

Minimax no Jogo da Velha

Lucas da Silva
Colégio Salesiano Itajaí
Itajaí, SC
lucsvieirabr@gmail.com

Rodrigo Lyra
Universidade do Vale de Itajaí
Itajaí, SC
rlyra@univali.br

Valdir José Corrêa
Colégio Salesiano Itajaí
Itajaí, SC
vavaprof.english@gmail.com

Alisson Steffens Henrique
Colégio Salesiano Itajaí
Itajaí, SC
ali.steffens@gmail.com

ABSTRACT

In game theory, Minimax is an algorithm that aims to solve decision making, based on a mathematical method that aims to minimize the possible maximum loss, also known as maximizing the minimum gain. The present work was developed as a research project and investigation of algorithms in an itinerary of computational thinking aimed at high school students. The research presents a customized implementation of the algorithm in a “tic-tac-toe”, where the Minimax algorithm is responsible for the AI moves that oppose the player. The implementation was done in JavaScript (WEB) and tests were performed with different equations in the definition of the nodes' score. With the tests, it was possible to perceive which steps and situations most impact the quality of the moves, as well as validate different parameters for the algorithm depending on the desired degree of difficulty. In the end, tests were also carried out in matches against real players, and it was possible to perceive that Minimax can provide different strategies and levels of difficulty in the tic-tac-toe game.

KEYWORDS

Tic-Tac-Toe, Artificial Intelligence, Minimax algorithm, Game theory, Depth-First Search

1 Introdução

A Teoria dos Jogos é uma teoria matemática cuja intenção é modelar fenômenos observáveis quando dois ou mais agentes de decisão interagem entre si e são concorrentes [1], como em algoritmos genéticos, redes neurais e jogos digitais. Desenvolvida por John von Neumann, a Teoria dos Jogos é amplamente discutida em diversos trabalhos [2, 9], tendo como base as árvores de decisões como representação da tabela de decisão.

Este trabalho propõe a utilização da Teoria de Decisão Algorítmica Minimax [6] para operar uma inteligência artificial (IA), simulando a capacidade humana de raciocinar, tomar decisões e resolver problemas [8] no Jogo da Velha, isto com a implementação de um algoritmo de busca em profundidade, ou DFS (Depth First Search) [7], juntamente com uma árvore de decisão, contendo todas as jogadas possíveis que a IA puder tomar. O melhor movimento será escolhido com base em uma pontuação atribuída a cada nó da árvore [8].

Neste trabalho é utilizada uma versão modificada do Minimax, que permite escolher uma determinada quantidade de níveis a serem procurados, mudando assim a qualidade da resposta encontrada pela rede.

2 Regras do Jogo da Velha

O “Jogo da Velha”, também chamado de Tic Tac Toe (TTT) é um jogo simples que possui como objetivo fazer combinações triplas na posição horizontal, vertical ou diagonal em um tabuleiro de 3x3 (matriz de três linhas por três colunas) [8].

Os jogadores assumem símbolos que os representam para realizar suas jogadas no tabuleiro em espaços vazios, alternando entre si, podendo ser o xis (X) ou o círculo (O). No caso da aplicação proposta neste trabalho, a IA será o jogador 2, que utiliza o círculo. Desta forma, sempre será a segunda a jogar. Se os dois jogadores sempre agirem da mesma forma ou possuírem conhecimento o bastante acerca do jogo, este sempre terminará em empate [10], como é possível observar em vários resultados obtidos durante os experimentos finais.

3 Teoria dos Jogos e Minimax

A Teoria dos Jogos é um ramo da Matemática Aplicada que estuda situações estratégicas, em que jogadores concorrentes escolhem diferentes ações na tentativa de maximizar seu ganho e/ou minimizar o ganho do adversário [2, 9].

Esta teoria é aplicável na análise de fenômenos sociais, econômicos e em jogos. É definida, também, como um conjunto de técnicas analíticas, destinadas a auxiliar na compreensão e análise de fenômenos observáveis [3].

