

Uma Análise Comparativa de Funções de Utilidade para o Algoritmo Minimax Aplicado ao Jogo da Onça

Felipe G. L. do Nascimento, Elloá B. Guedes

¹Grupo de Pesquisa em Sistemas Inteligentes
Laboratório de Sistemas Inteligentes
Escola Superior de Tecnologia
Universidade do Estado do Amazonas
Av. Darcy Vargas, 1200 – Manaus – Amazonas

{fgldn.eng, ebgcosta}@uea.edu.br

Resumo. *O Jogo da Onça é um legado cultural indígena brasileiro, caracterizado por um jogo de tabuleiro assimétrico no qual uma onça enfrenta catorze cachorros. Com vistas a desenvolver um jogador automático e inteligente para um desses personagens, este trabalho tem por objetivo analisar comparativamente diferentes funções de utilidade no contexto da busca adversarial com o algoritmo Minimax. Os resultados preliminares mostram a dificuldade na concepção destas funções e sugerem haver um personagem com mais probabilidade de vitórias que seu oponente.*

Abstract. *The Jaguar Game is a Brazilian indigenous cultural legacy characterized by an asymmetric board game in which a jaguar duels with fourteen dogs. Aiming at developing an automatic intelligent player for one of this characters, this work intends to compare different utility functions in an adversarial search context using the Minimax algorithm. Preliminary results reveal challenges upon conceiving such functions and suggest that one of the players has a higher victory probability than its opponent.*

1. Introdução

Com o advento da Computação, muitas estratégias ligadas à Inteligência Artificial (IA) passaram a ser desenvolvidas para jogos, permitindo que o computador viesse a ser um adversário notável em certas modalidades. Para os pesquisadores da IA, este domínio é interessante, pois: (i) sua natureza abstrata facilita o estudo; (ii) existe uma severa dificuldade de resolução deste tipo de problema; (iii) eventualmente há a impossibilidade de tomar decisões ótimas; e (iv) as más estratégias são severamente punidas [Russel e Norvig 2016].

Os primeiros relatos da utilização de técnicas da IA em jogos remontam ao ano de 1979. Desde então, diversas técnicas têm sido elaboradas e aplicadas em jogos visando a construção de algoritmos que simulem jogadores humanos, promovendo a competitividade. Algumas soluções, inclusive, fornecem ao computador características sobre-humanas, resultado do altíssimo poder de processamento computacional e do desenvolvimento de estratégias que permitem a análise das consequências muitíssimas jogadas à frente [Millington e Funge 2009]. Os mais notórios progressos da IA aplicadas aos jogos estão no âmbito de Xadrez e Go [Hsu 2002, Chen 2016].

Na tradição indígena brasileira, entretanto, há um jogo de tabuleiro, denominado “Jogo da Onça”, para o qual jogadores automáticos e inteligentes ainda não foram propostos na literatura, o que motivou a realização deste trabalho. Este jogo possui algumas particularidades muito específicas, tais como o fato de haver um número desigual de peças, havendo catorze cachorros versus uma onça, um tabuleiro assimétrico e tipos de jogadas e objetivos diferentes para os adversários. Considerando os desafios mencionados, este trabalho almeja propor, avaliar e analisar a proposição de diferentes funções de utilidade para os personagens do Jogo da Onça. Para tanto, considera a exploração da árvore de jogo segundo o algoritmo Minimax, uma estratégia tipicamente adotada para o desenvolvimento de jogadores automáticos em cenários de busca adversarial.

Para apresentar os resultados obtidos, este trabalho está organizado como segue. A fundamentação teórica encontra-se descrita na Seção 2, abordando os conceitos relativos ao Jogo da Onça e ao algoritmo Minimax. Os materiais e métodos utilizados neste trabalho são descritos na Seção 3. Os resultados e a discussão são apresentados na Seção 4. Por fim, na Seção 5, são postas as considerações finais e as perspectivas de trabalhos futuros.

2. Fundamentação Teórica

Nesta seção serão apresentados dois conceitos centrais ao entendimento deste trabalho: o Jogo da Onça, suas características e regras, e o algoritmo Minimax.

