

Eduardo
Blüthgen
Garcia

Ciência da Computação
Inteligência Artificial I - Professor
Wagner Igarashi
Universidade Estadual de Maringá

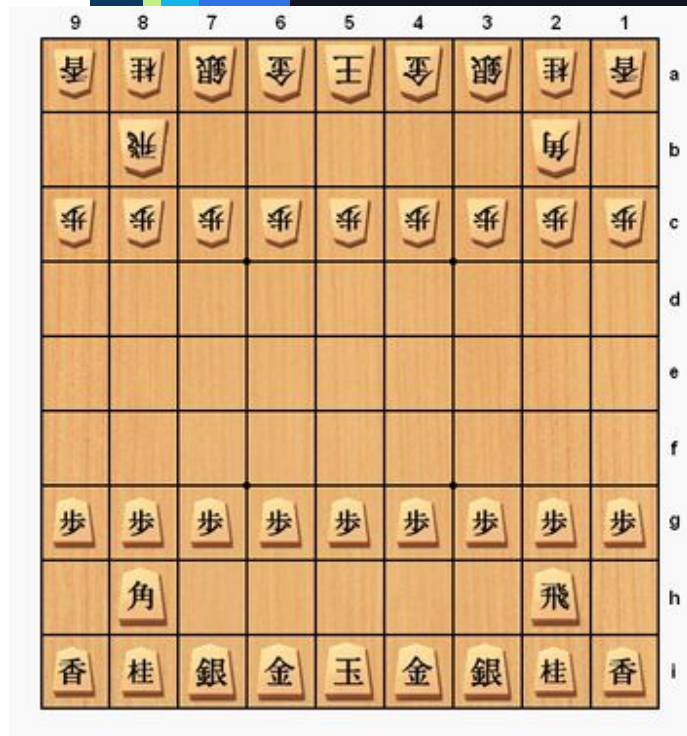
Trabalho de Inteligência Artificial

Implementação de um
agente jogador de *Shogi*

将棋 - *Shogi*

O *Shogi*, também conhecido como **Jogo dos Generais** ou **Xadrez Japonês**, é acreditado ser uma evolução do chaturanga indiano original, assim como o xadrez europeu.

Embora a base das regras e jogabilidade seja a mesma do xadrez, o jogo apresenta diversas diferenças, como por exemplo o maior tamanho do tabuleiro (9x9 ao invés de 8x8) e o número maior de peças (40 ao invés de 32)



Diferenças do Xadrez

O xadrez e o *shogi* evoluíram de maneira oposta:

No xadrez cada peça individual é mais poderosa e o jogo acaba sendo mais defensivo por isso, e no *shogi*, cada peça é menos poderosa e o jogo é mais agressivo.

Outra diferença é a promoção: no xadrez, apenas os Peões são capazes de serem promovidos, e somente ao alcançarem a última linha, e no *shogi* a maioria das peças é capaz de ser promovida em qualquer uma das três últimas linhas inimigas.

Peças capturadas

No *Shogi*, peças capturadas não são removidas do jogo, mas vão para a reserva do jogador que as capturou e podem ser devolvidas ao tabuleiro.

Uma peça pode ser colocada no tabuleiro em sua forma não promovida em qualquer espaço livre, com duas exceções para os peões: Não se pode colocar um peão em uma coluna na qual o jogador já possua outro peão, e não se pode colocá-los de forma que possam capturar o rei inimigo no próximo turno.

Empate

Existem duas formas de empate no jogo:

Sennichite, que se refere ao jogo estar em um *loop*, ou seja, o jogo entrou exatamente no mesmo estado quatro vezes. Quando isto é detectado, ocorre um empate.

Impasse, que ocorre quando os jogadores concordam em encerrar o jogo, e então são contados os pontos de cada jogador com base em suas peças. Caso nenhum dos jogadores obtenha pelo menos 24 pontos, ocorre um empate.

Plataforma

Os experimentos foram realizados em um laptop com processador Intel Core i5 de sétima geração, com 8GB de memória RAM DDR4 e 2TB de disco, em um sistema operacional Windows 10.

O algoritmo foi implementado na linguagem de programação *Python*.

Busca Competitiva

A Busca Competitiva é uma técnica de Inteligência Artificial que se baseia na existência de outros agentes no ambiente, e especificamente agentes que agem como oponentes da IA.

Os “oponentes” tentam ativamente impedir ou frustrar os objetivos, e por isso é interessante se utilizar uma estratégia de contingência, ou seja, tentando minimizar as perdas e não maximizar os ganhos.