Von Neumann [4] prova que todo jogo com dois agentes de decisão possui uma solução em estratégias mistas. Utilizado como um método para minimizar a perda máxima possível, o Minimax pode ser considerado como a maximização do ganho mínimo [8]. Em sua aplicação no Jogo da Velha, proposta neste trabalho, o algoritmo auxilia a encontrar a melhor jogada ao caminhar pelas opções válidas a partir do tabuleiro fornecido, considerando uma disputa entre dois jogadores, em que um tenta maximizar o ganho, enquanto o outro busca minimizar as chances de vitória do primeiro [8].

Comumente, o algoritmo Minimax utiliza-se de um algoritmo de busca que realiza uma varredura, percorrendo todos os vértices

XIV Computer on the Beach

30 de Março a 01 de Abril de 2023, Florianópolis, SC, Brasil

da árvore de decisão – representação de uma tabela de decisão sob a forma de árvore, permitindo a análise de condições complexas utilizando-se da lógica, sendo assim, um meio eficiente de construir classificadores que predigam classes baseadas nos valores de atributos de um conjunto de dados [14] -. A busca em profundidade (DFS), começa a análise em um nó raiz e explora tanto quanto possível cada um dos seus ramos, antes de retroceder ao início [8]. Este algoritmo foi utilizado pois move-se nos nodos de maneira vertical, o que facilita a soma das pontuações de cada nodo pai.

A Figura 1 demonstra a combinação da busca DFS com um algoritmo Minimax, percorrendo todos os movimentos possíveis a partir do estado inicial do jogo fornecido, atribuindo-os uma pontuação.

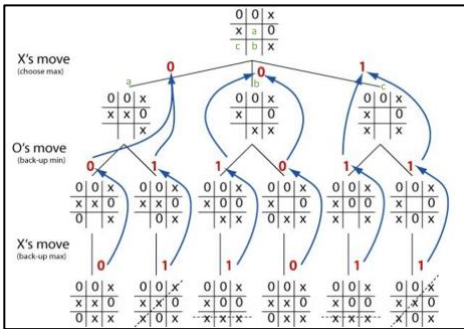


Figura 1: Busca em Profundidade e algoritmo Minimax [12]

Na Figura 1 é possível perceber que no primeiro nível há três possíveis jogadas para X, sendo que cada uma ainda abre mais duas situações para cada jogada de O. Ao observar o último nível de análise, é visto que X vence em 50% das situações jogando a ou b. Entretanto, para X em c gera-se uma situação em que independentemente da próxima jogada de O, X consegue vencer. Sendo esta então, sim, a melhor opção de jogada.

4 Implementação

A aplicação da teoria de decisão algorítmica Minimax para operar uma IA no TTT demonstrada neste trabalho foi feita utilizando a linguagem de programação interpretada estruturada JavaScript (JS), e com a linguagem de marcação HTML e CSS para a construção da página. Para esta implementação as linguagens citadas foram escolhidas por permitirem o lançamento da aplicação na web, o que permite que mais pessoas a utilizem.

O tabuleiro do TTT é de 3x3; logo, inicialmente possui apenas 9 (nove) lacunas para realizar as jogadas; assim, o espaço de varredura para a busca DFS é pequeno, permitindo a implementação de um algoritmo perfeito, não heurístico, já que não há limitação do seu espaço de busca. Ao longo de sua construção, diversos problemas surgiram, oriundos de um sistema de pontuação de movimentos e retorno ao nó não eficiente, e principalmente da limitação do espaço de varredura.

Inicialmente, fora proposta para a pontuação dos tabuleiros o seguinte esquema: vitória da IA (1), vitória do jogador (-1) e empate (0). A fim de permitir que o algoritmo pudesse ser utilizado com diferentes níveis de profundidade (sem necessariamente ter

que percorrer todas as probabilidades até a vitória), optou-se por um método alternativo de cálculo. Neste, ao invés de utilizar apenas os indicadores de vitória e derrota, é calculada uma média das pontuações obtidas até o nível investigado.