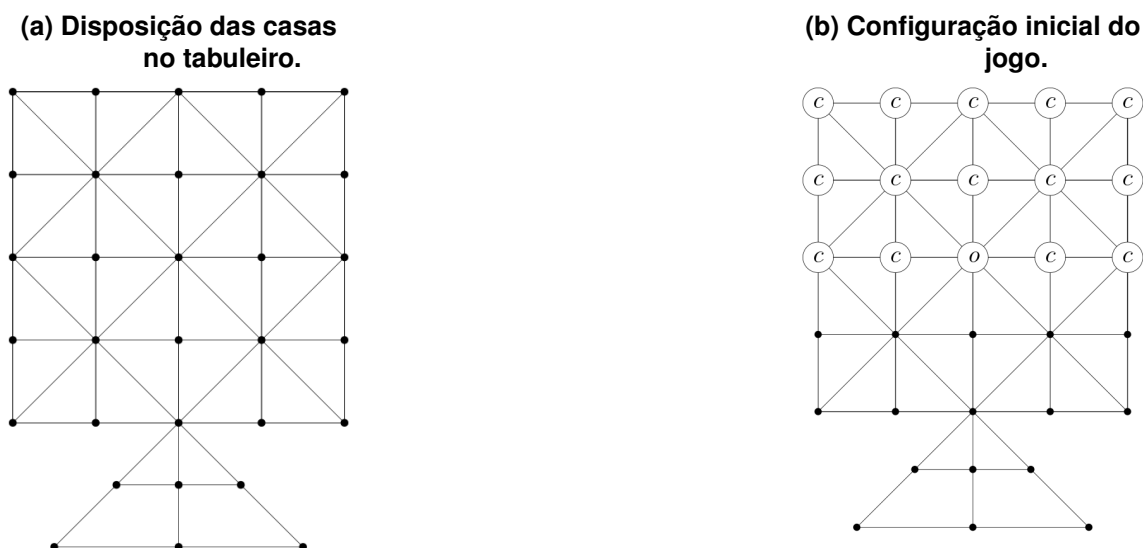
2.1. Jogo da Onça

O chamado *Jogo da Onça* é um jogo de tabuleiro da cultura indígena brasileira em que há dois tipos de personagens: onça e cachorro. A onça, representada por uma única peça, movimenta-se de maneira similar aos elementos do popular Jogo de Damas, podendo saltar sobre os cachorros a fim de comê-los e, portanto, eliminá-los. Um total de catorze cachorros, o outro tipo de personagem, deve atuar de maneira coletiva com vistas a imobilizar a onça. Vence aquele que primeiramente alcançar seu objetivo [Calderaro 2006].

Na tradição indígena, o tabuleiro é desenhado na areia batida e pedras ou grãos atuam na representação das peças do jogo. O tabuleiro é composto por figuras geométricas simples, como quadrados e triângulos, dispostos conforme ilustrado na Figura 1a. As interseções, chamadas de vértices e representadas como círculos na referida figura, representam as casas do tabuleiro, que é composto por 31 destas. Cada casa possui duas ou mais casas adjacentes. Uma casa é dita ser adjacente a outra quando há uma aresta que as conecta. O triângulo que se externa do quadrado na parte inferior é chamado de “toca da onça”.

A disposição inicial das peças no tabuleiro é mostrada na Figura 1b. A partir desta configuração, o jogador com a onça faz o primeiro movimento e então os jogadores se alternam a cada turno. As peças existentes, quer seja onça ou cachorro, podem se movimentar para quaisquer casas adjacentes do tabuleiro, desde que estas estejam vazias, uma casa por vez. A onça captura um cachorro de maneira análoga ao Jogo de Damas, fazendo um salto por cima do cachorro para uma casa vazia adjacente. Quando possível, também pode capturar vários cachorros de uma só vez com saltos duplos ou triplos, com a diferença que neste cenário tal feito não é de natureza obrigatória. Esta personagem vence quando consegue capturar cinco cachorros. O cachorro, por sua vez, não possui nenhum

Figura 1: Representações do tabuleiro do Jogo da Onça. A letra 'c' denota o cachorro e a letra 'o' denota a onça.



movimento de ataque, porém se faz em maior quantidade no tabuleiro. O jogador com os cachorros não pode capturar a onça, apenas mover as peças de forma a encurralá-la, deixando-a sem possibilidade de movimentos [Secretaria Municipal de Educação 2016].

