

UNIVERSIDADE FEDERAL DO RIO DE JANEIRO
INSTITUTO DE COMPUTAÇÃO
CURSO DE BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

CARLOS EDUARDO DA SILVA SANTOS
WILLIAN RAPHAEL ROSA GOMES

AVALIAÇÃO DE ESTRATÉGIAS PARA O ALGORITMO MINIMAX NO JOGO DA
ONÇA

RIO DE JANEIRO
2026

CARLOS EDUARDO DA SILVA SANTOS
WILLIAN RAPHAEL ROSA GOMES

AVALIAÇÃO DE ESTRATÉGIAS PARA O ALGORITMO MINIMAX NO JOGO DA
ONÇA

Trabalho de conclusão de curso de graduação
apresentado ao Instituto de Computação da
Universidade Federal do Rio de Janeiro como
parte dos requisitos para obtenção do grau de
Bacharel em Ciência da Computação.

Orientador: Prof. João Carlos Pereira da Silva

RIO DE JANEIRO
2026

CIP - Catalogação na Publicação

S237a	Santos, Carlos Eduardo da Silva Avaliação de estratégias para o Algoritmo Minimax no Jogo da Onça / Carlos Eduardo da Silva Santos e Willian Raphael Rosa Gomes. – Rio de Janeiro, 2026. 62 f. Orientador: João Carlos Pereira da Silva. Trabalho de Conclusão de Curso (Bacharelado em Ciência da Computação)- Universidade Federal do Rio de Janeiro, Instituto de Computação, Bacharel em Ciência da Computação, 2026. 1. Inteligência artificial. 2. Jogo da Onça. 3. Algoritmo Minimax. 4. Jogos de tabuleiro. 5. Poda Alpha-Beta. I. Gomes, Willian Raphael Rosa. II. Silva, João Carlos Pereira da (Orient.). III. Universidade Federal do Rio de Janeiro, Instituto de Computação. IV. Título.
-------	--

CARLOS EDUARDO DA SILVA SANTOS
WILLIAN RAPHAEL ROSA GOMES

AVALIAÇÃO DE ESTRATÉGIAS PARA O ALGORITMO MINIMAX NO JOGO DA
ONÇA

Trabalho de conclusão de curso de graduação
apresentado ao Instituto de Computação da
Universidade Federal do Rio de Janeiro como
parte dos requisitos para obtenção do grau de
Bacharel em Ciência da Computação.

Aprovado em 02 de Março de 2026

BANCA EXAMINADORA:

Documento assinado digitalmente
 JOAO CARLOS PEREIRA DA SILVA
Data: 09/03/2026 15:03:36-0300
Verifique em <https://validar.iti.gov.br>

João Carlos Pereira da Silva, D.Sc.
Instituto de Computação - UFRJ

Documento assinado digitalmente
 PAULO ROBERTO MANN MARQUES JUNIOR
Data: 09/03/2026 15:13:52-0300
Verifique em <https://validar.iti.gov.br>

Paulo Roberto Mann Marques Júnior, D.Sc.
Instituto de Computação - UFRJ

Documento assinado digitalmente
 SILAS PEREIRA LIMA FILHO
Data: 09/03/2026 19:16:44-0300
Verifique em <https://validar.iti.gov.br>

Silas Pereira Lima Filho, D.Sc.
Instituto de Computação - UFRJ

AGRADECIMENTOS

Primeiramente, gostaríamos de agradecer aos amigos e familiares por todo apoio e ajuda durante esse árduo processo e que muito contribuíram para realização desse trabalho.

Em especial ao nosso amigo Gabriel Ferrari pela sua contribuição na criação de todos os elementos gráficos utilizados no projeto, incluindo o design do jogo, a elaboração das peças, as imagens de fundo de tela inicial e do tabuleiro, a escolha da fonte e da palheta de cores bem como qualquer outro recurso visual presente neste trabalho, todos de sua autoria e que nos ajudaram a enriquecer este trabalho.

E por fim ao nosso professor João Carlos pela dedicação e orientação durante esse processo, pelas correções e trocas de conhecimento neste período.

RESUMO

Este trabalho busca estudar a aplicação da técnica de Inteligência Artificial conhecida com Algoritmo Minimax ao jogo de tabuleiro de origem indígena brasileira, conhecido como o jogo da onça ou adugo. O jogo envolve dois papéis distintos: o da onça, cujo o objetivo é capturar cinco cachorros, e o dos cachorros, cujo o objetivo é cercar a onça, impedindo-a de se movimentar. O objetivo do trabalho é fazer uma análise exploratória sobre a aplicação do algoritmo Minimax no jogo da onça, buscando analisar e propor funções de utilidade com o intuito de definir as melhores estratégias para ambos os agentes. Dessa forma, busca-se equilibrar o desempenho da onça e dos cachorros, analisando como diferentes funções de utilidade impactam no desempenho de ambos. Para isso, foram realizados diversos experimentos variando funções de utilidade, profundidade, limite de turnos, entre outros, afim de avaliar o impacto desses parâmetros no comportamento e desempenho dos agentes. Os resultados indicam que a onça possui uma vantagem considerável em relação aos cachorros, tornando evidente a necessidade de estratégias mais bem pensadas na jogabilidade dos cachorros para tornar o jogo mais equilibrado.

Palavras-chave: inteligência artificial; jogo da onça; algoritmo minimax; jogos de tabuleiro; poda alpha-beta.

ABSTRACT

This work aims to study the application of the Artificial Intelligence technique known as the Minimax Algorithm to a Brazilian indigenous board game known as the jaguar game or adugo. The game involves two distinct roles: the jaguar, whose objective is to capture five dogs, and the dogs, whose objective is to surround the jaguar, preventing it from moving. The objective of this work is to conduct an exploratory analysis of the application of the Minimax algorithm in the jaguar game, seeking to analyze and propose utility functions in order to define the best strategies for both agents. Thus, it seeks to balance the performance of the jaguar and the dogs, analyzing how different utility functions impact the performance of both. To this end, several experiments were carried out varying utility functions, depth, turn limits, among others, in order to evaluate the impact of these parameters on the behavior and performance of the agents. The results indicate that the jaguar has a considerable advantage over the dogs, highlighting the need for more well-thought-out strategies in the dogs' gameplay to make the game more balanced.

Keywords: artificial intelligence; machine learning; jogo da onça; minimax; board games; alpha-beta pruning.

LISTA DE ILUSTRAÇÕES

Figura 1 – Árvore Completa do jogo da velha	13
Figura 2 – Árvore Incompleta do jogo da velha	14
Figura 3 – Exemplo poda Alpha-Beta	15
Figura 4 – Tabuleiro com a configuração inicial do jogo	16
Figura 5 – Configuração do Tabuleiro – Exemplo 1	18
Figura 6 – Tela Inicial - Jogo da Onça	20
Figura 7 – Menu Jogar - Jogo da Onça	21
Figura 8 – Opções de Jogo (modo Comp vs Comp) - Jogo da Onça	21
Figura 9 – Tela Partida - Jogo da Onça	22
Figura 10 – Seleção e Reprodução do Histórico - Jogo da Onça	22
Figura 11 – Tabuleiro Numerado	23
Figura 12 – Configuração do Tabuleiro – Exemplo 2	25
Figura 13 – Histograma - Onça ($f_{o,0}$) x Cachorros ($f_{c,0}$) - Profundidade 5	28
Figura 14 – Histograma - Onça ($f_{o,1}$) x Cachorros ($f_{c,1}$) - Profundidade 5	28
Figura 15 – Árvore de decisão da onça	31
Figura 16 – Árvore de decisão dos cachorros	31
Figura 17 – Exemplo Funcionamento Minimax	37
Figura 18 – Trecho do log gerado no Experimento 5	40
Figura 19 – Distribuição das posições em que a onça foi derrotada no Experimento 5	41
Figura 20 – Turno 17 (Onça) - Partida do Experimento 5	41
Figura 21 – Turno 26 (Cachorros) e Turno 27 (Onça)	42
Figura 22 – Turno 31 (Onça) - Captura	42
Figura 23 – Turno 32 (Cachorros) e Turno 33 (Onça)	43
Figura 24 – Turno 35 (Onça) - Captura	43
Figura 25 – Turno 36 (Cachorros) e Turno 37 (Onça)	44
Figura 26 – Turno 40 (Cachorros) e Turno 41 (Onça)	44
Figura 27 – Turno 42 (Cachorros) - Vitória dos Cachorros	44
Figura 28 – Distribuição das posições em que a onça foi derrotada no Experimento 6	45
Figura 29 – Turno 40 (Cachorros) e Turno 132 (Cachorros)	46
Figura 30 – Turno 136 (Cachorros) e Turno 146 (Cachorros)	46
Figura 31 – Turno 149 (Onça) e Turno 150 (Cachorros)	47
Figura 32 – Turno 152 (Cachorros) - Vitória dos Cachorros	47
Figura 33 – Distribuição das posições em que a onça foi derrotada no Experimento 7	48
Figura 34 – Turno 15 (Onça) e Turno 19 (Onça)	48
Figura 35 – Turno 21 (Onça) e Turno 22 (Cachorros)	49
Figura 36 – Distribuição das posições em que a onça foi derrotada no Experimento 8	49

Figura 37 – Turno 5 (Onça) e Turno 17 (Onça)	50
Figura 38 – Turno 35 (Onça) e Turno 37 (Onça)	50
Figura 39 – Turno 105 (Onça) e Turno 144 (Onça)	51
Figura 40 – Turno 775 (Onça) e Turno 776 (Cachorros)	51
Figura 41 – Distribuição das posições em que a onça foi derrotada no Experimento 9	52
Figura 42 – Turno 7 (Onça) e Turno 15 (Onça)	53
Figura 43 – Turno 21 (Onça) e Turno 23 (Onça)	53
Figura 44 – Turno 997 (Onça) e Turno 999 (Onça)	54
Figura 45 – Configuração do Tabuleiro – Exemplo 3	54
Figura 46 – Configuração do Tabuleiro – Exemplo 4	55
Figura 47 – Graus dos vértices	56
Figura 48 – Histograma - Onça ($f_{o,0}$) x Cachorros ($f_{c,1}$) - Profundidade 5	60
Figura 49 – Histograma - Onça ($f_{o,0}$) x Cachorros ($f_{c,2}$) - Profundidade 5	60
Figura 50 – Histograma - Onça ($f_{o,1}$) x Cachorros ($f_{o,0}$) - Profundidade 5	60
Figura 51 – Histograma - Onça ($f_{o,1}$) x Cachorros ($f_{c,2}$) - Profundidade 5	61
Figura 52 – Histograma - Onça ($f_{o,2}$) x Cachorros ($f_{c,0}$) - Profundidade 5	61
Figura 53 – Histograma - Onça ($f_{o,2}$) x Cachorros ($f_{c,1}$) - Profundidade 5	61
Figura 54 – Histograma - Onça ($f_{o,2}$) x Cachorros ($f_{c,2}$) - Profundidade 5	62

LISTA DE TABELAS

Tabela 1 – Resultados obtidos do cenário de testes para as diferentes funções de utilidade.	19
Tabela 2 – Comparação entre Minimax Puro e com Poda Alpha-Beta	24
Tabela 3 – Resultados obtidos utilizando a função $f_{o,2}$ (Função 3.1)	26
Tabela 4 – Resultados obtidos sem Limite de Turnos e com Profundidade 5	27
Tabela 5 – Resultados da função $f_{c,3}$ (Função 3.3) com $w_1 = 0.4$ e $w_2 = 0.6$. . .	29
Tabela 6 – Resultados da função $f_{c,3}$ (Função 3.3) com $w_1 = 0.6$ e $w_2 = 0.4$. . .	30
Tabela 7 – Resultados obtidos após modificação no Minimax	32
Tabela 8 – Resultados obtidos após modificação no Minimax e sem falta de combatividade	33
Tabela 9 – Resultados obtidos contra a onça aleatória após modificação no Minimax e sem falta de combatividade	34
Tabela 10 – Resultados com onça em profundidade 3 e cachorros em profundidade 6	35
Tabela 11 – Resultados com onça em profundidade 6 e cachorros em profundidade 3	35
Tabela 12 – Resultados com a nova modificação no Minimax	37
Tabela 13 – Resultados Gerais das funções de utilidade	39
Tabela 14 – Resultados comparativo entre as funções da onça	39

SUMÁRIO

1	INTRODUÇÃO	10
2	REFERENCIAL TEÓRICO	13
2.1	MINIMAX	13
2.1.1	Poda Alpha-Beta	15
2.2	JOGO DA ONÇA	15
2.3	TRABALHOS RELACIONADOS	16
3	IMPLEMENTAÇÃO E EXPERIMENTOS	20
3.1	IMPLEMENTAÇÃO	20
3.2	EXPERIMENTO 1	22
3.3	EXPERIMENTO 2	24
3.4	EXPERIMENTO 3	27
3.5	EXPERIMENTO 4	29
3.6	EXPERIMENTO 5	30
3.7	EXPERIMENTO 6	32
3.8	EXPERIMENTO 7	33
3.9	EXPERIMENTO 8	34
3.10	EXPERIMENTO 9	36
3.11	ANÁLISE GERAL	38
4	ANÁLISE QUALITATIVA	40
4.1	EXPERIMENTO 5	40
4.2	EXPERIMENTO 6	45
4.3	EXPERIMENTO 7	47
4.4	EXPERIMENTO 8	49
4.5	EXPERIMENTO 9	51
4.6	VISÃO GERAL	54
5	CONCLUSÃO	57
	REFERÊNCIAS	59
	APÊNDICE A – HISTOGRAMAS UTILIZADOS NO EXPERIMENTO	
	3	60

1 INTRODUÇÃO

Atualmente, a inteligência artificial é uma área que abrange uma enorme variedade de subcampos, que vão das mais gerais (como aprendizado e percepção) até os mais específicas, como provar teoremas matemáticos, dirigir um carro em uma rua movimentada, ajudar a diagnosticar doenças e jogar xadrez. A inteligência artificial é relevante para qualquer área intelectual nos dias de hoje e é um campo realmente universal (RUSSELL; NORVIG, 2016).

Um dos primeiros subcampos a despertar o interesse da inteligência artificial foi a teoria dos jogos, motivado pela necessidade de desenvolver estratégias para jogar e vencer jogos. Jogos como xadrez (HSU, 2002), poker e Go (SILVER et al., 2016) atraíram a curiosidade de pesquisadores da área de inteligência artificial devido à sua natureza simplificada, sendo jogos em que um estado de jogo é facilmente representado e os agentes são restritos a um pequeno número de ações, definidas por regras precisas. Em contrapartida, jogos mais físicos, como o futebol e vôlei, possuem descrições mais complexas, uma maior variedade de ações por parte dos agentes e regras imprecisas ou subjetivas, o que torna mais desafiadora a aplicação de técnicas de inteligência artificial (RUSSELL; NORVIG, 2016).

Além disso, a área de teoria dos jogos é responsável por analisar as diferentes ações estratégicas envolvidas na tomada de decisões dos jogadores, buscando melhorar o retorno obtido. Também pode ser definida como um ambiente multiagente, onde as ações dos agentes devem ser consideradas em relação à influência aos outros agentes (FROZZA; SORIO; ALVARES, 1998).