A Figura 2 apresenta uma falha detectada nesta abordagem.

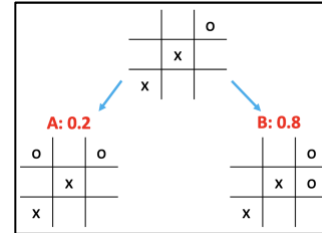


Figura 2: Falha no algoritmo de pontuação

É possível perceber o problema citado ao analisar a pontuação do movimento “A” e “B”, em que a “B” possui uma maior pontuação, algo que não deveria acontecer já que a “A” impede o jogador de realizar uma jogada que o levará a dois caminhos de vitória, forçando-o a bloquear a IA, caso contrário resultará em sua derrota.

Assim, para resolver este problema, quatro modificações foram realizadas, sendo elas: (i) não limitar a varredura da busca DFS e (ii) retornar assim que houver uma vitória ou derrota, (iii) novo sistema de pontuação, (iv) atribuir uma pontuação logo no início caso haja uma vitória ou derrota (respectivamente ∞ e $-\infty$). A Equação 1 demonstra o novo sistema de pontuação implementado.

$$\frac{\text{Sum}}{\text{Depth}} \rightarrow \begin{cases} \text{"O" wins} = 1 \\ \text{"X" wins} = -3 \\ \text{"Tie"} = 0.2 \end{cases}$$

Equação 1: Novo sistema de pontuação e fórmula

A Equação 1 funciona com a razão da soma da pontuação de cada movimento do ramo analisado (sum) e a profundidade que o algoritmo alcançou na árvore (depth).

5 Experimentos

Para a criação do algoritmo, diversos testes foram realizados, e ao longo destes foram avaliadas várias equações com resultados ruins, dentre todas, destacam-se os erros demonstrados na Figura 3a e 3b (com a IA jogando como O em ambas as situações).

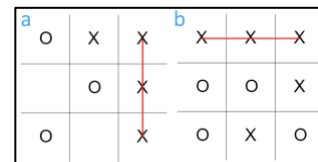


Figura 3: Exemplo de erro no algoritmo

Inicialmente, é evidente que a origem do erro demonstrado na Figura 3a refere-se a uma priorização do ganho máximo em detrimento do ganho mínimo do adversário, isto é, o algoritmo

XIV Computer on the Beach

30 de Março a 01 de Abril de 2023, Florianópolis, SC, Brasil

sobrepe a vitória da IA ao invés de impedir efetivamente a do jogador.

Já a versão demonstrada na Figura 3b é oriunda de um algoritmo totalmente falho, que basicamente realiza jogadas em qualquer lacuna livre. A fim de otimizar o algoritmo de busca DFS realizaram-se experimentos com versões heurísticas, limitando a profundidade da varredura, o que implicou diretamente na qualidade dos primeiros movimentos, resultando na derrota da IA. Com a análise dos resultados obtidos foi percebida a possibilidade da criação de níveis de dificuldade, com a implementação deste sistema heurístico. Nas Figuras 4 e 5 são observados 4 exemplos dessa implementação.

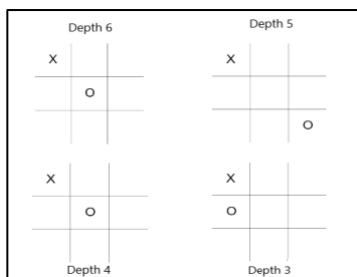


Figura 4: Movimento inicial em relação a profundidade

Ao analisar a Figura 4 pode-se perceber que a alteração proposta modifica o movimento inicial da IA.

Na Figura 5 percebemos uma intensificação desta alteração ao verificar a reação da IA perante uma jogada completa.