Além das vitórias da onça ou do cachorro, é possível haver empates. Estes ocorrem quando, durante uma partida, for comprovado que uma posição se repetiu quatro vezes. É o chamado *empate pela repetição de lances*, que deve ser reclamado pelo jogador no momento em que ocorrer [Secretaria Municipal de Educação 2016]. O empate ocorre quando há falta de combatividade entre os jogadores, situação não desejada no contexto de um jogo entre adversários.

Este jogo tem por objetivo proporcionar, de maneira lúdica, o desenvolvimento do raciocínio lógico, dedutivo e a criação de estratégias. Cita-se também a colaboração no desenvolvimento de aspectos sociais, como a convivência em coletividade e a elaboração de inúmeras estratégias de sobrevivência, aprendizados essenciais para a vida adulta [Bettin e Pretto 2016]. Uma perspectiva também enfatizada por outros autores que ressaltam a relevância deste jogo diz respeito à *etnomatemática*, ou seja, ao aprendizado não-formal de conceitos matemáticos, incluindo neste caso, por exemplo, as figuras geométricas existentes no tabuleiro [de Oliveira Sardinha e Gaspar 2010].

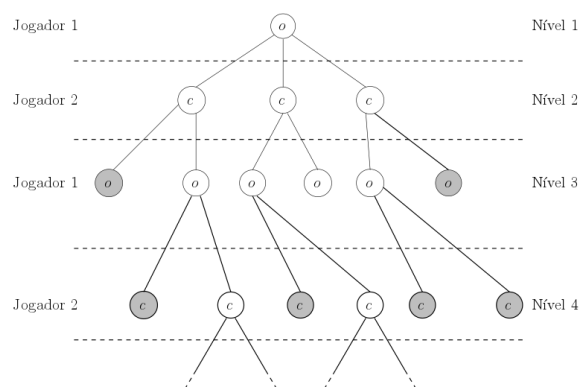
2.2. Algoritmo Minimax

Na âmbito da Teoria dos Jogos, os tipos de jogos de tabuleiro variam bastante quanto ao *número de jogadores*, *objetivos* de cada jogador, quanto à *informação* que cada jogador tem sobre o jogo e quanto aos *algoritmos* de IA que podem ser aplicados ao mesmo e que fazem com que um computador simule um dos jogadores [Millington e Funge 2009]. Um dos algoritmos de IA mais comumente aplicados a jogos de Tabuleiro é o Minimax [Millington e Funge 2009].

O Minimax atua efetuando uma busca na *árvore de jogo* – uma árvore em que os nós são estados de jogo e as arestas são possíveis movimentos do jogador da vez, para

aquele estado. Um *estado* de jogo caracteriza-se por uma disposição ou configuração de peças no tabuleiro [Russel e Norvig 2016], como o estado inicial do Jogo da Onça, apresentado anteriormente na Figura 1b. Uma árvore de jogo completa contém todos os estados possíveis de um jogo. A Figura 2 exemplifica uma árvore para o Jogo da Onça, em que os círculos representam diferentes estados de jogo. Cada nível contém um conjunto de estados distintos referentes a um jogador específico, situação característica de jogos baseados em turnos. Os estados destacados em cinza simbolizam *estados finais* – os nós folha da árvore de jogo –, nos quais o jogo termina, ou seja, ou um dos jogadores ganha ou há empate.

Figura 2: Representação abstrata de uma árvore para o Jogo da Onça.



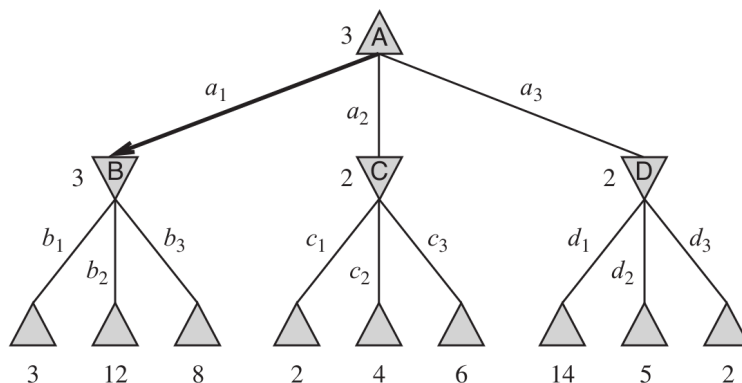
Gerar uma árvore de jogo completa significa encontrar todas as jogadas distintas que podem ocorrer durante qualquer partida, o que na maioria das vezes torna-se computacionalmente intratável, pois a maioria dos jogos de tabuleiro possuem uma quantidade colossal de estados de jogo – a exemplo do Xadrez que, segundo Russel e Norvig, possui cerca de 10^{40} estados de jogo distintos [Russel e Norvig 2016]. Para contornar esta dificuldade, utiliza-se uma *árvore de busca* para se referir a uma árvore que está sobreposta a uma árvore de jogo completa.