Inicialmente, as pesquisas em torno da área tiveram como um dos focos principais o xadrez, com a empresa IBM desenvolvendo um supercomputador especificamente para jogá-lo. O computador, chamado Deep Blue, possuía 256 co-processadores e era capaz de avaliar aproximadamente 100 milhões de posições por segundo. Em fevereiro de 1996, o campeão do mundo de xadrez, Garry Kasparov, enfrentou pela primeira vez o supercomputador Deep Blue em uma série de seis partidas, na qual Kasparov venceu três, empatou duas e perdeu uma, obtendo a pontuação final de 4 a 2 para Kasparov (o empate vale 0,5 pontos para cada jogador). Essa derrota foi a primeira vez em que um computador derrotou um campeão mundial sob regras normais de tempo. (CAMPBELL; JR; HSU, 2002)

Após a derrota do Deep Blue no primeiro embate, os pesquisadores da IBM promoveram uma série de melhorias, aumentando a capacidade de processamento de 100 milhões de jogadas por segundo para 200 milhões de jogadas por segundo. Em maio de 1997 foi promovido um novo encontro entre Kasparov e o Deep Blue, dessa vez com resultado favorável ao computador da IBM. O programa derrotou Garry Kasparov pelo placar de 3,5 a 2,5, em um evento que ficou conhecido como "o mais espetacular evento de xadrez

na história", tornando o Deep Blue o primeiro computador a ter vencido um campeão mundial de xadrez num torneio com regras oficiais de tempo. (CAMPBELL; JR; HSU, 2002)

Com o avanço do tempo e da tecnologia, as estratégias adotadas por algoritmos de inteligência artificial foram se modificando. Em oposição ao método de força bruta ou de tentar traduzir o conhecimento de um excelente jogador para a máquina, ganharam destaque técnicas que possibilitassem ao computador aprender a jogar de forma autônoma, como o aprendizado por reforço.

No caso do Go, foi desenvolvido um sistema chamado AlphaGo que introduziu técnicas inovadoras ao combinar redes neurais profundas com o algoritmo Monte Carlo Tree Search (SILVER et al., 2016), conseguindo alcançar um desempenho sobre-humano, sendo o primeiro programa a derrotar um campeão mundial de Go, em 2016. Sendo desenvolvido posteriormente uma versão aprimorada do sistema, chamado de AlphaGo Zero, que conseguiu derrotar sua versão anterior (SILVER et al., 2017). Esse tipo de estudo é interessante dentro da área de inteligência artificial, pois demonstra a capacidade desses sistemas de desenvolver estratégias inéditas, não conhecidas dos jogadores humanos.

Nesse sentido, este trabalho utiliza técnicas fundamentais da inteligência artificial clássica, mais especificamente o algoritmo Minimax para explorar as dinâmicas do jogo da onça, que é um jogo de origem indígena brasileiro, pertencente aos povos Bororós, Manchineris e Guaranis (SARDINHA; GASPARI, 2010). Esse algoritmo constrói uma árvore de possibilidades de jogadas futuras, onde o computador busca maximizar suas chances de vitória, enquanto assume que o adversário irá realizar o melhor movimento possível para minimizá-las. Entretanto, devido a grande quantidade de estados possíveis no jogo da onça, utilizamos a poda Alpha-Beta, uma técnica de otimização que permite ao algoritmo podar ramos da árvore de decisão que não influenciarão o resultado final, reduzindo drasticamente o esforço computacional.

A escolha do algoritmo Minimax ocorreu por ser uma abordagem mais tradicional, buscando entender as dificuldades encontradas em um cenário menos complexo do que aquele normalmente abordado por técnicas de aprendizado por reforço. Além disso, acreditamos que a utilização do algoritmo Minimax pode descrever adequadamente o problema e os seus diferentes cenários, explorando modificações nas funções de utilidade, no grau de profundidade da busca e possíveis adaptações no algoritmo para se ajustar ao problema.

O principal objetivo do trabalho é fazer uma análise exploratória sobre a aplicação do algoritmo Minimax no jogo da onça, analisando e propondo funções de utilidade com o intuito de definir as melhores estratégias para ambos os agentes, buscando equilibrar o desempenho entre eles. Para isso, foram realizados diversos experimentos, com diferentes funções de utilidade, o que permitiu, através de análises quantitativas e qualitativas, avaliar o desempenho obtido e identificar pontos de melhorias.

Como objetivo adicional, buscamos difundir o jogo da onça, destacando sua relevância

cultural e histórica. Embora seja um jogo estratégico, com dinâmica assimétrica e muitas possibilidades estratégicas, ainda é pouco explorado na literatura científica quando comparamos com outros jogos, como xadrez e Go. Portanto, o trabalho não apenas contribui para área de inteligência artificial, mas também contribui para valorização e difusão da cultura indígena brasileira.

Dessa forma, as principais contribuições deste trabalho são:

- Propor e analisar diferentes funções de utilidade para os agentes do jogo da onça (onça e cachorros), com o objetivo de promover maior equilíbrio competitivo entre os dois lados do jogo.
- Realizar experimentos comparativos entre diferentes funções de utilidade, conduzindo análises quantitativas e qualitativas para identificar pontos fortes, limitações e possíveis melhorias.
- Apresentar modificações no funcionamento do algoritmo Minimax para jogos de tabuleiro assimétricos, como o jogo da onça.
- Contribuir para a difusão do jogo da onça, destacando sua relevância cultural e histórica no contexto da cultura indígena brasileira e incentivando sua exploração em pesquisas na área de inteligência artificial.

Este trabalho está organizado da seguinte forma. O capítulo 2 introduz conceitos básicos sobre o algoritmo Minimax e a Poda Alpha-Beta, descreve detalhadamente as regras e características do jogo da onça e apresenta alguns trabalhos relacionados ao jogo. O capítulo 3 são apresentados os experimentos realizados com as funções de utilidades e seus resultados quantitativos com diferentes parâmetros. O capítulo 4 mostra uma análise qualitativa dos experimentos realizados no capítulo anterior com foco nas partidas ganhas pelos cachorros, buscando identificar padrões. O Capítulo 5 apresenta as conclusões e sugestões para trabalhos futuros.

2 REFERENCIAL TEÓRICO

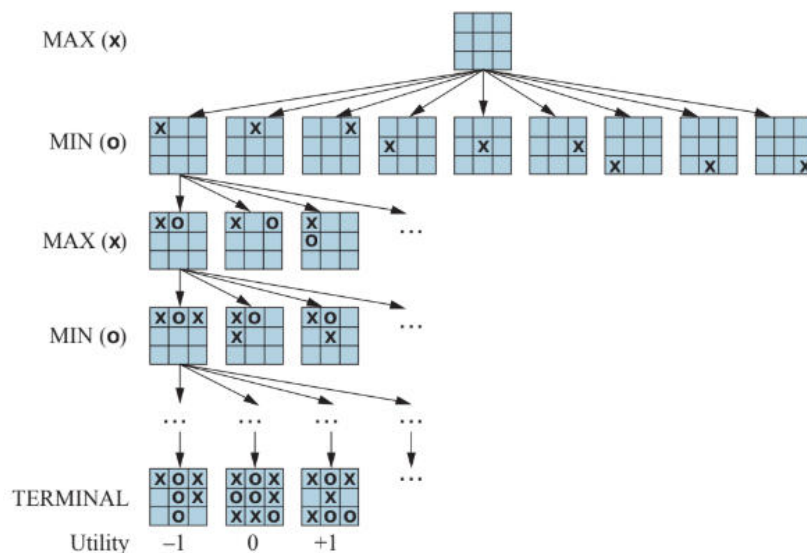
2.1 MINIMAX

O algoritmo Minimax (RUSSELL; NORVIG, 2016) é uma técnica para jogos com as seguintes características:

- Jogos determinísticos;
- Jogadas alternadas entre dois jogadores;
- Soma zero, em que o ganho de um jogador é exatamente igual à perda do outro;
- Informação perfeita, em que todos os jogadores têm conhecimento das jogadas anteriores.

Seu objetivo é determinar a melhor estratégia para o jogador Max, assumindo que o oponente Min irá jogar de forma ideal. O algoritmo é recursivo e explora a árvore do jogo até as suas folhas, propagando o valor obtido pelas funções de utilidade à medida que as chamadas recursivas retornam até a raiz, desfazendo a pilha de chamadas construída. Por fim, retorna a jogada que maximiza o valor da função de utilidade para o jogador Max.

Figura 1 – Árvore Completa do jogo da velha

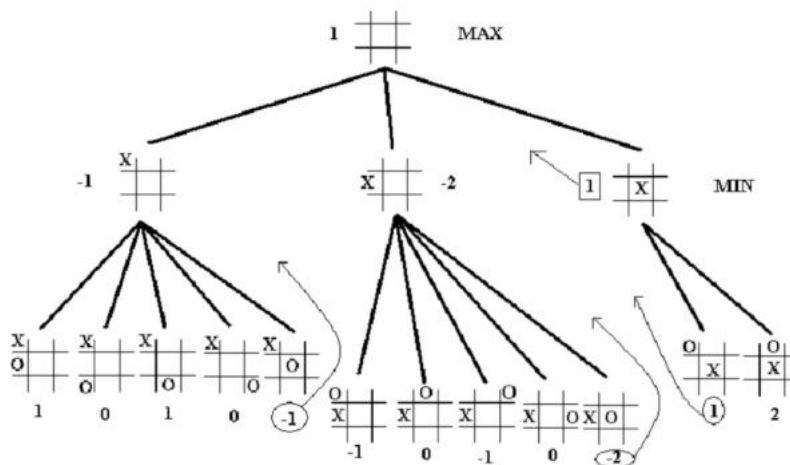


Fonte: RUSSELL; NORVIG (2016, p. 148)

Na Figura 1, vemos a árvore do jogo da velha, onde o algoritmo Minimax tenta encontrar a melhor jogada. Cada nó representa um estado do tabuleiro e os níveis representam os turnos dos jogadores (X e O). Ao final, são calculados os valores dos estados terminais

que podem ser perda (-1), empate (0) ou vitória (+1). Com isso, o algoritmo indica os estados terminais que levam o jogador X à vitória.

Figura 2 – Árvore Incompleta do jogo da velha



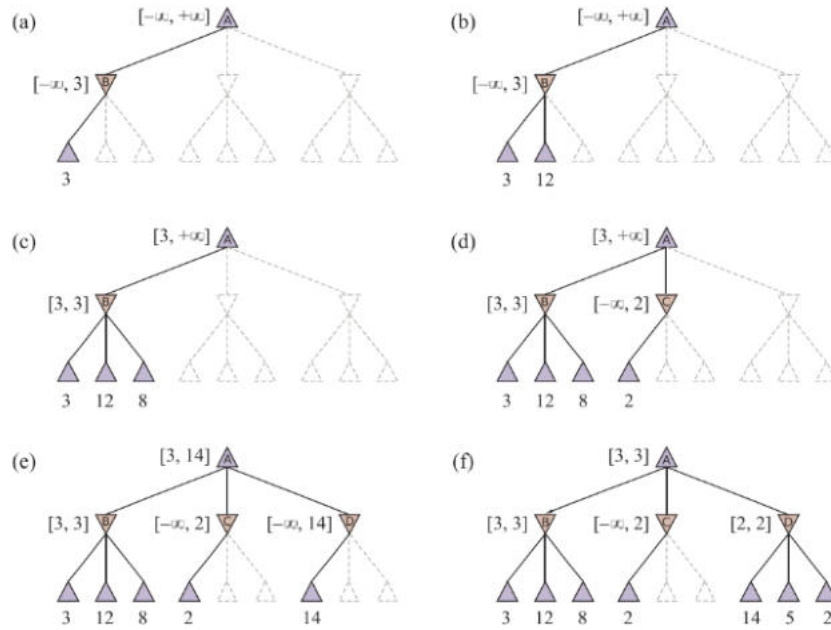
Fonte: SILVA, J. C. P. Material didático da disciplina Inteligência Artificial. Rio de Janeiro: Bacharelado em Ciência da Computação – UFRJ, 2020.

Porém, não é viável gerar a árvore completa para todos os jogos, pois se torna um processo muito demorado. A complexidade da árvore de jogo é dada por b^d , onde b representa o fator de ramificação do jogo e d o comprimento médio do jogo. No caso do jogo da velha, a complexidade da árvore do jogo é relativamente pequena, sendo 4^9 . Dessa forma, é possível gerar e analisar a árvore completa do jogo.

Entretanto, em jogos mais complexos, como o jogo da onça, a complexidade da árvore do jogo é maior devido à maior quantidade de peças e movimentos possíveis em cada turno, com o valor de $4,667^{70}$, como mostra o trabalho (NASCIMENTO; GUEDES, 2019). Para esses casos, é gerado uma árvore do jogo até determinada profundidade e seus nós são avaliados por uma função de utilidade.

Por exemplo, na Figura 2, não é gerada a árvore completa do jogo da velha, sendo necessário utilizar uma função de utilidade para avaliar o valor dos nós terminais da árvore incompleta de busca. Na figura, esses valores são representados nos nós terminais da árvore, indicando o resultado da avaliação heurística de cada estado terminal considerado, com a seta indicando os estados e seus respectivos valores que foram escolhidos e propagados para os nós superiores. Além disso, é possível utilizar recursos de otimização, como a poda Alpha-Beta para fazer essas avaliações de forma mais rápida e eficiente (RUSSELL; NORVIG, 2016).

Figura 3 – Exemplo poda Alpha-Beta



Fonte: RUSSELL; NORVIG (2016, p. 153)

2.1.1 Poda Alpha-Beta

A poda Alpha-Beta é uma técnica utilizada para otimização aplicada ao Minimax com o objetivo de reduzir o número de nós avaliados pelo Minimax, sem alterar o resultado final do algoritmo. Ele para de avaliar um ramo quando pelo menos uma possibilidade foi encontrada que prova que o ramo é pior do que um ramo examinado anteriormente.

O algoritmo mantém dois valores, α e β , que representam, respectivamente, o menor valor que um nó Max pode receber e o maior valor que um nó Min pode receber. Se $\beta \leq \alpha$, ele interrompe a busca naquele ramo.

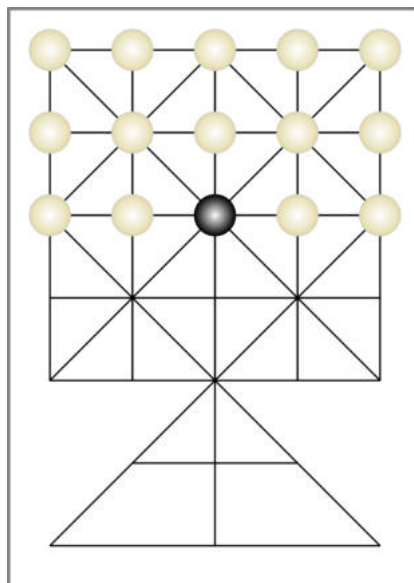
Na Figura 3, temos um exemplo de poda Alpha-Beta. Inicialmente os valores α e β , começam, respectivamente, com valores $-\infty$ e $+\infty$. Ao avaliar um dos ramos do nó B, atualiza o valor de β para 3, em seguida são avaliados os outros ramos do nó B, porém nenhum deles tem valor menor que 3, então o valor de β continua sendo 3, e retorna até o nó A, atualizando o valor de α para 3. Após isso, o nó C começa a ser avaliado e atualiza β com o valor de 2, porém α tem o valor de 3, fazendo a condição $\beta \leq \alpha$ ser verdadeira, e aplicando a poda Alpha-Beta, não avaliados os outros ramos no nó C.

2.2 JOGO DA ONÇA

O *jogo da onça*, também conhecido como *adugo* pelos indígenas, é um jogo de tabuleiro de origem indígena brasileiro, presente em diversas tribos indígenas, como os Bororós, no estado do Mato Grosso, os Manchineris no Acre, e os Guaranis em São Paulo (SARDINHA;

GASPAR, 2010). Na tradição indígena, é comum que o tabuleiro seja desenhado na areia e que pedras sejam utilizadas para representar as peças.