Algoritmo Minimax

O algoritmo Minimax é uma aplicação da Busca Competitiva, que explora uma árvore de estados na qual a cada nível se alterna qual dos jogadores está agindo, ou seja, para se selecionar qual ação será realizada considera-se que o oponente sempre agirá de maneira ótima, escolhendo a ação de menor valor possível.

Para a aplicação do algoritmo é necessário uma situação *zero-sum*, na qual existe um ganhador e um perdedor.

Poda *alfa-beta*

Para otimizar o algoritmo Minimax, é importante se diminuir o tamanho da árvore de busca através de uma poda, e a poda *alfa-beta* tenta identificar quando um “galho” da árvore vai possuir um valor obsoleto em relação a outro já explorado. Assim, não é necessário se explorar os outros filhos deste nó.

Durante o turno de “max”, após se explorar todos os filhos do primeiro nó, caso algum filho do segundo nó possuir valor menor que qualquer filho do anterior, já se sabe que este nó não será escolhido por Max.

Limitações de tempo

Mesmo aplicando-se a poda *alfa-beta*, o algoritmo ainda pode demorar consideravelmente para uma decisão caso a árvore seja grande o suficiente.

Uma possibilidade para a redução do tempo é se estabelecer um limite de profundidade máxima a ser explorada, o que necessita da utilização de uma heurística para se estimar o valor dos estados não finais da árvore.

Heurísticas de Estimativa

Neste trabalho, foram implementadas 6 heurísticas, baseadas em se dar um valor para cada peça, tanto no tabuleiro quanto na reserva, e subtrair o total do oponente da soma, incentivando a captura de peças inimigas.

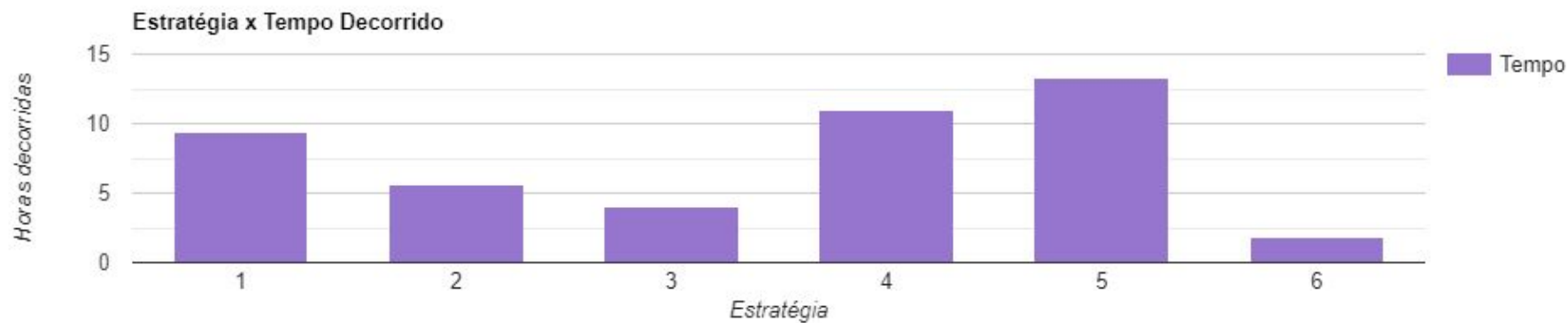
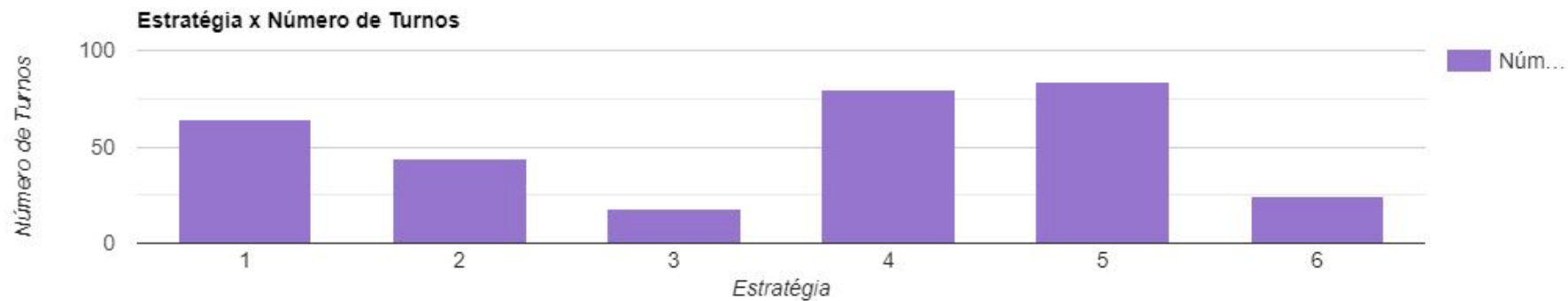
As heurísticas formam três pares nos quais a diferença entre elas é apenas que a segunda incentiva a se devolver peças ao tabuleiro ao se dividir o valor de peças fora do tabuleiro pela metade.