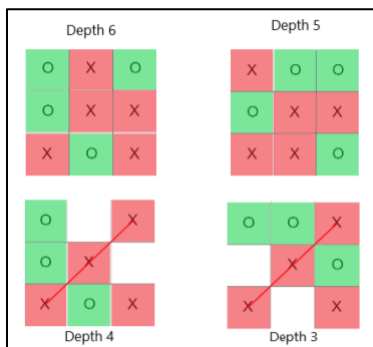


Figura 5: Reação da IA com determinado movimento

Percebe-se que o algoritmo com limitação de profundidade 6 e 5 consegue bloquear o ataque, resultando em empate. Já na limitação 4 e 3 de profundidade a IA é derrotada, e em todos podemos perceber que ela reage de forma diferente ao mesmo movimento.

Com o intuito de melhorar a eficiência do algoritmo, propõe-se um teste dos futuros movimentos do jogador logo após escolher a melhor jogada, caso essa não resulte na vitória da IA; assim, se o próximo movimento do jogador resultar em sua vitória, altera-se o movimento da IA por esse. Assim como podemos analisar no fluxograma representado na Figura 6.

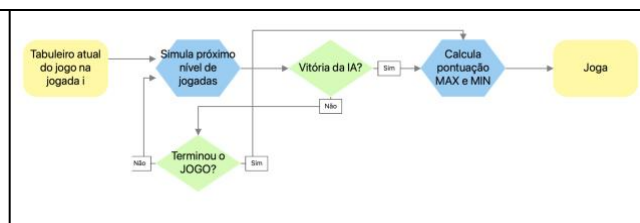


Figura 6: Fluxograma do algoritmo

Com a implementação demonstrada na Figura 6, o algoritmo consegue bloquear os movimentos futuros do jogador de forma eficiente, minimizando as chances de vitória do adversário. Ao fazer projeções das futuras jogadas do oponente após a escolha de seu movimento, a IA consegue reavaliar sua decisão e priorizar o ganho mínimo do adversário, caso a sua jogada escolhida não a leve à vitória. Assim, com uma simulação das futuras jogadas após o seu movimento, citada anteriormente, a IA consegue eliminar as chances de ser derrotada. Podemos perceber as vantagens desta implementação com a comparação demonstrada na Figura 7 ao analisar a reação do algoritmo antes e depois respectivamente, perante o mesmo movimento.

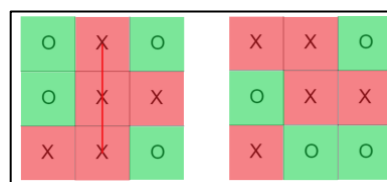


Figura 7: Antes e Depois da implementação proposta

Ao aplicar e regular a implementação citada acima, percebe-se uma melhora nos movimentos da IA, evidentemente representada na Figura 7. Entretanto, tal alteração resultou no aumento do tempo de processamento do algoritmo ao escolher o primeiro movimento. Vale ressaltar, que este aumento é insignificante, mas perceptível em certas máquinas. Desta forma, uma possível otimização da busca foi considerada. Todavia, resultou em uma piora significativa dos movimentos da IA. Assim, foi decidido por manter essa alteração.

Com essa aplicação, a IA consegue prever certos movimentos futuros do jogador, podendo assim anular as possibilidades da mesma se encontrar em uma “espécie de xeque-mate” em que, independentemente da jogada realizada resultará em sua derrota, tais cenários citados podem ser percebidos na Figura 8 e, vale ressaltar, com essa nova mudança e o impedimento do ganho máximo do jogador, estas situações não ocorrem mais.

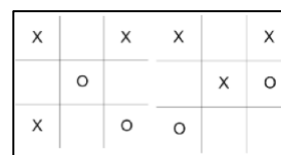


Figura 8: Exemplos de Xeque-mate

XIV Computer on the Beach

30 de Março a 01 de Abril de 2023, Florianópolis, SC, Brasil

Na Figura 8 é evidente duas formas de Xeque-mate que a IA sofria antes da mudança em seu algoritmo. Nos dois exemplos acima, a próxima jogada da IA se torna irrelevante, já que o jogador Xis (X) possui duas maneiras distintas de obter sua vitória, assim percebemos a necessidade da alteração proposta para anular os movimentos defeituosos que originam este cenário de derrota iminente.