Figura 4 – Tabuleiro com a configuração inicial do jogo



Fonte: LIFE OF RILEY. Adugo.svg. 2010. Disponível em: <https://pt.wikipedia.org/wiki/Ficheiro:Adugo.svg>. Acesso em: 02 mar. 2026.

O tabuleiro possui ao todo 31 casas disponíveis, interligadas por linhas (verticais, horizontais e diagonais) que dão forma ao tabuleiro, possui uma peça representando a onça (peça preta) e outras catorze os cachorros (peças brancas), como mostra a Figura 4. Os movimentos são similares ao jogo de damas, com as peças podendo se movimentar para uma casa adjacente a sua atual, desde que essa esteja livre.

Existem dois objetivos distintos para cada agente. O objetivo da onça é capturar cinco cachorros, enquanto o objetivo dos cachorros é impossibilitar a onça de realizar qualquer movimento, semelhante ao xeque-mate no xadrez. Para a onça capturar um cachorro, ela deve realizar um salto sobre o cachorro adjacente e pousar em uma casa vazia na mesma direção. Além disso, nas regras tradicionais do jogo, a onça pode realizar capturas em sequência, de maneira semelhante ao jogo de damas, porém como os resultados no trabalho (NASCIMENTO; GUEDES, 2019) apresentaram uma vantagem para a onça, visando o equilíbrio entre os agentes, a captura em sequência não foi utilizada na implementação deste trabalho.

2.3 TRABALHOS RELACIONADOS

No trabalho de (FARIA; ARAÚJO; BAUMANN, 2017) é relatada a experiência no uso do jogo da onça no desenvolvimento educacional de alunos do 6º e 7º ano do EAJA (Educação de Adolescentes, Jovens e Adultos) do Estado de Goiás. Através do jogo,

eles puderam trabalhar conceitos matemáticos presentes no jogo e no tabuleiro, como noções de figuras geométricas, de ângulo, raciocínio lógico e estratégia, além de trazer o enriquecimento histórico e cultural sobre os indígenas e a origem do jogo .

Para além dos aspectos culturais e históricos, é fundamental considerar a teoria dos jogos para compreender as estratégias envolvidas no jogo da onça. Jogos com características como serem determinísticos, com jogadas alternadas, dois jogadores, soma zero e informação perfeita são adequados para implementação de técnicas de busca competitivas. (RUSSELL; NORVIG, 2016)

Nesse mesmo contexto, a pesquisa de (NASCIMENTO; GUEDES, 2019) acerca do jogo da onça faz uma análise comparativa das funções de utilidade propostas, que também utilizam do algoritmo MiniMax e da poda Alfa-Beta. Além disso, o estudo inclui uma estimativa para o tamanho da árvore do jogo.

A função de utilidade $f_{o,1}$ (Função 2.1), é uma das funções propostas pela pesquisa (NASCIMENTO; GUEDES, 2019). A função recebe x , que é a quantidade de cachorros capturados, e multiplica por 200. Assim, a onça é bonificada por capturar cachorros, atribuindo 200 pontos a cada captura, adotando uma estratégia gulosa que sempre prioriza a captura.

$$f_{o,1}(x) = 200x \quad (2.1)$$

Além dessa, outras duas funções de utilidade foram definidas para os cachorros a $f_{c,1}$ (Função 2.2) e a $f_{c,2}$ (Função 2.3). A primeira busca quantificar a restrição de movimentação da onça a partir da quantidade de cachorros vizinhos que ela pode capturar, com a função recebendo t , que é o número total de casas adjacentes à onça que possuem cachorro, e p sendo o número de cachorros adjacentes à onça que não podem ser capturados. Enquanto a segunda visa quantificar a área de movimentação livre da onça, considerando os movimentos diretos (casas alcançáveis em um único movimento) e indiretos (casas alcançáveis por meio de uma sequência de movimentos através das casa livres), recebendo y , que representa o número de casas livres acessíveis diretamente e indiretamente pela onça, obtido através de uma busca em largura no tabuleiro.

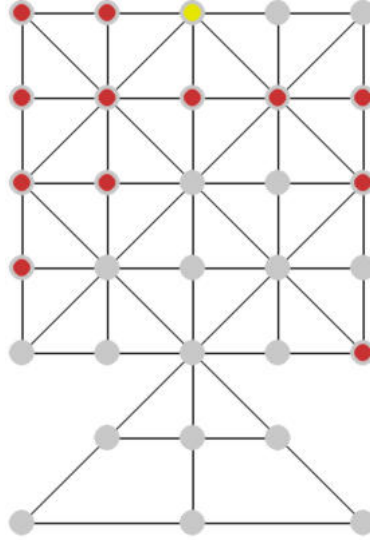
$$f_{c,1}(p, t) = \begin{cases} \text{trunc} \left(1000 \cdot \frac{p}{t} \right), & \text{se } t > 0 \\ 0, & \text{caso contrário} \end{cases} \quad (2.2)$$

$$f_{c,2}(y) = \min(1000, \max(-1000, 1000 - 100 \cdot y)) \quad (2.3)$$

Para exemplificar o cálculo das funções de utilidade $f_{c,1}$ (Função 2.2) e $f_{c,2}$ (Função 2.3), vamos considerar a configuração de tabuleiro mostrado na Figura 5. Para a função $f_{c,1}$ (Função 2.2), temos o valor de $t = 4$ e $p = 3$, pois 4 casas adjacentes à onça possuem cachorros e em 3 dessas casas os cachorros não podem ser capturados, então temos

$f_{c,1}(3, 4) = \text{trunc} \left(1000 \cdot \frac{3}{4} \right) = 750$. Já para a função $f_{c,2}$ (Função 2.3), temos o valor de $y = 18$, pois a onça pode alcançar as 18 casas livres (2 casas diretamente e 16 indiretamente), então temos $f_{c,2}(18) = \min(1000, \max(-1000, 1000 - 100 \cdot 18)) = -800$.

Figura 5 – Configuração do Tabuleiro – Exemplo 1



Em (NASCIMENTO; GUEDES, 2019), foi estabelecido que todas as funções propostas deveriam estar no intervalo $[-1000, +1000]$ e que todas as pontuações atribuídas seriam inteiras. Para efeito comparativo, foi adotado, inicialmente, um comportamento de exploração aleatória no tabuleiro, tanto para a onça quanto para os cachorros. Outra observação, foi a falta de combatividade (movimentos não orientados ao objetivo individual) por ambos os agentes, o que levou a estabelecer um critério para determinar o empate (identifica quando o agente repete ações em mais da metade dos turnos analisados, indicando falta de combatividade), e qual agente foi responsável pela falta de combatividade.

Para avaliar as funções de utilidade, foram realizadas partidas entre os pares de funções presentes com diferentes níveis de profundidade, entre eles 5, 6, 8 e 10. As partidas tiveram até 25 turnos e cada configuração foi repetida um total de 100 vezes.

Por exemplo, para onça existia 2 tipos de comportamentos, a caminhada aleatória pelo tabuleiro ($f_{o,0}$) e a guiada pela função de utilidade $f_{o,1}$ (Função 2.1), enquanto para os cachorros existiam a caminhada aleatória pelo tabuleiro ($f_{c,0}$) e as duas funções de utilidade $f_{c,1}$ (Função 2.2) e $f_{c,2}$ (Função 2.3), então uma combinação possível seria a caminhada aleatória para a onça e a função de utilidade $f_{c,1}$ (Função 2.2) para os cachorros, essa combinação era utilizada com os níveis de profundidade 5, 6, 8 e 10. Os testes abrangeram todas as combinações possíveis de funções entre onça e os cachorros, como mostra a Tabela 1.

Tabela 1 – Resultados obtidos do cenário de testes para as diferentes funções de utilidade.

Função de Utilidade		Profundidade	Vitórias		Partidas não concluídas	Falta de Combatividade (%)	
Onça	Cachorro		Onça	Cachorro		Onça	Cachorro
$f_{o,0}$	$f_{c,0}$	-	28	0	72	-	-
$f_{o,1}$	$f_{c,0}$	5	92	0	8	0	12.50
$f_{o,1}$	$f_{c,0}$	6	92	0	8	0	0
$f_{o,1}$	$f_{c,0}$	8	92	0	8	0	37.50
$f_{o,1}$	$f_{c,0}$	10	91	0	9	0	22.22
$f_{o,0}$	$f_{c,1}$	5	1	4	95	0	23.16
$f_{o,0}$	$f_{c,1}$	6	25	4	71	0	16.90
$f_{o,0}$	$f_{c,1}$	8	24	6	70	0	8.57
$f_{o,0}$	$f_{c,1}$	10	43	2	55	0	10.90
$f_{o,0}$	$f_{c,2}$	5	75	0	25	0	12.00
$f_{o,0}$	$f_{c,2}$	6	46	3	51	0	15.69
$f_{o,0}$	$f_{c,2}$	8	25	5	70	0	20.00
$f_{o,0}$	$f_{c,2}$	10	54	0	46	0	21.74
$f_{o,1}$	$f_{c,1}$	5	2	0	98	1.02	33.67
$f_{o,1}$	$f_{c,1}$	6	1	0	99	0	4.04
$f_{o,1}$	$f_{c,1}$	8	1	0	99	0	2.02
$f_{o,1}$	$f_{c,1}$	10	1	0	99	0	6.06
$f_{o,1}$	$f_{c,2}$	5	29	0	71	0	19.72
$f_{o,1}$	$f_{c,2}$	6	11	0	89	0	5.62
$f_{o,1}$	$f_{c,2}$	8	7	0	93	0	8.60
$f_{o,1}$	$f_{c,2}$	10	21	0	79	0	7.59

Fonte: NASCIMENTO; GUEDES (2019,p. 9)

Com resultados obtidos, foi observado que a função de utilidade $f_{o,1}$ (Função 2.1) teve um desempenho satisfatório, registrando um bom quantitativo de vitórias (vencendo em média 91,75% das partidas dentre todas as profundidades citadas) contra um adversário se movendo de forma aleatória e com uma média de vitórias de aproximadamente 17% contra a função $f_{c,2}$ (Função 2.3).

Já quando esta enfrenta a função $f_{c,1}$ (Função 2.2), há um confronto maior, com o número de partidas não concluídas muito elevado, como é mostrado na Tabela 1. Em relação as funções de utilidade proposta para os cachorros, se observou que elas são pouco efetivas em representar uma estratégia coletiva eficiente, pelo baixo índice de vitórias num geral (pouco mais de 1% dos casos totais) e pelo número de empates causados por esse agente pela falta de combatividade .

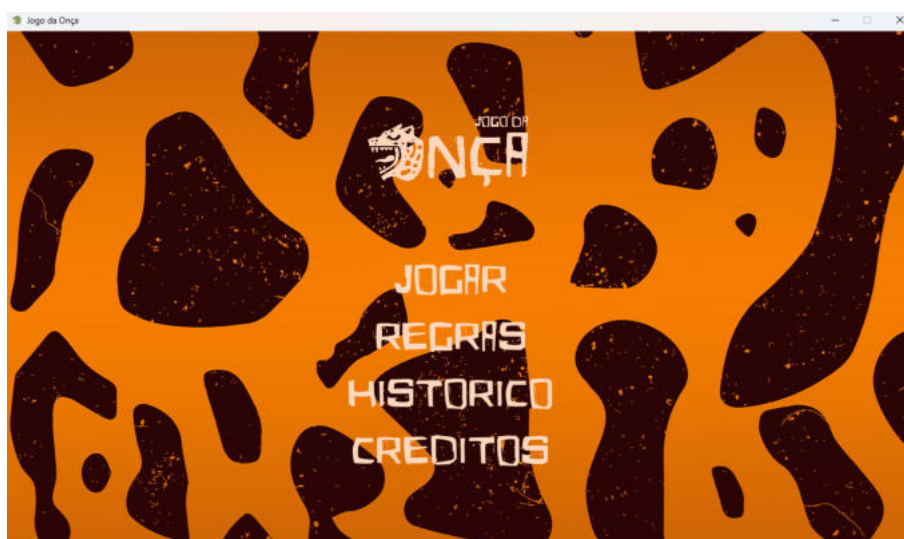
3 IMPLEMENTAÇÃO E EXPERIMENTOS

Neste capítulo, serão apresentados e discutidos os experimentos realizados, com o objetivo de melhorar o desempenho dos agentes e encontrar estratégias que busquem o equilíbrio entre eles. Para isso, os experimentos variam as funções de utilidade para avaliação dos agentes, assim como o limite de turnos e a presença de falta de combatividade, além da profundidade de busca na árvore do jogo. Além disso, também estão presentes detalhes sobre a implementação do jogo.

3.1 IMPLEMENTAÇÃO

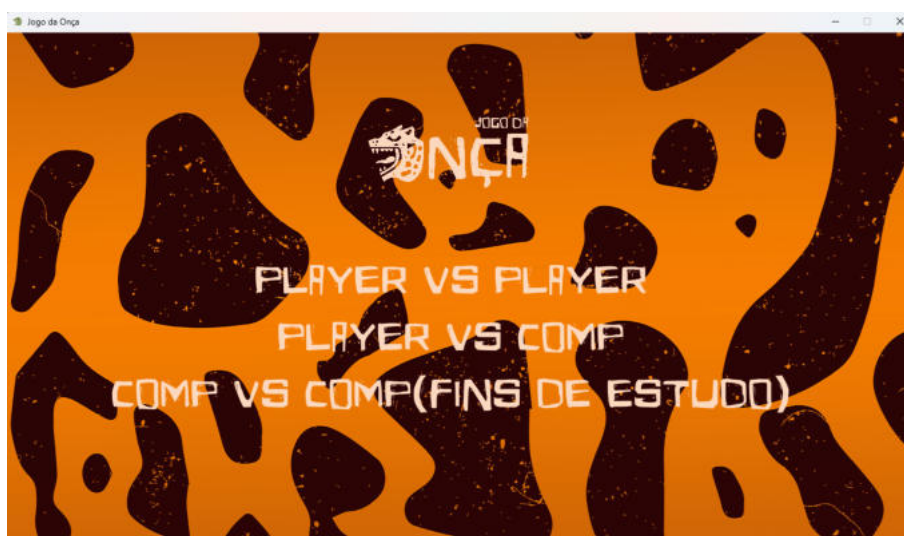
O jogo foi desenvolvido na linguagem de programação Python, utilizando de sua vasta opções de bibliotecas. Mais especificamente, a biblioteca Pygame (COMMUNITY, 2026), que foi a base no desenvolvimento do jogo. A escolha se fez pela facilidade da linguagem Python e por estarmos habituados a utilizá-la, otimizando assim o tempo gasto na implementação.

Figura 6 – Tela Inicial - Jogo da Onça



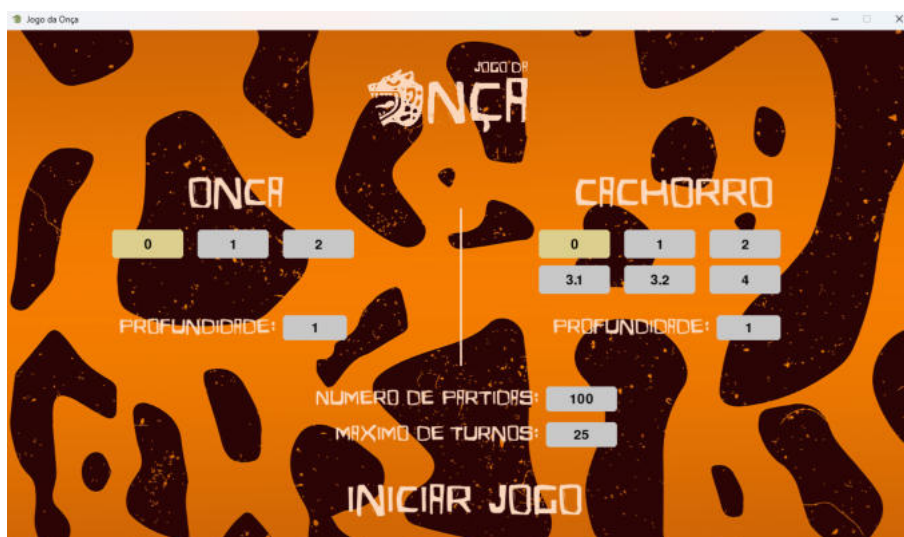
O jogo possui menus, como mostra a Figura 6, no qual o jogador pode selecionar diferentes opções, tais como Jogar, Regras, Histórico e Créditos. Ao selecionar a opção Jogar, é exibido um segundo menu, de acordo com a Figura 7, com as opções: Player vs Player, no qual dois jogadores humanos jogam entre si; Player vs Comp, em que um jogador humano enfrenta o programa; e Comp vs Comp, no qual o programa joga contra si mesmo.

