessed on 2019-04-21.