Testes realizados

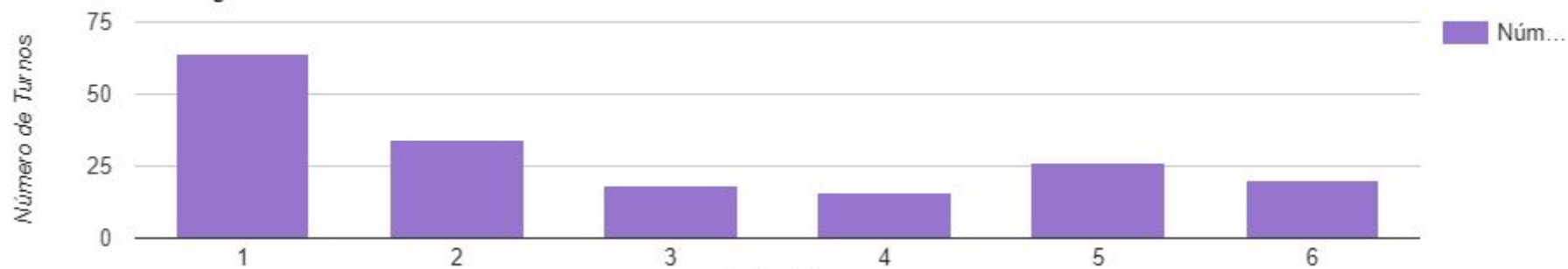
O algoritmo minimax implementado foi testado contra outro jogador minimax, contra um jogador de movimentos aleatórios e com jogadores gulosos utilizando as 6 heurísticas utilizadas.

No primeiro caso, a execução foi cancelada após dois dias de execução.

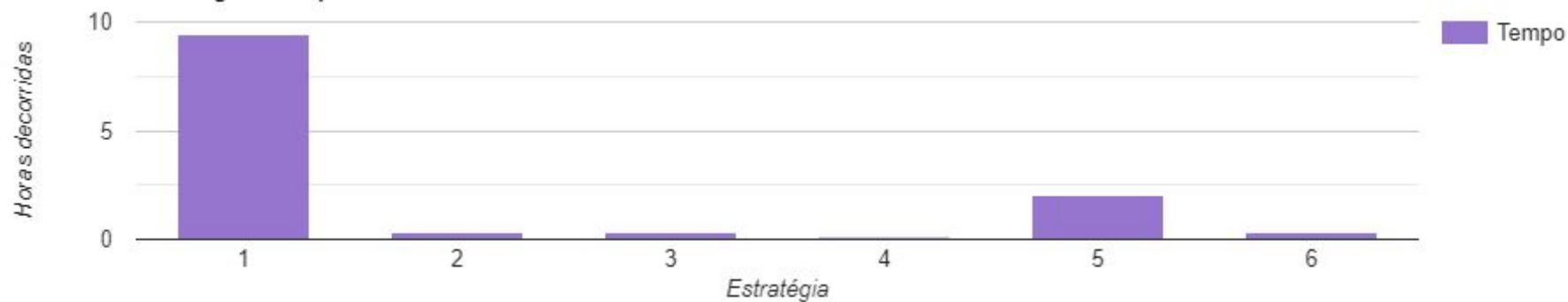
Contra o jogador aleatório, o agente minimax obteve vitórias comparativamente rápidas, levando em média 37 turnos (desvio padrão de 14 e variância de 201), ou 15 minutos e 36 segundos.



Estratégia x Número de Turnos



Estratégia x Tempo Decorrido



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.

Referências

Materiais de aula

<https://kartikkukreja.wordpress.com/2015/07/04/adversarial-search/>

http://www.cs.cornell.edu/courses/cs4700/2011fa/lectures/06_AdversarialSearch.pdf

https://courses.edx.org/asset-v1:ColumbiaX+CSMM.101x+1T2017+type@asset+block@AI_edx_adversarial_search_small_size.pdf