Figura 7 – Menu Jogar - Jogo da Onça



Após escolher uma das opções de jogo, é possível definir os parâmetros, como o número limite de turnos, qual função de utilidade será utilizada para cada agente, o número de partidas, assim como a profundidade de cada agente também, como mostra a Figura 8.

Figura 8 – Opções de Jogo (modo Comp vs Comp) - Jogo da Onça



Além disso, é possível ver a execução das partidas por meio da tela da partida que exibe além do tabuleiro e as peças, o número do turno, a quem pertence cada turno e o número de cachorros capturados, como mostra a Figura 9.

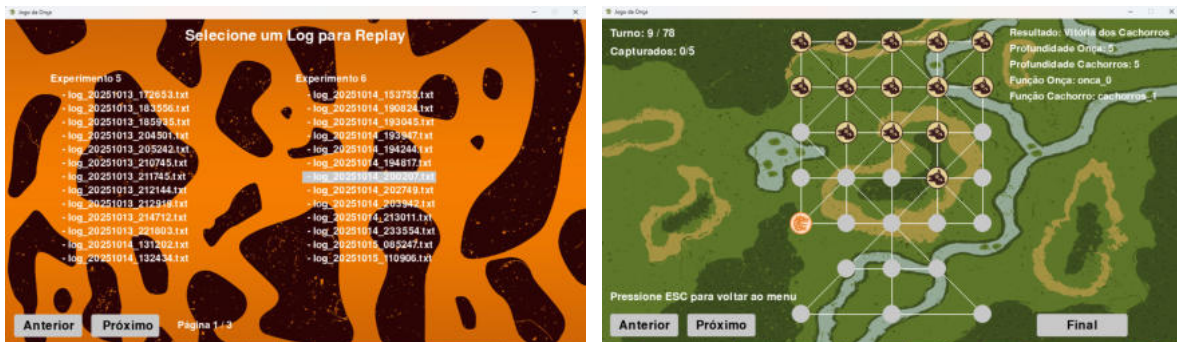
Também foi desenvolvido um sistema para guardar as partidas para serem analisadas posteriormente. Esse se encontra na aba Histórico do menu inicial do jogo. Nessa aba é possível consultar os registros de partidas salvas que estão organizadas por experimento,

Figura 9 – Tela Partida - Jogo da Onça



assim como ao selecionar um arquivo de partida, observar a movimentação turno a turno daquela partida selecionada, como é possível ver nas imagens Figura 10.

Figura 10 – Seleção e Reprodução do Histórico - Jogo da Onça



Toda a implementação desenvolvida neste trabalho, juntamente com código-fonte, imagens e registros de execução, está organizada e acessível no GitHub¹, onde é possível acompanhar o funcionamento do programa com mais detalhes.

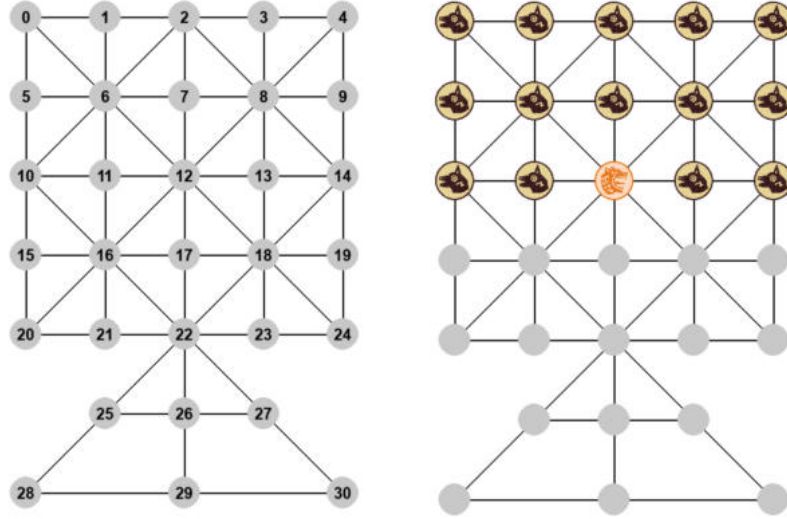
3.2 EXPERIMENTO 1

Para este experimento, tomamos como ponto de partida uma das funções de utilidade que foram apresentadas no trabalho (NASCIMENTO; GUEDES, 2019). A função de utilidade $f_{o,1}$ (Função 2.1), calcula o valor da função objetivo baseado na quantidade de cachorros capturados pela onça. Essa função utiliza uma estratégia gulosa, focando sempre em capturas.

¹ GOMES, W; SANTOS, C. Implementação do Jogo da Onça. GitHub, 2025. Disponível em: <<https://github.com/Dudu300599/Trabalho-Final.git>>. Acesso em: 4 dez. 2025.

Neste experimento, nós realizamos partidas contra o programa, sendo o programa responsável pelo papel da onça e nós pelos cachorros. Com o objetivo de analisar o comportamento da função de utilidade $f_{o,1}$ (Função 2.1) e identificar possíveis melhorias.

Figura 11 – Tabuleiro Numerado



Ao utilizar a função $f_{o,1}$ (Função 2.1) em nosso Minimax, observamos que o algoritmo sempre escolhia a casa de menor índice entre as que tinham o maior valor de utilidade. Um exemplo disso ocorre no início do jogo, quando a onça tem três movimentos possíveis com o valor de utilidade 0, já que não irá capturar nenhum cachorro nesses movimentos, sendo eles para as casas 16, 17 e 18, como mostra a Figura 11. Nesse caso, o Minimax sempre retornava a menor casa, a 16. Com isso, a onça sempre tendia a se mover para parte superior esquerda do tabuleiro. Além de sempre repetir os mesmos movimentos para diferentes partida, fazendo a onça ter um comportamento previsível.

Para evitar esse comportamento da onça e tornar seu comportamento mais dinâmico, armazenamos todas as jogadas empatadas com o maior valor de utilidade em um vetor e, posteriormente, selecionamos aleatoriamente uma das jogadas armazenadas. Com isso, no exemplo citado anteriormente, ela não escolherá sempre a casa 16, mas sim aleatoriamente entre as casas 16, 17 e 18.

Além disso, outro problema identificado ocorreu ao utilizar profundidade maiores que 2, fazendo a onça apresentar um comportamento de não combatividade, evitando capturar cachorros. Isso acontece porque o Minimax não consegue avaliar se é melhor realizar a captura no turno atual ou no seguinte. Com profundidade 3, por exemplo, o algoritmo simula os turnos da onça, dos cachorros e da onça novamente. Nesse caso, ele não diferencia se é mais vantajoso capturar no primeiro turno ou no segundo turno da onça, o que leva a situações em que faltava apenas um cachorro para vitória, mas a onça evitava a captura. Esse comportamento se acentua à medida que a profundidade aumenta, mas não foi relatado no trabalho (NASCIMENTO; GUEDES, 2019).

Somado a isso, quanto maior a profundidade, maior é o número de nós avaliados e o tempo de execução, resultando em tempos de execução muito elevados, não sendo viável utilizar valores maiores de profundidade. Para reduzir o tempo de execução e tornar viável a utilização de valores maiores de profundidade, utilizamos a Poda Alfa-Beta, que reduziu de forma considerável o tempo de execução, como mostra a Tabela 2.

Tabela 2 – Comparação entre Minimax Puro e com Poda Alpha-Beta

Comparação entre Minimax Puro e Minimax com Poda Alfa-Beta				
Profundidade	Minimax Puro		Minimax com Poda Alfa-Beta	
	Tempo (s)	Número de Nós	Tempo (s)	Número de Nós
2	0,0015	33	0,0019	15
4	0,0867	2128	0,0189	196
6	4,2851	139411	0,2161	2485
8	258,3019	9486845	1,8879	37780
10	-	-	27,3274	663458
12	-	-	481,9028	12540716

Em relação à jogabilidade da onça, utilizando a função $f_{o,1}$ (Função 2.1), foi identificado que ao adotar um comportamento guloso a onça se expõe a situações de risco, pois é possível sacrificar cachorros para colocar a onça em uma posição desfavorável e de fácil vitória para os cachorros. Em nossos testes, jogando contra a onça, conseguimos vencer sacrificando, em média, dois cachorros e posicionando-a nas casas com menos arestas.

Com base neste experimento, destacam-se as seguintes contribuições:

- Identificação de comportamento previsível da onça ao utilizar a função de utilidade $f_{o,1}$ (Função 2.1), devido à escolha da casa de menor índice em casos de empate do valor da função de utilidade.
- Implementação da escolha aleatória entre jogadas com o mesmo valor da função de utilidade, tornando o comportamento da onça menos previsível e mais dinâmico.
- Aplicação da poda Alpha-Beta ao algoritmo Minimax, reduzindo significativamente o número de nós avaliados e o tempo de execução.
- Observação de que uma estratégia puramente gulosa baseada em capturas pode expor a onça a posições desfavoráveis.

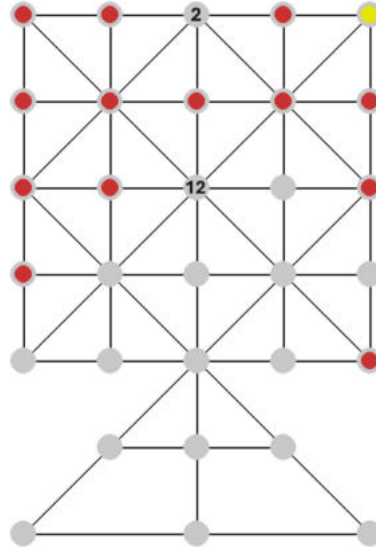
3.3 EXPERIMENTO 2

Observando o comportamento da onça no experimento anterior, modificamos a função $f_{o,1}$ (Função 2.1), colocando a quantidade de movimentos possíveis da onça naquele vértice, com o intuito de fazer a onça evitar os vértices com poucos movimentos disponíveis. A função $f_{o,2}$ (Função 3.1) recebe x , que é a quantidade de cachorros capturados, e n , que

é quantidade de movimentos possíveis daquele vértice, e multiplica, respectivamente, por 200 e 10. Esses valores foram escolhidos porque a onça pode realizar, no máximo, oito movimentos, com isso o valor máximo $10n$ será 80, sendo menor que qualquer captura. Dessa forma, o maior peso ainda será a captura, tornando a função de utilidade mais equilibrada, porém priorizando capturas.

$$f_{o,2}(x, n) = 200x + 10n \quad (3.1)$$

Figura 12 – Configuração do Tabuleiro – Exemplo 2



Na Figura 12, temos um exemplo onde é possível a onça realizar dois movimentos de captura. Ao utilizar a função $f_{o,1}$ (Função 2.1), os dois movimentos de capturas tem o mesmo valor de utilidade, fazendo com que a escolha entre a casa 2 ou a casa 12 seja feita de forma aleatória, mesmo com a casa 12 sendo uma opção melhor por ter mais movimentos disponíveis. Já utilizando a função $f_{o,2}$ (Função 3.1), a casa escolhida será a 12, pois a função leva em consideração o número movimentos disponíveis da onça, mantendo o comportamento guloso, porém de forma mais criteriosa.

Para verificar o desempenho da função $f_{o,2}$ (Função 3.1), realizamos testes contra todas as outras funções de utilidade dos cachorros propostas pelo trabalho (NASCIMENTO; GUEDES, 2019), como mostra a Tabela 3. Os testes seguiram os mesmos parâmetros especificados pelo trabalho (NASCIMENTO; GUEDES, 2019), considerando partidas com duração máxima de 25 turnos e cada configuração de pares testada 100 vezes. Entretanto, no trabalho (NASCIMENTO; GUEDES, 2019) não foi estabelecido qual o critério para falta de combatividade, ele apenas informa que se os movimentos forem repetidos mais da metade dos turnos analisados, é declarado falta de combatividade, não especificando o que seria o movimento. Em nossos testes, consideramos os movimentos como os pares

(origem, destino). Caso esses pares se repitam em mais da metade dos turnos analisados, será declarado falta de combatividade.

Tabela 3 – Resultados obtidos utilizando a função $f_{o,2}$ (Função 3.1)

Funções de Utilidade		Profundidade	Vitórias		Partidas não concluídas	Falta de Combatividade	
Onça	Cachorros		Onça	Cachorros		Onça	Cachorros
$f_{o,2}$	$f_{c,0}$	5	30	0	70	1	1
$f_{o,2}$	$f_{c,0}$	6	32	0	68	3	0
$f_{o,2}$	$f_{c,1}$	5	46	0	54	3	2
$f_{o,2}$	$f_{c,1}$	6	36	0	64	6	0
$f_{o,2}$	$f_{c,2}$	5	15	0	85	8	3
$f_{o,2}$	$f_{c,2}$	6	19	0	81	6	1

O algoritmo para detecção da falta de combatividade analisa os movimentos realizados por cada agente (onça e cachorros) ao longo dos turnos, e verifica se houve repetição excessiva. A cada nova jogada, ele verifica se o movimento é novo ou já foi realizado anteriormente. Se o movimento é novo, ele é armazenado ao conjunto de movimentos diferentes do jogador. Já quando o movimento é repetido, contabiliza-se um ponto de repetição. Ao final, é comparado o total de movimento repetidos com a quantidade de movimentos distintos. Caso o número de repetições seja igual ou superior à metade da quantidade de movimentos diferentes, considera-se que houve falta de combatividade.

Para exemplificar o funcionamento do algoritmo, considere um cenário em que a onça realizou três movimentos: (12,18), (18,12) e (12,18). Neste caso, o par (12,18) foi repetido em dois dos três movimentos, correspondendo a mais da metade do total, sendo então considerado falta de combatividade por parte da onça. O mesmo critério é aplicado aos cachorros.

Em relação aos resultados apresentados nas Tabelas 1 e 3, a função $f_{o,2}$ (Função 3.1) apresentou um desempenho superior contra a função $f_{c,1}$ (Função 2.2) em relação a função $f_{o,1}$ (Função 2.1), com aproximadamente 41% de vitórias. Já contra os cachorros caminhando aleatoriamente pelo tabuleiro ($f_{c,0}$), a função $f_{o,2}$ (Função 3.1) apresentou um desempenho inferior, vencendo cerca de 30% das partidas, enquanto a função $f_{o,1}$ (Função 2.1) venceu em média 91,75% das partidas. Por fim, contra a função $f_{c,2}$ (Função 2.3) o desempenho de ambas as funções foram semelhantes, com aproximadamente 17% de vitórias.

Com base neste experimento, destacam-se as seguintes contribuições:

- Criação da função de utilidade $f_{o,2}$ (Função 3.1), mantendo a captura como principal critério de decisão, mas considerando a quantidade de movimentos disponíveis para tornar a estratégia mais criteriosa.
- Implementação de um algoritmo para detecção de falta de combatividade dos agentes, baseado na repetição de pares de movimentos (origem, destino) ao longo das partidas.