O algoritmo Minimax, por sua vez, atua obtendo estimativas de melhor jogada para um determinado personagem a partir do estado corrente e da árvore de busca. Para cada estado, um *valor minimax* é calculado a partir de jogadas futuras, simuladas recursivamente, supondo-se que ambos os jogadores tenham um desempenho ótimo. Assim, deseja-se maximizar o desempenho de um dos jogadores em detrimento da minimização do desempenho do adversário. De tal modo, o valor minimax determina então uma estratégia ótima para cada estado da árvore de jogo e corresponde à *utilidade* para um estado. Dessa maneira, o algoritmo Minimax percorre todo o caminho descendente até os estados finais (folhas) e propaga os melhores valores minimax de volta pela árvore, conforme termina a recursão, segundo uma estratégia de *backtracking* [Russel e Norvig 2016].

Para um melhor entendimento deste algoritmo, a Figura 3 ilustra a estratégia adotada pelo mesmo, em que nós \triangle são estados que se deseja maximizar e os nós ∇ , estados que se deseja minimizar. As arestas a_1, a_2, a_3 representam diferentes jogadas para o estado A , o mesmo vale para estados B e C . O algoritmo começa no nó A e desce recursivamente até B que, por sua vez, encontra 3 estados finais: 3, 12 e 8, valores que julgam o quão bom um estado é para ser maximizado. Estes números representam valores minimax

referentes a jogadas de minimização para o estado B . Como tem-se o desempenho ótimo, ele deve escolher a jogada que possui o menor valor, ou seja, movimento b_1 com valor 3. O mesmo vale para os estados C e D . Após obter todos os valores para os estados em que o adversário joga, o algoritmo retorna ao estado A e este escolhe a jogada que maximize sua vitória – neste caso: jogada a_1 de valor 3.

Figura 3: Exemplo de árvore de jogo com os valores minimax para cada nó.
Fonte: [Russel e Norvig 2016]



Infelizmente, como mencionado, explorar toda uma árvore de jogo é uma tarefa intratável. Para contornar esta situação, uma busca na árvore de jogo realizada pelo algoritmo Minimax é feita até um certo ponto – um *limiar* –, geralmente definido por um nível na árvore a ser descendido na busca. Acontece que após explorar todos os nós abaixo deste limiar, há a possibilidade de a recursão não encontrar um estado final, fazendo com que não haja retorno de valor no algoritmo. É neste cenário em que as *funções de utilidade* exercem um papel estratégico: se ao chegar neste limiar o algoritmo não encontrar nenhum estado final, a função de utilidade de estado é chamada para retornar uma estimativa que aquele estado teria se a árvore de jogo completa fosse considerada.

O projeto de funções de utilidade deve levar em conta conhecimentos a respeito do jogo em si, pois toda a informação estratégica do Minimax sobre qual melhor jogada futura a ser realizada é capturado exclusivamente por esta função. Desta maneira, no cerne da elaboração de bons jogadores inteligentes usando Minimax, reside o projeto das funções de utilidade.

3. Materiais e Métodos

Para alcançar os objetivos propostos neste trabalho, primeiramente será considerada uma investigação dos conceitos de IA aplicada a jogos e de como estes ocorrem no Jogo da Onça, visando atribuir uma caracterização compatível dos elementos presentes neste jogo com a nomenclatura formal da referida área. Isto auxiliará na identificação dos elementos de interesse e de como devem ser projetados, implementados e aferidos. Em analogia com outros jogos de tabuleiro, a modelagem segundo uma busca adversarial tem se mostrado mais compatível.