6 Resultados

Com a definição de uma equação eficaz para a pontuação das jogadas, espera-se que as partidas terminem sempre em empate - com o jogador realizando o movimento inicial- como consta-se no Quadro 1.

Jogador	Empates	Vitórias IA	Vitórias Jogador
1	3	0	0
2	3	0	0
3	2	1	0
4	3	0	0
5	3	0	0
6	2	1	0
7	3	0	0
8	3	0	0
9	3	0	0
10	2	1	0

Quadro 1: Resultados das partidas

Realizando uma análise gráfica dos resultados obtidos demonstrada na Figura 9, há uma taxa de 10% de vitória da IA, proveniente possivelmente de uma distração dos jogadores, já que o esperado é sempre resultar em empate.

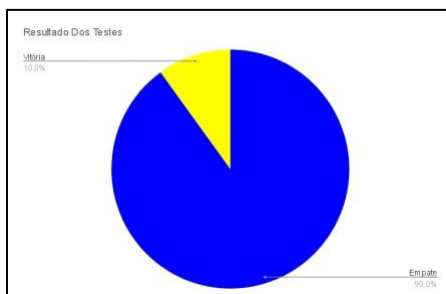


Figura 9: Desempenho da IA nas partidas

Experimentos acerca do tempo de resposta da IA perante o primeiro movimento do jogador, demonstram um tempo de processamento aproximado de 1 segundo, variando em relação ao dispositivo utilizado.

Vários testes foram realizados em busca de reduzir esse tempo, entretanto nenhum deles se mostram realmente efetivos, e como tal valor é relativo ao dispositivo utilizado e se aplica apenas ao movimento inicial - todos os próximos movimentos da IA são instantâneos – optou-se por manter desta forma.

7 Conclusão

Ao longo dos experimentos demonstrados acerca da aplicação da teoria de decisão algorítmica - Minimax- proposta por John von Neumann no Jogo da Velha, percebe-se que a IA nunca é derrotada, e que o resultado esperado com o jogador iniciando é sempre empate. Além disso, destaca-se que ela obteve uma porcentagem de 90% de empate e 10% de vitória.

Tais resultados são esperados com a aplicação desse algoritmo para operar uma IA no TTT. Os resultados apresentados por Sant'ana (2014) [13], se assemelham aos demonstrados no decorrer deste artigo, muito embora a proporção seja deveras diferente e, é importante destacar que o experimento de Sant'ana (2014) foi realizado contra outra IA que realizava movimentos aleatórios.

Em 50.000 jogos, o jogador virtual implementado por Sant'ana (2014) venceu cerca de 91% das vezes e empatou 9% das vezes [13], concluindo assim, que o esperado da IA ao jogar contra um algoritmo que realize movimentos aleatórios é de vitória, e esses valores tendem a modificar grandemente quando o adversário é um humano, entretanto, em ambos os casos a porcentagem de vitória do adversário se mantém nula.

Sobre outro aspecto, o jogador virtual criado por Ávila [15], demonstra resultados semelhantes aos apresentados na Figura 9, levando em conta o mesmo cenário de humano contra IA em que o humano realiza a primeira jogada. Assim conclui-se que o resultado esperado quando o jogador inicia é de empate e, em casos de falha do jogador, a sua derrota, como demonstrado na Figura 9.

Em todo o processo de implementação deste algoritmo surgiram inúmeras dificuldades como: a criação de uma fórmula efetiva para a análise e pontuação dos ramos de possibilidades, otimização do algoritmo e reformulação do código de uma maneira que fossem possíveis futuras análises e modificações na equação. Durante o processo de implementação foram utilizadas e testadas diversas fórmulas como as citadas acima, e a cada nova modificação, testes com jogadores voluntários, analisando a resposta da IA para deferentes movimentos.