3.4 EXPERIMENTO 3

Com base no experimento anterior, notamos um alto número de partidas não concluídas, encerradas pelo limite de turnos. Por isso, decidimos replicar os testes do trabalho (NASCIMENTO; GUEDES, 2019) sem limitar os turnos e com profundidade limitada a 5, valor escolhido por ser o menor valor utilizado nos testes e por ter um tempo de execução inferior aos outros valores, com objetivo de encontrar um limiar mais adequado para estabelecer um limite de turnos.

Tabela 4 – Resultados obtidos sem Limite de Turnos e com Profundidade 5

Funções de Utilidade		Média de Turnos	Profundidade	Vitórias		Partidas não concluídas	Falta de Combatividade	
Onça	Cachorros			Onça	Cachorros		Onça	Cachorros
$f_{o,0}$	$f_{c,0}$	70,09	-	53	0	47	46	1
$f_{o,0}$	$f_{c,1}$	69,27	5	62	0	38	35	3
$f_{o,0}$	$f_{c,2}$	70,70	5	45	0	55	49	6
$f_{o,1}$	$f_{c,0}$	31,80	5	89	0	11	9	2
$f_{o,1}$	$f_{c,1}$	32,99	5	94	0	6	5	1
$f_{o,1}$	$f_{c,2}$	40,87	5	91	0	9	8	1
$f_{o,2}$	$f_{c,0}$	31,52	5	95	0	5	5	0
$f_{o,2}$	$f_{c,1}$	31,18	5	95	0	5	4	1
$f_{o,2}$	$f_{c,2}$	30,64	5	91	0	9	9	0

Utilizamos todas as combinações possíveis de confronto entre as funções presentes no trabalho (NASCIMENTO; GUEDES, 2019), adicionamos também a função função $f_{o,2}$ (Função 3.1), como mostra a Tabela 4. Dessa forma, as partidas só eram encerradas por vitórias ou por falta de combatividade dos agentes.

Como mostram os resultados da Tabela 4, os cachorros não obtiveram nenhuma vitória, por mais longas que tenham sido as partidas, mesmo quando enfrentaram a onça que não possuía qualquer estratégia para vencer ($f_{o,0}$). Definimos como a média de turnos o somatório dos turnos de cada partida individual dividido pelo número de partidas totais que em nossos testes será o valor de 100 partidas para cada par de teste. Observamos que o comportamento aleatório da onça ($f_{o,0}$) e a falta de efetividade dos cachorros em conseguir vencer as partidas resultaram em uma média elevada de aproximadamente 70 turnos. Enquanto, para as funções da onça que tinham uma estratégia para vencer, a média foi menor, chegando a diminuir para menos da metade em alguns casos.

A partir dos testes realizados neste experimento, optamos por criar histogramas sobre o número de turnos de cada partida, a fim de definir um novo limite de turnos maior que os 25 turnos definidos no trabalho (NASCIMENTO; GUEDES, 2019), com o objetivo de reduzir o número de partidas encerradas pelo limite de turnos. O limiar foi definido no ponto que separa os 15% maiores turnos do restante da distribuição. Esses valores representam a cauda longa da distribuição, caracterizada por partidas atípicas que duram significativamente mais do que o comportamento normal observado. Por exemplo, na

Figura 13 – Histograma - Onça ($f_{o,0}$) x Cachorros ($f_{c,0}$) - Profundidade 5

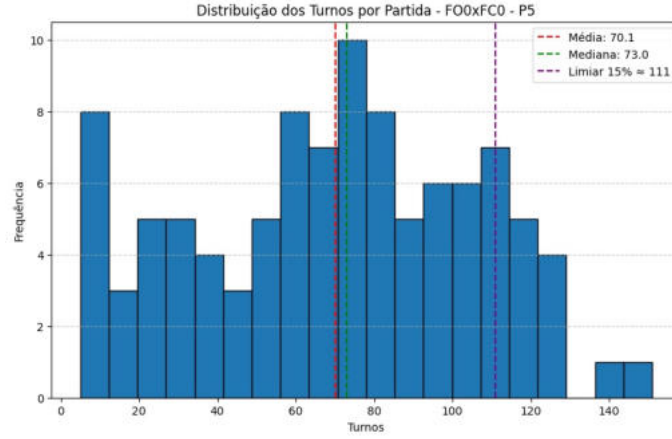
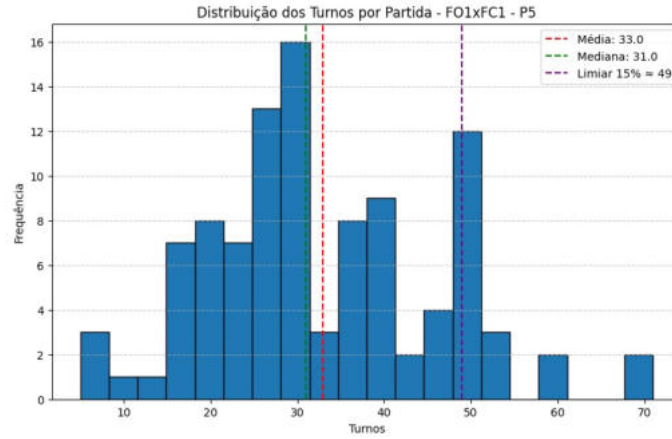


Figura 13, esse limiar de 15% é aproximadamente 111 turnos, indicando que 15% das partidas estão acima desse limite estabelecido.

Figura 14 – Histograma - Onça ($f_{o,1}$) x Cachorros ($f_{c,1}$) - Profundidade 5



Com base na Tabela 4 e nos Histogramas 13 e 14, além de outros histogramas (para todas as combinações possíveis de função que estão presentes na Tabela 4) que estão presentes no Apêndice A, e que nos auxiliaram a decidir o limite de turnos para cada função distinta da onça. Deste modo, estabelecemos que para a onça com comportamento aleatório ($f_{o,0}$) o limite seria de 110 turnos, para a onça com comportamento guiado pela função função $f_{o,1}$ (Função 2.1) o limite seria de 60 turnos, enquanto para a onça com o comportamento guiado pela função função $f_{o,2}$ (Função 3.1) o limite seria na casa de 45 turnos.

Com base neste experimento, destacam-se as seguintes contribuições:

- Avaliação de todas as combinações de confronto entre as funções de utilidade da

onça e dos cachorros, incluindo a função proposta $f_{o,2}$ (Função 3.1).

- Estabelecimento de novos limites de turnos: 110 turnos para $f_{o,0}$ (Função Aleatória), 60 turnos para $f_{o,1}$ (Função 2.1) e 45 turnos para $f_{o,2}$ (Função 3.1).

3.5 EXPERIMENTO 4

O experimento anterior mostrou que mesmo sem ter um valor limitante no número de turnos os cachorros continuam sem registrar vitórias. Então considerando o caráter distinto das estratégias das funções $f_{c,1}$ (Função 2.2) e $f_{c,2}$ (Função 2.3), decidimos combiná-las em busca de melhores resultados, pois separadamente elas não conseguiram registrar nenhuma vitória. Portanto, nesse experimento, iremos utilizar os limites de turnos encontrados no experimento anterior e ponderando os pesos w_1 e w_2 das funções $f_{c,1}$ (Função 2.2) e $f_{c,2}$ (Função 2.3) combinadas. Os pesos w_1 e w_2 das funções foram normalizados de acordo com a Equação 3.2.

$$\sum_{i=1}^n w_i = 1 \quad (3.2)$$

$$f_{c,3}(y) = w_1 \cdot f_{c,1}(y) + w_2 \cdot f_{c,2}(y) \quad (3.3)$$

As funções $f_{c,1}$ (Função 2.2) e função $f_{c,2}$ (Função 2.3) apresentam estratégias diferentes para vencer o jogo, sendo a primeira com uma estratégia mais conservadora focada em evitar capturas pela onça, enquanto a segunda tem um caráter mais agressivo focada em limitar os movimentos livres da onça, sendo assim optamos por combinar as duas, buscando melhorar os resultados, como mostrado na função $f_{c,3}$ (Função 3.3).

Primeiramente, definimos $w_1 = 0.4$ e $w_2 = 0.6$, buscando um maior equilíbrio entre as duas funções e priorizando uma estratégia mais agressiva. Posteriormente, invertemos os pesos com $w_1 = 0.6$ e $w_2 = 0.4$, priorizando uma estratégia mais conservadora.

Tabela 5 – Resultados da função $f_{c,3}$ (Função 3.3) com $w_1 = 0.4$ e $w_2 = 0.6$

Funções de Utilidade		Limite de Turnos	Profundidade	Vitórias		Partidas não concluídas	Falta de Combatividade	
Onça	Cachorros			Onça	Cachorros		Onça	Cachorros
$f_{o,0}$	$f_{c,3}$	110	5	37	1	62	42	0
$f_{o,1}$	$f_{c,3}$	60	5	88	0	12	9	0
$f_{o,2}$	$f_{c,3}$	45	5	76	0	24	11	2

Utilizando a primeira configuração de pesos e tendo uma estratégia mais agressiva conseguimos registrar a primeira vitória por parte dos cachorros em nossos testes contra a onça desprovida de estratégia para vencer ($f_{o,0}$). Registramos também outras 37 vitórias da onça e 62 partidas não concluídas, das quais 20 delas foram encerradas pelo limite de turnos e o restante por falta de combatividade da onça. Já quando a onça volta a ter

uma estratégia voltada para vencer como nas funções $f_{o,1}$ (Função 2.1) e $f_{o,2}$ (Função 3.1) voltamos a não registrar vitória por parte dos cachorros como se pode ver na Tabela 5.

Tabela 6 – Resultados da função $f_{c,3}$ (Função 3.3) com $w_1 = 0.6$ e $w_2 = 0.4$

Funções de Utilidade		Limite de Turnos	Profundidade	Vitórias		Partidas não concluídas	Falta de Combatividade	
Onça	Cachorros			Onça	Cachorros		Onça	Cachorros
$f_{o,0}$	$f_{c,3}$	110	5	50	0	50	37	1
$f_{o,1}$	$f_{c,3}$	60	5	86	0	14	4	2
$f_{o,2}$	$f_{c,3}$	45	5	76	0	24	5	1

Quando invertemos a configuração dos pesos, voltamos a não registrar vitórias por parte dos cachorros, como mostra a Tabela 6. No entanto, observamos que o limite de turnos está adequado, desprezando poucas partidas em relação ao limite de 25 turnos utilizado no trabalho (NASCIMENTO; GUEDES, 2019). Em relação as funções de utilidade dos cachorros, observamos sua baixa eficiência em vencer partidas, o que nos leva a questionar sua capacidade de representar uma estratégia satisfatória para os cachorros.

Com base neste experimento, destacam-se as seguintes contribuições:

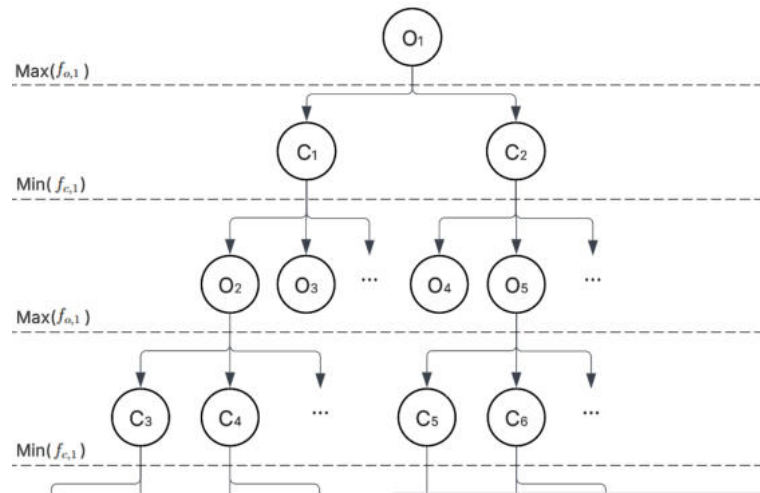
- Criação da função de utilidade combinada $f_{c,3}$ (Função 3.3), obtida pela ponderação das funções $f_{c,1}$ (Função 2.2) e $f_{c,2}$ (Função 2.3), buscando integrar estratégias conservadoras e agressivas dos cachorros.
- Avaliação de diferentes configurações de pesos (w_1 e w_2) para analisar o impacto da priorização de cada estratégia no desempenho dos cachorros.
- Registro da primeira vitória dos cachorros em nossos testes ao utilizar a configuração de pesos $w_1 = 0.4$ e $w_2 = 0.6$ contra a onça com comportamento aleatório $f_{o,1}$.

3.6 EXPERIMENTO 5

Nos experimentos anteriores, o Minimax recebia apenas uma função de utilidade tanto para maximizar, quanto para minimizar, independente do agente (onça ou cachorro) que estava jogando, i.e., tanto para avaliação do seu próprio movimento quanto para o movimento do adversário era utilizada uma mesma função. Ou seja, quando os cachorros tentavam minimizar uma jogada da onça, eles estavam usando a mesma função de avaliação que orientava seus próprios movimentos, o mesmo acontecia com a onça. Dessa forma, os cachorros não estavam avaliando o risco de serem capturados, o que pode ter levado a dificuldade em performar uma boa estratégia para vencer partidas.

Diferentemente de outros jogos como xadrez ou damas, no jogo da onça, os agentes possuem objetivos distintos, movimentações distintas e número de peças diferentes para cada agente, não sendo adequado a utilização da mesma função de utilidade para avaliar os

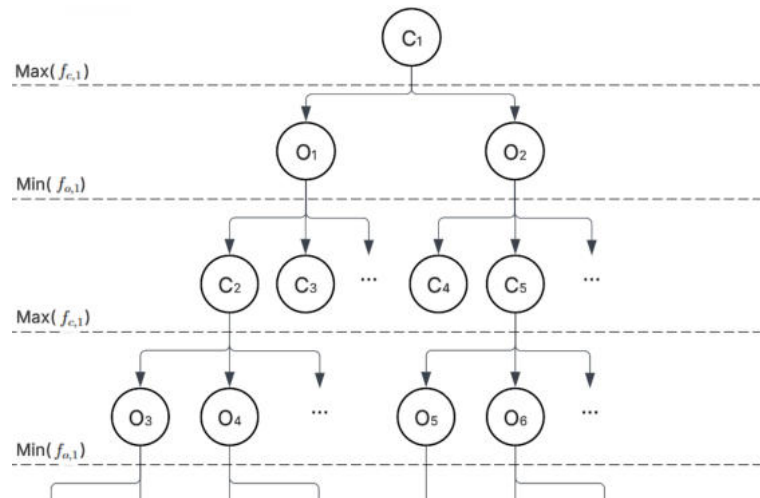
Figura 15 – Árvore de decisão da onça



nós de Max e Min de uma árvore. Além disso, o trabalho de (NASCIMENTO; GUEDES, 2019) não descreve detalhadamente como foi realizada a implementação do Minimax e como foram utilizadas as funções de utilidade.

Para contornar essa limitação, fizemos algumas modificações no Minimax, que agora recebe como entrada as duas funções de utilidades distintas, utilizadas para avaliar cada jogada de acordo com o agente correspondente. Por exemplo, em um confronto entre as funções $f_{o,1}$ (Função 2.1) e $f_{c,1}$ (Função 2.2), ao buscar a melhor jogada para onça, utilizamos duas funções de utilidade distintas, em que se busca maximizar as jogadas da onça com base na função $f_{o,1}$ (Função 2.1) e minimizar as jogadas dos cachorros com base na função $f_{c,1}$ (Função 2.2), como mostra a Figura 15.