Em seguida, para construir um jogador inteligente para o Jogo da Onça segundo esta abordagem de busca, é essencial projetar uma boa função de utilidade, isto é, uma

estimativa do estado atual do tabuleiro e da sua relação com a vitória ou derrota no jogo [Russel e Norvig 2016]. De acordo com a literatura [Millington e Funge 2009, Capítulo 8], uma função de utilidade apropriada possui características particulares, tais como: se o tabuleiro está numa configuração final, esta função deve retornar a pontuação máxima quando o jogador a que se refere vence o jogo, e a pontuação mínima em caso contrário; nas demais situações, deve refletir um número de pontos compatível com a proximidade do jogador de ganhar ou perder o jogo. Porém, projetar tais funções não é uma tarefa fácil e, para contorná-la, costuma-se usar em um jogador o negativo da função utilidade do adversário ou estratégias que levem em conta a quantidade de peças adversárias, tais como no Xadrez, ou até mesmo técnicas de Aprendizado de Máquina [Russel e Norvig 2016].

No escopo deste trabalho, para o Jogo da Onça, serão consideradas diferentes funções de utilidade, visando uma comparação e análise de quais delas são mais apropriadas aos personagens a que se referem. Tanto no caso da Onça quanto do Cachorro, também serão consideradas movimentações aleatórias, como uma estratégia para projetar um jogador ingênuo, contra o qual espera-se uma vitória com alta probabilidade se o adversário for minimamente inteligente.

Para traçar estratégias eficazes para os jogadores é essencial então projetar tais funções levando em conta as especificidades de cada jogador em particular, atribuindo pontuações compatíveis com as características e objetivos que possuem. Ressalta-se, em especial, a não trivialidade nesta proposição para o jogo considerado: os adversários encontram-se em número diferente, uma onça versus catorze cachorros, isto é, um jogo entre desiguais, com movimentos e objetivos diferentes, em que a missão da onça é capturar cachorros e a dos cachorros é encurralar a onça. Em contraste com os jogos de Damas e Xadrez, por exemplo, os jogadores encontram-se em mesma quantidade, com o mesmo objetivo e com o mesmo tipo de movimentação de peças. No caso do Jogo da Onça, cita-se ainda a assimetria do tabuleiro, situação pouquíssimo frequente em jogos desta natureza.

Na proposição de funções de utilidade para este jogo, estabeleceu-se que os valores deveriam estar no intervalo $[-1000, +1000]$ com pontuações intermediárias inteiras, visando aproveitar-se da eficiência da aritmética de inteiros no âmbito computacional [Millington e Funge 2009]. No caso da Onça, além das jogadas aleatórias, considerou-se a proposição de uma função de utilidade que contabiliza +200 pontos a cada cachorro capturado, de modo que se efetue uma estimativa crescente à medida que esta concretiza o seu objetivo. No caso dos Cachorros, esta proposição foi menos trivial, pois deveria considerar a inteligência coletiva das peças com vistas a encurralar a onça, ao passo que este cálculo não poderia ser custoso para fins de eficiência. Neste caso, diferentes estratégias foram traçadas e serão avaliadas individualmente. A Tabela 1 apresenta as diferentes funções de utilidade propostas e as estratégias contempladas pelas mesmas.

Para avaliar as funções de utilidade consideradas, serão promovidos jogos ótimos entre os pares de funções apresentados, totalizando 10 pares de funções a serem confrontadas. Considerando o tamanho intratável de configurações de jogo existentes, é essencial adotar uma estratégia de caminhamento na árvore de jogo, com vistas a limitar o número de jogadas à frente que serão examinadas. Para tanto adotou-se então o algoritmo Minimax com profundidade de caminhamento na árvore de busca igual a 5 ou 6. As partidas consideradas terão até 25 turnos e serão realizadas 100 partidas por configuração, totali-

Tabela 1: Funções de utilidade propostas para os diferentes personagens do Jogo da Onça.