Ademais, vale destacar a contribuição deste projeto para a comunidade de aprendizado. Durante três dias, este trabalho fora apresentado em um estande e avaliado por uma banca na Feirinha Científica e Cultural do Colégio Salesiano de Itajaí, promovendo e estimulando a pesquisa científica no meio tecnológico em diversos estudantes e professores, além de gerar uma proximidade das IAs com o ambiente de ensino, fomentando a curiosidade deles acerca ao tema.

Por fim, conclui-se que o teorema matemático Minimax aplicado em uma IA no TTT, mostra-se extremamente efetivo, obtendo resultados muito satisfatórios, fornecendo uma dificuldade um pouco maior ao jogo simples que é o Jogo da Velha.

Referências

- [1] Roseli Rocha Teixeira. 2018. Situações de conflito: uma aplicação da teoria dos jogos em sala de aula.
- [2] David Jonnes Francez. 2017. Uma introdução à teoria dos jogos.
- [3] Eduardo Botti Abbade. 2009. Aplicação da teoria dos jogos na análise de alianças estratégicas.
- [4] J. von Neumann. Zur Theorie der Gesellschaftsspiele. Mathematische Annalen, vol. 100, pp. 295-320. Traduzido por S. Bargmann: On the Theory of Games of Strategy

XIV Computer on the Beach

30 de Março a 01 de Abril de 2023, Florianópolis, SC, Brasil

-
- em Contributions to the Theory of Games, vol. 4, pp. 13-42, A. W. Tucker e R. D. Luce (editores), Princeton University Press, 1959.
- [5] E. Zermelo, Über eine Anwendung der Mengenlehre auf die theorie des Schachspiels. Atas do D'écimo Quinto Congresso Internacional de Matemáticos, vol. 2, pp. 501-504, 1913.
 - [6] H. Ben-El-Mechaiekh, R. Dimand, Von Neumann, Ville, and the Minimax Theorem, STOREP Conference on the History of Decision Theory, Siena, June 2005.
 - [7] Gerard J. Holzmann, Doron Peled, and Mihalis Yannakakis. 1997. On Nested Depth First Search.
 - [8] Aluísio de Ávila, Angélica Caetane Pelizza, Bruno Barbosa Prettz, Gabriel Camara Mandeli, Renan Rosauro Rigon, Fábio José Parreira, Sidnei Renato Silveir. 2016. Simulação do Jogo Tic-Tac-Toe com o uso de Técnicas de Inteligência Artificial.
 - [9] J. von Neumann, Oskar Morgenstern. 1944. Theory of Games and Economic Behavior.
 - [10] ALVES, W. (2015) Jogos de Tabuleiro
 - [11] Miguel López Mamani. 2020. DFS VS BFS.
 - [12] Elnaggar, Ahmed & Gadallah, Mahmoud & Mostafa, Mostafa & Eldeeb, Hesham. (2014). A Comparative Study of Game Tree Searching Methods. International Journal of Advanced Computer Science and Applications. 5. 68-77. 10.14569/IJACSA.2014.050510.
 - [13] SANT'ANA, P. (2014). Inteligência Artificial para o Jogo da Velha. Acesso sem setembro de 2022.
 - [14] Marcelo Hiroshi Shiba 1 Rosangela Leal Santos 2, 3 José Alberto Quintanilha 2 Hae Yong Kim 1. (2005). Classificação de imagens de sensoriamento remoto pela aprendizagem por árvore de decisão: uma avaliação de desempenho. Anais XII Simpósio Brasileiro de Sensoriamento Remoto, Goiânia, Brasil
 - [15] Aluísio de Ávila, Angélica Caetane Pelizza, Bruno Barbosa Prettz, Gabriel Camara Mandeli, Renan Rosauro Rigon, Fábio José Parreira, Sidnei Renato Silveira. (2016). Simulação do Jogo Tic-Tac-Toe com o uso de Técnicas de Inteligência Artificial.