Figura 16 – Árvore de decisão dos cachorros



Da mesma forma, analisando a perspectiva dos cachorros, ao buscar a melhor jogada para eles, as funções são aplicadas de forma inversa, buscando maximizar as jogadas dos cachorros com base na função $f_{c,1}$ (Função 2.2) e minimizar a jogada da onça com base na função $f_{o,1}$ (Função 2.1), como mostra a Figura 16. Dessa forma, não utilizando a mesma função para ambos os agentes, garantindo que os objetivos de cada agente sejam considerados de forma adequada dentro da execução do Minimax.

Após a modificação no Minimax, realizamos os testes sem limite de turnos, sendo cada configuração de pares de funções testada 100 vezes com profundidade 5, valor escolhido por ser o menor valor utilizado nos testes anteriores e por ter um tempo de execução inferior aos outros valores. Com exceção das funções $f_{o,0}$ e $f_{c,0}$ que não possuem uma função de utilidade definida, escolhendo aleatoriamente entre os movimentos disponíveis.

Tabela 7 – Resultados obtidos após modificação no Minimax

Funções de Utilidade		Pesos		Média de Turnos	Profundidade	Vitórias		Falta de Combatividade	
Onça	Cachorros	w_1	w_2			Onça	Cachorros	Onça	Cachorros
$f_{o,1}$	$f_{c,1}$	-	-	52,65	5	75	1	24	0
$f_{o,1}$	$f_{c,2}$	-	-	68,69	5	47	2	50	1
$f_{o,1}$	$f_{c,3}$	0,4	0,6	62,08	5	41	4	55	0
$f_{o,1}$	$f_{c,3}$	0,6	0,4	55,33	5	54	0	45	1
$f_{o,2}$	$f_{c,1}$	-	-	50,19	5	83	0	13	4
$f_{o,2}$	$f_{c,2}$	-	-	62,26	5	58	4	37	1
$f_{o,2}$	$f_{c,3}$	0,4	0,6	50,38	5	48	2	47	3
$f_{o,2}$	$f_{c,3}$	0,6	0,4	49,49	5	58	0	41	1

Analisando os resultados apresentados na Tabela 7, observa-se uma melhora no desempenho dos cachorros em relação aos experimentos anteriores, com eles apresentando um número maior de vitórias, com destaque para os confrontos $f_{o,1}$ (Função 2.1) x $f_{c,3}$ (Função 3.3) e $f_{o,2}$ (Função 3.1) x $f_{c,2}$ (Função 2.3), ambos com 4 vitórias dos cachorros. Além disso, tivemos um número muito grande de partidas finalizadas por falta de combatividade da onça, sendo um indício que os cachorros estariam oferecendo uma maior resistência a onça.

Com base neste experimento, destacam-se as seguintes contribuições:

- Modificação do algoritmo Minimax para utilizar duas funções de utilidade distintas, permitindo que cada agente (onça e cachorros) avalie as jogadas de acordo com seus próprios objetivos.
- Observação de melhora no desempenho dos cachorros em relação aos experimentos anteriores, registrando aumento no número de vitórias.

3.7 EXPERIMENTO 6

No experimento 5, tivemos um grande número de partidas não finalizadas por falta de combatividade da onça. Por esse motivo, decidimos replicar o experimento 5 com os mesmos parâmetros, porém sem a verificação de falta de combatividade. Ao realizar os testes, todas as partidas foram finalizadas normalmente.

Esse comportamento difere do que foi descrito no trabalho (NASCIMENTO; GUEDES, 2019), onde foi observado um alto número de repetições de movimentos por parte de ambos os agentes, com partidas presas em um ciclo. Comportamento esse que foi a motivação da adoção da falta de combatividade. Contudo, em nosso trabalho, não observamos partidas presas em um ciclo, devido o critério estabelecido de desempate entre os movimentos que possuem o mesmo valor para a função de utilidade, com as partidas não apresentando um comportamento de ciclo dos movimentos. Sendo assim, não foi necessário a utilização de métodos para verificação da falta de combatividade dos agentes.

Tabela 8 – Resultados obtidos após modificação no Minimax e sem falta de combatividade

Funções de Utilidade		Pesos		Média de Turnos	Profundidade	Vitórias	
Onça	Cachorros	w_1	w_2			Onça	Cachorros
$f_{o,1}$	$f_{c,1}$	-	-	63,04	5	99	1
$f_{o,1}$	$f_{c,2}$	-	-	98,93	5	95	5
$f_{o,1}$	$f_{c,3}$	0,4	0,6	82,71	5	99	1
$f_{o,1}$	$f_{c,3}$	0,6	0,4	69,50	5	100	0
$f_{o,2}$	$f_{c,1}$	-	-	60,30	5	100	0
$f_{o,2}$	$f_{c,2}$	-	-	91,45	5	97	3
$f_{o,2}$	$f_{c,3}$	0,4	0,6	78,68	5	98	2
$f_{o,2}$	$f_{c,3}$	0,6	0,4	71,16	5	99	1

Como mostram os resultados na Tabela 8, mesmo sem a falta de combatividade, os cachorros apresentaram um número de vitórias próximo ao experimento anterior. Destaca-se a função $f_{c,2}$ (Função 2.3), que obteve o maior número de vitória, 5 e 3, contra as funções $f_{o,1}$ (Função 2.1) e $f_{o,2}$ (Função 3.1), respectivamente. Além disso, a função $f_{c,2}$ (Função 2.3) registrou a maior média de turnos, demonstrando maior resistência contra a onça.

Com base neste experimento, destacam-se as seguintes contribuições:

- Verificação de que o critério de desempate entre movimentos com mesmo valor de utilidade evitou ciclos, permitindo que todas as partidas fossem finalizadas normalmente mesmo sem a verificação de falta de combatividade.
- Observação de que os cachorros mantiveram um número de vitórias semelhante ao experimento anterior, mesmo sem a verificação de falta de combatividade.

3.8 EXPERIMENTO 7

Nesse experimento, temos como objetivo testar as funções dos cachorros contra a função $f_{o,0}$, que não possui função de utilidade definida, selecionando aleatoriamente entre os movimentos disponíveis. Com a modificação do Minimax, que precisa receber duas funções de utilidade, utilizamos a função $f_{o,1}$ (Função 2.1) para a execução do algoritmo,

já que a função $f_{o,0}$ não possui uma. Assim, o Minimax recebe a função dos cachorros e tenta maximizá-la, enquanto recebe a função $f_{o,1}$ (Função 2.1) e tenta minimizá-la.

Os testes foram realizados seguindo o mesmo padrão do Experimento 6, sem limite de turnos, sendo cada configuração de pares de funções testada 100 vezes com profundidade 5 e sem partidas finalizadas por falta de combatividade.

Tabela 9 – Resultados obtidos contra a onça aleatória após modificação no Minimax e sem falta de combatividade

Funções de Utilidade		Pesos		Média de Turnos	Profundidade	Vitórias	
Onça	Cachorros	w_1	w_2			Onça	Cachorros
$f_{o,0}$	$f_{c,1}$	-	-	91,12	5	95	5
$f_{o,0}$	$f_{c,2}$	-	-	95,85	5	97	3
$f_{o,0}$	$f_{c,3}$	0,4	0,6	90,55	5	95	5
$f_{o,0}$	$f_{c,3}$	0,6	0,4	94,53	5	91	9

Observando os resultados apresentados na Tabela 9, verifica-se um maior número de vitória em comparação aos outros experimentos, com o confronto entre $f_{o,0}$ e $f_{c,3}$ (Função 3.3) com os pesos $w_1 = 0.6$ e $w_2 = 0.4$, alcançando 9 vitórias. Embora tenha tido um aumento no número de vitórias por parte dos cachorros, ainda existe uma dificuldade em apresentar um desempenho satisfatório contra um adversário desprovido de uma estratégia para vencer. Outro ponto relevante foi o desempenho inferior da função $f_{c,2}$ (Função 2.3) em relação às outras funções, comportamento que não havia sido observado nos experimentos anteriores.

Com base neste experimento, destacam-se as seguintes contribuições:

- Adaptação da execução do Minimax para esse cenário, utilizando a função $f_{o,1}$ (Função 2.1) para minimizar as jogadas da onça, uma vez que a $f_{o,0}$ não possui função de utilidade definida.
- Identificação de que, apesar do aumento nas vitórias, os cachorros ainda apresentam dificuldade em obter desempenho consistente, mesmo contra um adversário sem estratégia definida.

3.9 EXPERIMENTO 8

Neste experimento, temos como o objetivo conceder uma vantagem no número de jogadas que um agente enxerga à frente do outro, buscando equilibrar a disputa entre eles. Dessa forma, reduzimos a profundidade da onça para 3, enquanto a profundidade dos cachorros foi aumentada para 6.

Além disso, por conta do custo de tempo para a execução dos testes, só foi levado em consideração a função mais ingênua da onça $f_{o,1}$ (Função 2.1), contra todas as funções

disponíveis para os cachorros, com exceção da função $f_{c,0}$ (Aleatória). O restante dos parâmetros continuam os mesmos do Experimento 6, pois não foi utilizado limite de turnos nem falta de combatividade, e também foram deixadas de fora as funções de comportamento aleatório para ambos os agentes.

Tabela 10 – Resultados com onça em profundidade 3 e cachorros em profundidade 6

Funções de Utilidade		Pesos		Média de Turnos	Profundidade		Vitórias	
Onça	Cachorros	w_1	w_2		Onça	Cachorros	Onça	Cachorros
$f_{o,1}$	$f_{c,1}$	-	-	63,53	3	6	100	0
$f_{o,1}$	$f_{c,2}$	-	-	139,76	3	6	96	4
$f_{o,1}$	$f_{c,3}$	0,4	0,6	91,73	3	6	98	2
$f_{o,1}$	$f_{c,3}$	0,6	0,4	87,82	3	6	99	1

Observando os resultado obtidos na Tabela 10, podemos notar, em comparação com os experimentos anteriores, que não houve uma melhoria significativa no número de vitórias dos cachorros. Inclusive, com a $f_{c,1}$ (Função 2.2) não conquistando nenhuma vitória nas 100 partidas e tendo uma média de turnos próxima a do Experimento 6, onde as profundidades eram as mesmas para ambos os agentes. A função com melhor desempenho foi $f_{c,2}$ (Função 2.3), obtendo 4 vitórias nas 100 partidas disputadas e tendo uma elevada média de turnos, em relação as outras funções, o que pode indicar uma maior resistência do que no Experimento 6.

Continuando com os testes, com o intuito de verificar a influência da profundidade no desempenho dos agentes, optamos por realizar um segundo conjunto de confrontos entre as funções, utilizando a configuração inversa de profundidades, com profundidade 6 para onça e profundidade 3 para os cachorros.

Tabela 11 – Resultados com onça em profundidade 6 e cachorros em profundidade 3

Funções de Utilidade		Pesos		Média de Turnos	Profundidade		Vitórias		Partidas não concluídas
Onça	Cachorros	w_1	w_2		Onça	Cachorros	Onça	Cachorros	
$f_{o,1}$	$f_{c,1}$	-	-	46,46	6	3	100	0	0
$f_{o,1}$	$f_{c,2}$	-	-	532,14	6	3	73	12	15
$f_{o,1}$	$f_{c,3}$	0,4	0,6	608,19	6	3	62	7	31
$f_{o,1}$	$f_{c,3}$	0,6	0,4	601,18	6	3	65	13	22

Analisando os resultados apresentados na Tabela 11, observa-se que algumas partidas não foram concluídas. Embora este experimento tenha sido configurado sem um limite de turnos explícito, alguma partidas alcançaram o valor máximo padrão do sistema para encerramento da partida por turnos, estabelecido em 999 turnos. Sendo um comportamento inédito, pois não aconteceu em nenhum outro experimento, esse comportamento sugere a ocorrência de um estado de equilíbrio defensivo persistente ou de uma ausência de movimentos decisivos por um período prolongado.

Além disso, podemos observar uma média elevada de turnos em todos os confrontos, exceto contra a função $f_{c,1}$ (Função 2.2), que novamente não obteve nenhuma vitória. Sendo um comportamento similar aos apresentados na Tabela 10, indicando que a profundidade não tem um impacto significativo no desempenho da $f_{c,1}$ (Função 2.2) contra a função $f_{o,1}$ (Função 2.1). Em relação as outras funções, elas apresentaram o maior número de vitórias, entre todos os experimentos, com a função $f_{c,3}$ (Função 3.3), com os pesos $w_1 = 0.6$ e $w_2 = 0.4$, apresentando 13 vitórias.

No geral, os resultados apresentados nas Tabelas 10 e 11, sugerem um impacto negativo do aumento da profundidade no comportamento dos agentes. Por exemplo, a onça apresenta um desempenho melhor com profundidade 3 em comparação com profundidade 6, o que reflete diretamente no número de vitórias. O mesmo padrão também é observado para os cachorros, indicando que profundidades mais altas, em vez de melhorar as decisões, acabam prejudicando o desempenho de ambos os agentes.

Com base neste experimento, destacam-se as seguintes contribuições:

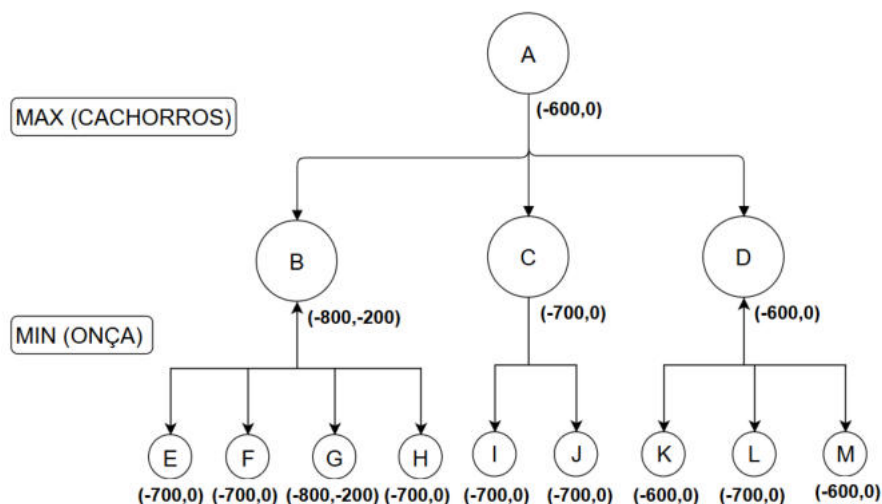
- Identificação de partidas extremamente longas, atingindo o limite máximo de 999 turnos, indicando possíveis estados de equilíbrio defensivo ou ausência prolongada de movimentos decisivos.
- Verificação de que profundidades maiores podem impactar negativamente o desempenho dos agentes, sugerindo que o aumento da profundidade nem sempre resulta em decisões melhores.

3.10 EXPERIMENTO 9

Analizando os resultados dos experimentos anteriores, observamos um baixo desempenho dos cachorros, com poucas vitórias. Com isso, decidimos realizar uma nova modificação no funcionamento do algoritmo Minimax, de modo que ele passe a avaliar os nós terminais por meio de duas funções de utilidade, sendo uma referente aos cachorros e outra à onça, e a utilidade desses nós é expressa como uma tupla contendo os valores dessas duas funções. Além disso, não é possível utilizar poda Alpha-Beta após essa modificação, pois haveria perda de informação de uma das funções de utilidade, o que torna a profundidade mais limitada em relação aos experimentos anteriores.