Funções de Utilidade da Onça	
$f_{o,0}$	Jogador aleatório
$f_{o,1}$	+200 pontos a cada cachorro capturado
Funções de Utilidade do Cachorro	
$f_{c,0}$	Jogador aleatório
$f_{c,1}$	$-f_{o,1}$
$f_{c,2}$	A contagem da média de cachorros protegidos adjacentes e total de cachorros adjacentes à onça
$f_{c,3}$	Visa simular uma ideia de matilha, em que tem-se uma pontuação relativa às distâncias entre os cachorros e a onça. A partir da distância igual a 1, esta pontuação é de [+50, +100, +10, +1] e igual a zero para distâncias superiores.
$f_{c,4}$	Visa diminuir o número de casas livres para as quais a onça pode se mover

zando 2.000 partidas a serem executadas e contabilizadas. Destes jogos será considerado o quantitativo de partidas concluídas com vitória da onça, partidas concluídas com vitória do cachorro e partidas não concluídas.

O ambiente computacional utilizado para executar tais confrontos consiste de um computador *desktop* com processador Intel Core i5, 8GB de memória RAM, placa gráfica GTX 1050 4 GB e 500 GB de disco rígido. A linguagem de programação Python será utilizada para implementar o Jogo da Onça em si, considerando o paradigma de programação orientado a objetos, e também será utilizada para automatizar o cenário de testes contemplado e para computar e analisar as métricas de interesse.

4. Resultados e Discussão

Conforme a metodologia elencada para realização deste trabalho, o primeiro passo consistiu na identificação dos conceitos de IA no âmbito do Jogo da Onça. Neste jogo, o tabuleiro é tido como o ambiente, o lugar interagível dos agentes, com característica *completamente observável*, ou seja, as entidades têm acesso ao estado completo do ambiente em cada instante. Neste contexto, os agentes correspondem às entidades, que podem ser artificiais (um algoritmo) ou humanas (uma pessoa), e que representam os jogadores do Jogo da Onça, sendo, portanto, um *jogo multiagente*.

O Jogo da Onça pode ser também classificado como sendo *determinístico* pois, em uma determinada jogada, o próximo estado é totalmente determinado pelo estado atual e pela ação de um dos agentes, como mover uma peça ou comer um cachorro, por exemplo. Além disso, segue uma *dinâmica baseada em turnos*, pois os agentes se alternam em cada jogada, de maneira análoga ao Jogo de Damas, Xadrez, dentre outros jogos de tabuleiro. Na dinâmica do Jogo da Onça, um jogador ganha se o seu oponente perde. Isto o classifica como um jogo de *soma-zero*; a vitória de um jogador é a derrota de seu oponente. Por último, neste jogo, um agente conhece perfeitamente todas as ações que ocorreram anteriormente, assim como a posição de todas as peças em todos os momentos, ou seja, o Jogo da Onça é de *informação perfeita*. Cada agente sabe qual será o resultado de cada movimento e quais são as opções para o próximo passo, desde o início do jogo.

Após a identificação dos conceitos, deu-se prosseguimento à implementação do Jogo da Onça utilizando a linguagem Python. Para tanto, utilizou-se o paradigma de

Tabela 2: Resultados obtidos do cenário de testes para as diferentes funções de utilidade.

Função de Utilidade da Onça	Função de Utilidade do Cachorro	Limiar de Profundidade	Vitórias da Onça (%)	Vitórias do Cachorro (%)	Partidas não Concluídas (%)
$f_{o,0}$	$f_{c,0}$	—	9	0	91
$f_{o,0}$	$f_{c,1}$	5	0	0	100
$f_{o,0}$	$f_{c,2}$	5	0	7	93
$f_{o,0}$	$f_{c,3}$	5	1	1	100
$f_{o,0}$	$f_{c,4}$	5	0	2	98
$f_{o,1}$	$f_{c,0}$	5	50	0	50
$f_{o,1}$	$f_{c,1}$	5	0	0	100
$f_{o,1}$	$f_{c,2}$	5	0	0	100
$f_{o,1}$	$f_{c,3}$	5	0	0	100
$f_{o,1}$	$f_{c,4}$	5	0	0	100
$f_{o,0}$	$f_{c,1}$	6	0	0	100
$f_{o,0}$	$f_{c,2}$	6	0	0	100
$f_{o,0}$	$f_{c,3}$	6	0	0	100
$f_{o,0}$	$f_{c,4}$	6	0	3	97
$f_{o,1}$	$f_{c,0}$	6	63	0	37
$f_{o,1}$	$f_{c,1}$	6	0	0	100
$f_{o,1}$	$f_{c,2}$	6	0	0	100
$f_{o,1}$	$f_{c,3}$	6	0	5	95
$f_{o,1}$	$f_{c,4}$	6	0	1	99

tre as estratégias avaliadas, aquela que atribuiu +200 a cada cachorro capturado pela onça é a mais procedente dentre todas, visto que leva a um maior número de vitórias.

Por fim, ao examinar os jogos não concluídos, foi possível também observar que falta muita combatividade por parte dos cachorros, os quais dificilmente se reúnem de maneira efetiva para encurralar a onça, revelando que as funções de utilidade até então propostas não implementam estratégias produtoras para conduzir à vitória deste tipo de jogador.

5. Considerações Finais e Trabalhos Futuros

Este trabalho apresenta os resultados de uma análise comparativa de funções de utilidade para o algoritmo Minimax aplicado ao Jogo da Onça. Para alcançar este resultado foi necessário realizar a identificação dos conceitos de IA existentes neste jogo, relevando que o Jogo da Onça pode então ser classificado como completamente observável, multi-agente, de soma-zero e de informação perfeita. Após esta identificação, realizou-se a implementação do referido jogo em Python segundo o paradigma orientado a objetos.

Foram então concebidas e comparadas diferentes funções de utilidade para o Jogo da Onça considerando o caminhamento na árvore de busca segundo o algoritmo Minimax, uma vez que a árvore de jogo em si é intratável. Estabeleceu-se um cenário de avaliação, com parâmetros pré-fixados e replicabilidade, e executaram-se os jogos pré-determinados com as diferentes funções propostas.

Os resultados obtidos mostraram que há uma predominância de vitórias da onça,

sugerindo que este jogador pode vir a ter maior probabilidade que o seu oponente. Verificou-se ainda uma função de utilidade adequada para este jogador, que acrescenta pontos à medida que captura cachorros. No geral, as funções de utilidade estabelecidas para o cachorro não foram producentes, revelando uma dificuldade em capturar na função de utilidade a inteligência coletiva que existe por trás deste agente.

Considerando os desafios verificados, em trabalhos futuros deseja-se avaliar mais cenários comparativos, incluindo outras funções de utilidade e aumentando o número de turnos considerados. Almeja-se também verificar a existência na literatura de funções de utilidade aplicadas a outros jogos que possam ser úteis na modelagem de uma estratégia eficaz para o personagem do cachorro, e ainda a aplicação de outros algoritmos de busca adversarial, como o *Alpha-Beta Pruning*.

6. Agradecimentos

Os autores agradecem o apoio financeiro provido pela Fundação de Amparo à Pesquisa do Estado do Amazonas (FAPEAM) por meio dos programas PAIC/UEA/2018-2019 e PPP 04/2017 e as sugestões providas pelos Profs. André Valente e Marcos Bandeira.

Referências

- Bettin, A. D. H. e Pretto, V. (2016). A construção do jogo indígena a partir da matemática. In *Anais do XXI SIEDUCA – A escola já não é a mesma: novos tempos, novos paradigmas*, volume 21, pages 266–273.
- Calderaro, K. C. L. (2006). O universo lúdico das crianças indígenas. Centro Cultural dos Povos da Amazônia. Disponível em <https://bit.ly/2U3tWQS>. Acessado em 11 de maio de 2019.
- Chen, J. X. (2016). The evolution of computing: Alphago. *Computing in Science & Engineering*, 18(4):4–7.
- de Oliveira Sardinha, A. G. e Gaspar, M. T. J. (2010). Jogos indígenas aplicados ao ensino de Matemática. In *Anais do X Encontro Nacional de Educação Matemática*, pages 1–10, Salvador, Bahia.
- Hsu, F.-H. (2002). *Behind Deep Blue*. Princeton University Press, Estados Unidos, 1 edição.
- Millington, I. e Funge, J. (2009). *Artificial Intelligence for Games*. Elsevier, Estados Unidos, 2 edição.
- Russel, S. e Norvig, P. (2016). *Artificial Intelligence – A Modern Approach*. Pearson, Nova Jersey, 3 edição.
- Secretaria Municipal de Educação (2016). O jogo da onça - aprenda a jogar. Prefeitura de São Paulo. Disponível em: <https://bit.ly/2YAVX0P>. Acessado em 11 de maio de 2019.