Por exemplo, na Figura 17, temos uma árvore que busca encontrar a melhor jogada para os cachorros e possui profundidade 2. Cada nó terminal é avaliado por duas funções, denominadas f_c (ponto de vista dos cachorros) e f_o (ponto de vista da onça), e sua utilidade é expressa como uma tupla (f_c, f_o) . Como exemplo, o nó G que possui utilidade $(-800, -200)$, representando, respectivamente, os valores retornados pelas duas funções. Após avaliação dos nós terminais, o algoritmo prossegue segundo a lógica do Minimax, com o nível Min selecionando os menores valores do segundo componente da tupla de

Figura 17 – Exemplo Funcionamento Minimax



utilidade, correspondente à função f_o , e esses valores são propagados para os nós de cima da árvore. Em seguida, no nível Max, os cachorros escolhem o maior valor do primeiro componente da tupla de utilidade, correspondente à função f_c levando em conta as escolhas feitas no nível anterior pela onça.

Cabe ainda destacar que, em situações de empate entre os valores dos componentes das tuplas, a escolha é realizada aleatoriamente entre os valores empatados, a fim de assegurar variabilidade estratégica e evitar decisões determinísticas.

Os testes foram realizados com limite de 999 turnos (valor máximo padrão do sistema), sendo que cada configuração de pares de funções foi testada 100 vezes, com profundidade 4 para os cachorros e profundidade 3 para onça, pois nos experimentos anteriores a onça já havia demonstrado vantagem em relação aos cachorros. Além disso, não houve partidas finalizadas por falta de combatividade.

Tabela 12 – Resultados com a nova modificação no Minimax

Funções de Utilidade		Pesos		Média de Turnos	Profundidade		Vitórias		Partidas não concluídas
Onça	Cachorros	w_1	w_2		Onça	Cachorros	Onça	Cachorros	
$f_{o,0}$	$f_{c,1}$	-	-	111,75	x	4	100	0	0
$f_{o,0}$	$f_{c,2}$	-	-	270,21	x	4	23	73	4
$f_{o,0}$	$f_{c,3}$	0,4	0,6	218,95	x	4	25	75	0
$f_{o,0}$	$f_{c,3}$	0,6	0,4	197,69	x	4	39	60	1
$f_{o,1}$	$f_{c,1}$	-	-	38,32	3	4	100	0	0
$f_{o,1}$	$f_{c,2}$	-	-	93,96	3	4	77	21	2
$f_{o,1}$	$f_{c,3}$	0,4	0,6	96,62	3	4	77	22	1
$f_{o,1}$	$f_{c,3}$	0,6	0,4	55,95	3	4	82	18	0
$f_{o,2}$	$f_{c,1}$	-	-	33,87	3	4	100	0	0
$f_{o,2}$	$f_{c,2}$	-	-	178,88	3	4	80	6	14
$f_{o,2}$	$f_{c,3}$	0,4	0,6	148,95	3	4	86	5	9
$f_{o,2}$	$f_{c,3}$	0,6	0,4	214,4	3	4	80	2	18

Analisando os resultados apresentados na Tabela 12, observa-se que as funções dos cachorros tiveram um elevado número de vitórias contra a função aleatória da onça ($f_{o,0}$) em relação aos outros experimentos, com exceção da função $f_{c,1}$ (Função 2.2) que não obteve nenhuma vitória. As demais funções apresentaram o melhor desempenho entre todos os experimentos realizados, com destaque para as funções $f_{c,3}$ (Função 3.3), com os pesos $w_1 = 0.4$ e $w_2 = 0.6$, e $f_{c,2}$ (Função 2.3), que obtiveram, respectivamente, 75 e 73 vitórias.

Contra as demais funções da onça, os resultados foram semelhantes, os cachorros obtiveram o maior número de vitórias em comparação aos outros experimentos. A função $f_{c,1}$ (Função 2.2) continuou sem nenhuma vitória, enquanto as funções $f_{c,3}$ (Função 3.3), com os pesos $w_1 = 0.4$ e $w_2 = 0.6$, e $f_{c,2}$ (Função 2.3), continuaram apresentando os melhores desempenho entre as outras funções dos cachorros. Já entre as funções da onça, destacou-se a função $f_{o,2}$ (Função 3.1), apresentando o maior número de vitórias e o menor número de derrotas em comparação às demais funções da onça.

Com base neste experimento, destacam-se as seguintes contribuições:

- Implementação de uma nova modificação no algoritmo Minimax, permitindo que os nós terminais sejam avaliados por duas funções de utilidade distintas, uma para os cachorros e outra para a onça.
- Identificação da impossibilidade de utilização da poda Alpha-Beta após a modificação, devido à perda de informação associada às duas funções de utilidade.
- Observação de melhora significativa no desempenho dos cachorros, com aumento expressivo no número de vitórias em relação aos experimentos anteriores.

3.11 ANÁLISE GERAL

No geral, observando os experimentos, as funções dos cachorros não apresentaram um desempenho competitivo quando confrontadas com as estratégias das onças. Com a função $f_{c,2}$ (Função 2.3) conseguindo os melhores números, com 8,9% de vitórias e 66,9% de derrotas nas partidas que disputou. Analisando os experimentos de forma individual, ela também foi a função dos cachorros que mais se destacou, como por exemplo nos experimentos 6 e 8, nos quais obteve 8 e 16 vitórias, respectivamente. Já a função $f_{c,1}$ (Função 2.2) apresentou o pior desempenho entre as funções dos cachorros, com apenas 0,5% de vitórias e 85,7% de derrotas das partidas que disputou, algo que também fica evidente na análise individual dos experimentos.

Em relação a $f_{c,3}$ (Função 3.3), com os pesos $w_1 = 0.4$ e $w_2 = 0.6$, ela se mostrou mais competitiva do que com os pesos $w_1 = 0.6$ e $w_2 = 0.4$. Embora ambos os casos apresentem percentuais de vitórias próximos, observa-se um desempenho ligeiramente superior com os pesos $w_1 = 0.4$ e $w_2 = 0.6$, com 9,7% de vitórias, em comparação aos 8,0% obtidos

Tabela 13 – Resultados Gerais das funções de utilidade

Função de Utilidade	Pesos		Número de Partidas	Vitórias (Qtd.)	Derrotas (Qtd.)	Vitórias (%)	Derrotas (%)	Partidas não Concluídas (Qtd.)	Partidas não Concluídas (%)
	w_1	w_2							
$f_{o,1}$	-	-	2500	2087	114	83,5%	4,6%	299	12,0%
$f_{o,2}$	-	-	2300	1598	25	69,5%	1,1%	677	29,4%
$f_{c,1}$	-	-	1500	7	1285	0,5%	85,7%	208	13,9%
$f_{c,2}$	-	-	1500	133	1004	8,9%	66,9%	363	24,2%
$f_{c,3}$	0,4	0,6	1300	126	930	9,7%	71,5%	244	18,8%
$f_{c,4}$	0,6	0,4	1300	104	979	8,0%	75,3%	217	16,7%

com os pesos $w_1 = 0.6$ e $w_2 = 0.4$. Além disso, com os pesos $w_1 = 0.4$ e $w_2 = 0.6$, houve um número menor de derrotas (71,5%), em comparação aos 75,3% obtidos com $w_1 = 0.6$ e $w_2 = 0.4$, como mostra a Tabela 13.

As funções da onça mostraram um comportamento bastante competitivo, apresentando bons percentuais de vitórias e baixos percentuais de derrota. A princípio a $f_{o,1}$ (Função 2.1) se destaca pelo maior número de vitórias em relação a $f_{o,2}$ (Função 3.1). Contudo, esses valores não refletem exatamente o desempenho entre elas, pois a função $f_{o,2}$ (Função 3.1) teve um número muito maior de partidas não concluídas, principalmente devido ao experimento 2, que possuía um limite baixo de turnos, impactando seu resultado geral.

Tabela 14 – Resultados comparativo entre as funções da onça

Função de Utilidade	Número de Partidas	Vitórias (Qtd.)	Derrotas (Qtd.)	Vitórias (%)	Derrotas (%)	Partidas não Concluídas (Qtd.)	Partidas não Concluídas (%)
$f_{o,1}$	1700	1394	75	82,0%	4,4%	231	13,6%
$f_{o,2}$	1700	1420	25	83,5%	1,5%	255	15,0%

Além disso, no experimento 8 foi utilizada apenas a função $f_{o,1}$ (Função 2.1), e nesse experimento os cachorros alcançaram o maior número de vitórias, também impactando no resultado geral. Por esse motivo, a Tabela 14 desconsidera os experimentos 2 e 8, evidenciando um desempenho ligeiramente superior da função $f_{o,2}$ (Função 3.1), com 83,5% de vitórias e 1,5% de derrotas em relação a $f_{o,1}$ (Função 2.1) com 82,0% de vitórias e 4,4% de derrotas.

4 ANÁLISE QUALITATIVA

Neste capítulo, serão analisados de forma qualitativa os experimentos do Capítulo 3, com foco na interpretação dos resultados obtidos e na identificação de padrões relevantes. O objetivo principal é identificar padrões nas partidas em que os cachorros obtiveram vitórias, afim de propor alguma nova função de utilidade.

Figura 18 – Trecho do log gerado no Experimento 5

```

Turno 14: Cachorros: 6 → 10
Possíveis jogadas: [0, 10]
Valor Utilidade: 200
Tempo: 0.1757s
Nós avaliados: 11008
Estado Atual do Tabuleiro: [[0, 1, 1, 1, 1, 1, 0, 1, 0, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, -1, 0, 0, 0, 0, 0, 0, 0, 0]]
Turno 15: Onça: 22 → 17
Possíveis jogadas: [16, 17, 18, 21, 23, 25, 26, 27]
Valor Utilidade: 800
Tempo: 0.1184s
Nós avaliados: 6770
Estado Atual do Tabuleiro: [[0, 1, 1, 1, 1, 1, 0, 1, 0, 1, 1, 1, 1, 1, 1, 0, -1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]]
Turno 16: Cachorros: 11 → 16
Possíveis jogadas: [6, 16]
Valor Utilidade: 200
Tempo: 0.2099s
Nós avaliados: 12965
Estado Atual do Tabuleiro: [[0, 1, 1, 1, 1, 1, 0, 1, 0, 1, 1, 0, 1, 1, 1, 1, -1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]]

```

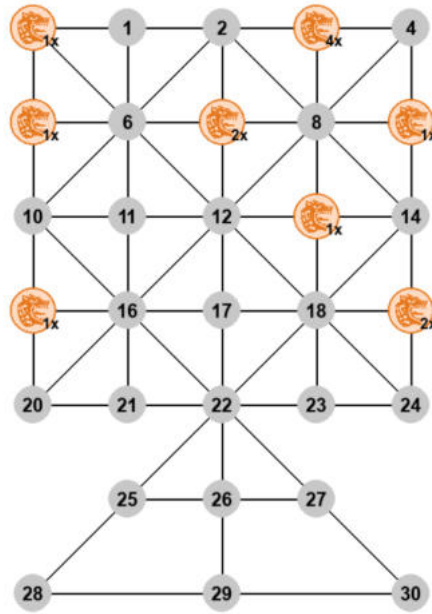
As análises serão realizadas a partir do Experimento 5, pois foi a partir dele que implementamos os registros individuais de cada partida. Essas análises foram feitas simultaneamente com os Experimentos a fim de identificar melhorias para os próximos. Esses registros armazenam as jogadas de cada agente, o valor da função de utilidade associado a cada jogada, todos os movimentos possíveis que o agente poderia realizar, o tempo de execução da ação, entre outras informações, como mostra a Figura 18. Além disso, adicionamos uma nova funcionalidade na interface do jogo, onde é possível reproduzir as partidas através dos registros, com o objetivo de facilitar as análises.

Como primeira parte da análise, buscamos analisar as casas em que a onça fica presa ao fim do jogo. A ideia é tentar encontrar algum padrão entre as casas mais comuns em que a onça foi encurralada.

4.1 EXPERIMENTO 5

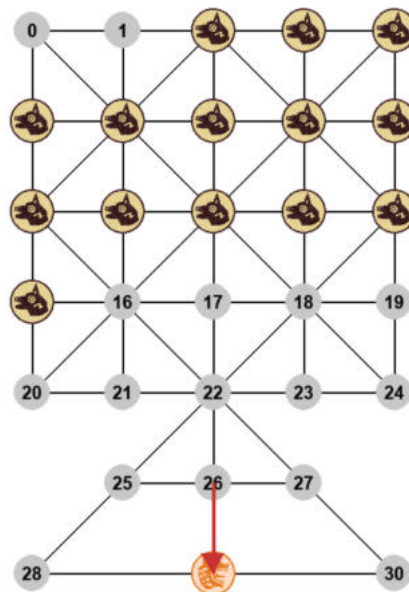
No Experimento 5, tivemos ao todo, 13 vitórias dos cachorros, considerando todos os confrontos realizados. Além disso, as casas em que a onça foi encurralada, foram as casas 0, 3, 5, 7, 9, 13, 15 e 19, sendo encurralada quatro vezes na casa 3, duas vezes na casa 19 e duas vezes na casa 7, nas demais casas ela foi encurralada apenas uma vez, como mostra a Figura 19. Dentre as casas citadas, seis (0, 3, 5, 9, 15 e 19) delas são casas com apenas três arestas e as outras duas (7 e 13) com quatro arestas.

Figura 19 – Distribuição das posições em que a onça foi derrotada no Experimento 5



Analisaremos uma das partidas em que os cachorros obtiveram vitória encurralando a onça na casa 7. Essa partida foi escolhida, pois a casa 7 possui quatro arestas, sendo mais difícil de conseguir prender a onça. Além disso, a partida tem os parâmetros de teste presentes no Experimento 5, utilizando as funções $f_{o,2}$ (Função 3.1) e $f_{c,2}$ (Função 2.3), para ditar o comportamento dos agentes.

Figura 20 – Turno 17 (Onça) - Partida do Experimento 5



Após o início da partida, a onça assume um comportamento de se afastar, enquanto os cachorros procuram permanecer em bloco, agindo de forma defensiva, como mostra a Figura 20. Ao chegar no turno 26, os cachorros abrem uma possibilidade de captura para a onça (Figura 21), que em um primeiro momento não realiza a captura até o turno 31, onde realiza sua primeira captura (Figura 22). Após isso, os cachorros dão mais duas possibilidades de captura para onça, porém ela não efetua nenhuma captura (Figura 23).

Figura 21 – Turno 26 (Cachorros) e Turno 27 (Onça)

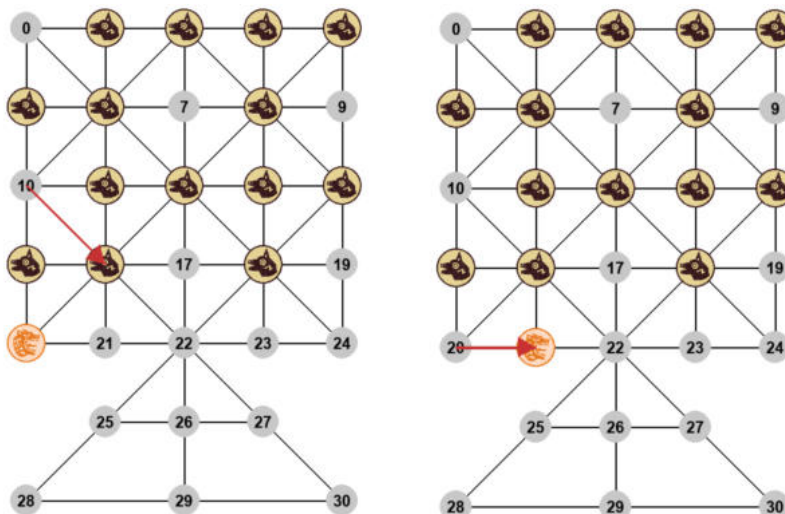
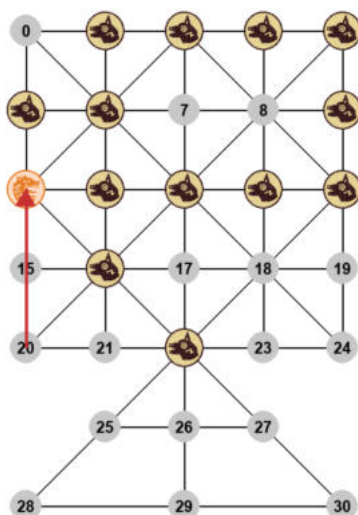


Figura 22 – Turno 31 (Onça) - Captura



Nos movimentos seguintes, a onça realiza uma captura do cachorro que está na casa 12, se movimentando para casa 8 (Figura 24). Ao fim do turno 36 existem 2 possibilidades de captura para onça, um cachorro na casa 12 e um cachorro na casa 13, porém ela não opta pela captura, o que viria a se tornar um erro decisivo na partida (Figura 25). Nos turnos

seguintes, ela ainda teria sua última chance de sair do cerco que estava se formando pelos cachorros capturando o cachorro na casa 12 ou 13, porém ela optou por se movimentar para a casa 7 (Figura 26) e no turno 42 foi encurralada (Figura 27), encerrando assim a partida.

Figura 23 – Turno 32 (Cachorros) e Turno 33 (Onça)

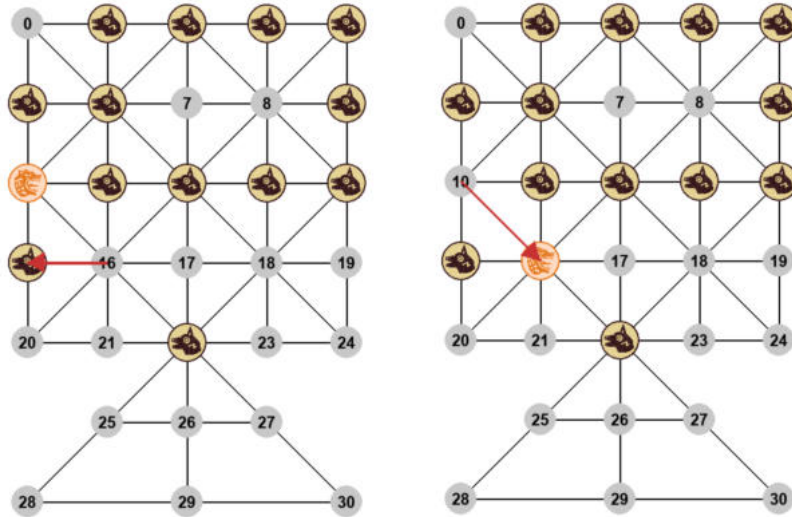
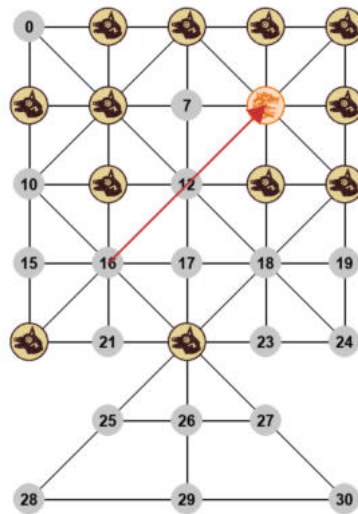


Figura 24 – Turno 35 (Onça) - Captura



Com isso, podemos ver um comportamento levemente errático da onça e os cachorros se arriscando bastante para conseguir a vitória, o que pode justificar o baixo número de vitórias dos cachorros nesse experimento. Podemos ver a dificuldade da função $f_{c,2}$ (Função 2.3) em se manter em bloco ao mesmo tempo em que busca uma estratégia mais agressiva de cerco.

Figura 25 – Turno 36 (Cachorros) e Turno 37 (Onça)

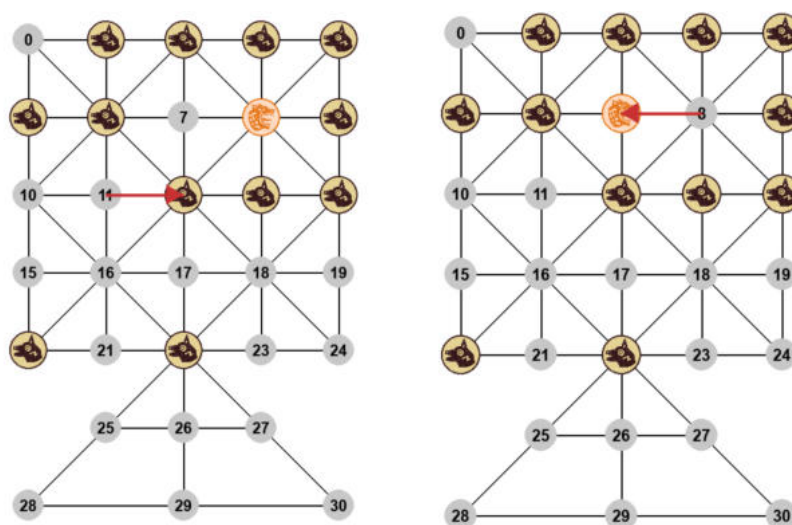


Figura 26 – Turno 40 (Cachorros) e Turno 41 (Onça)

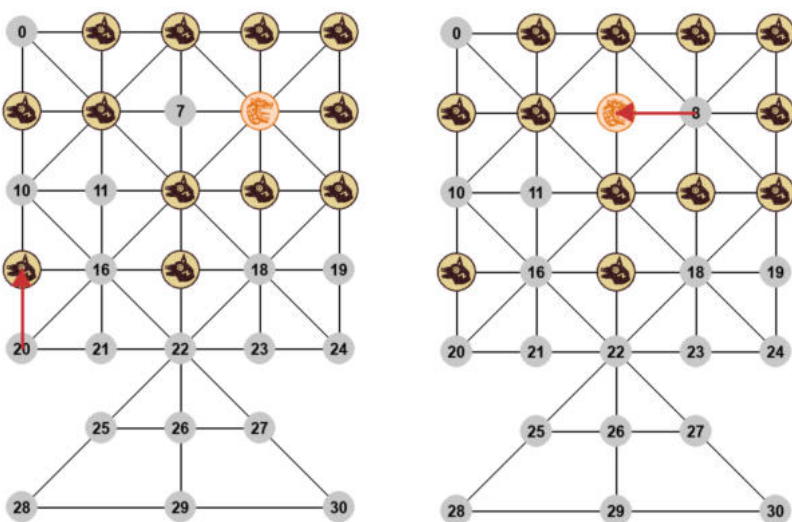
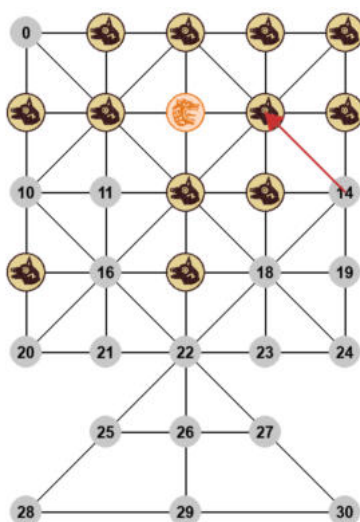


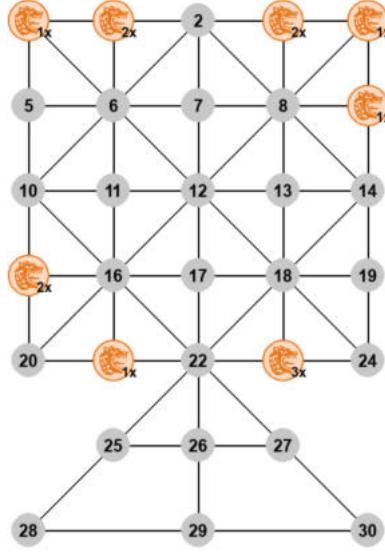
Figura 27 – Turno 42 (Cachorros) - Vitória dos Cachorros



4.2 EXPERIMENTO 6

Já no Experimento 6, outras 13 vitórias dos cachorros foram registradas ao todo. Além disso, as casas em que a onça foi encurralada, foram as casas 0, 1, 3, 4, 9, 15, 21 e 23, sendo três vezes na casa 23, duas vezes na casa 1, duas vezes na casa 3 e duas vezes na casa 15, nas demais casa ela foi encurralada apenas uma vez, como mostra a Figura 28. Todas as casas em que a onça foi encurralada nesse experimento possuem três arestas.

Figura 28 – Distribuição das posições em que a onça foi derrotada no Experimento 6



Para o Experimento 6, vamos analisar uma das três partidas em que a onça terminou presa na casa 23. Além dos parâmetros presentes no Experimento 6, as funções de utilidade que guiam os movimentos dos agentes são, a função $f_{o,1}$ (Função 2.1) e a função $f_{c,2}$ (Função 2.3).

Com um início de partida sem muitos movimentos interessantes, temos uma primeira imagem do jogo ao fim do turno 40 (Figura 29), onde a onça se encontra dentro da popularmente chamada toca da onça (parte triangular do tabuleiro).

A onça se mantém sem conseguir sair de lá, mas também sem um cerco estar se formando para cima dela até o turno 132 (Figura 29), onde ela se encontra na casa 22, tendo a possibilidade de realizar uma captura, sobre o cachorro na casa 26, porém ela opta para se mover para a casa 23.

No turno 136 (Figura 30), podemos ver um cerco se aproximando dela no canto inferior direito do tabuleiro, com apenas as casas 18 e 23 livres. Após mais uma permutação de posições, podemos ver que o cerco se mantém sobre a onça no turno 146 (Figura 30).

No turno 150 (Figura 31), os cachorros fazem um movimento decisivo que coloca a onça em xeque, movimentando um cachorro da casa 19 para a casa 24, com a onça posicionada

Figura 29 – Turno 40 (Cachorros) e Turno 132 (Cachorros)

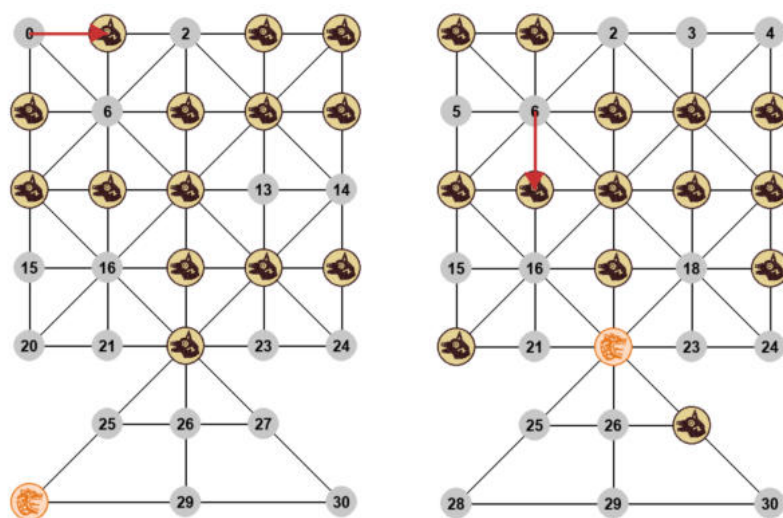
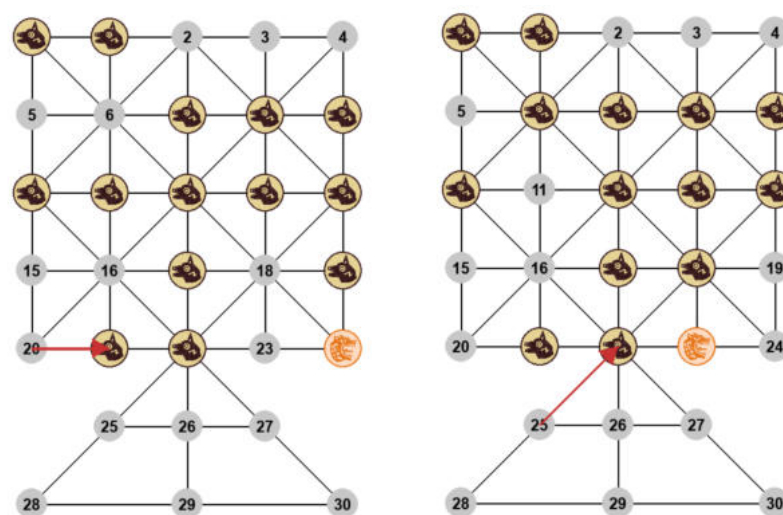


Figura 30 – Turno 136 (Cachorros) e Turno 146 (Cachorros)



na casa 18, resta para ela se movimentar para a casa 19 ou para a casa 23, ela opta pela casa 23 e os cachorros finalizam a partida movimentando o cachorro da casa 14 para a casa 18, prendendo a onça na casa 23 (Figura 32).

Figura 31 – Turno 149 (Onça) e Turno 150 (Cachorros)

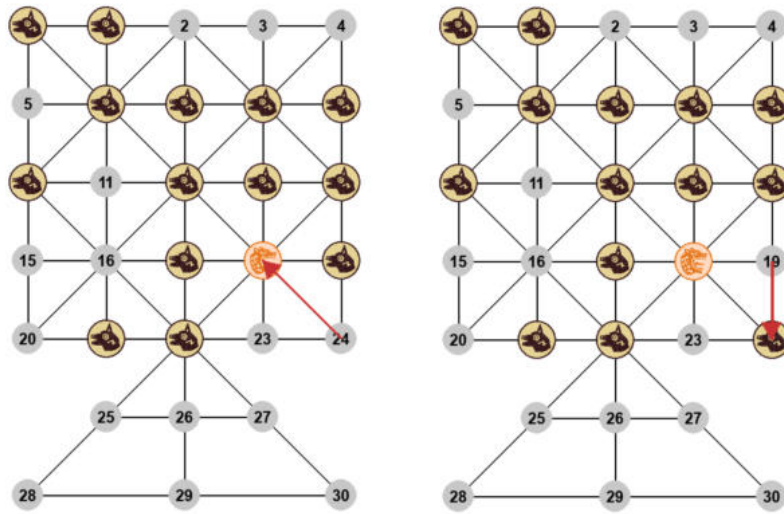
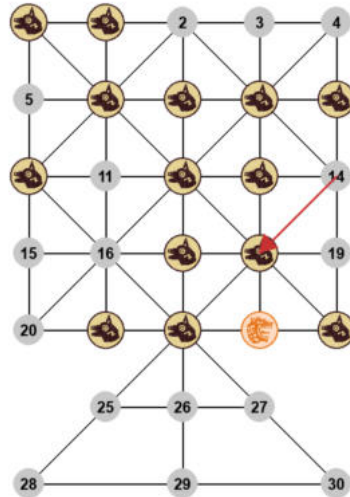


Figura 32 – Turno 152 (Cachorros) - Vitória dos Cachorros



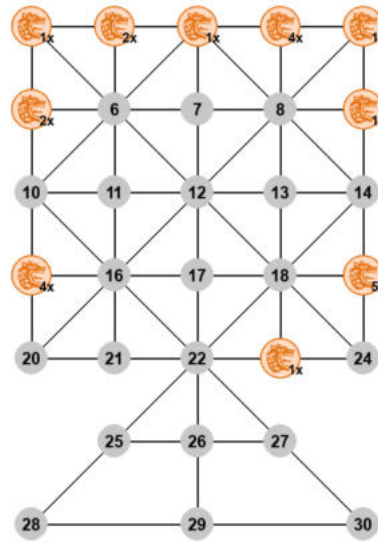
Caso a onça tivesse optado pela casa 19 anteriormente, abriria a possibilidade dos cachorros se movimentarem da casa 13 para a casa 18 e também iriam prender a onça. Vale ressaltar que essa vitória dos cachorros aconteceu sem ter nenhum cachorro capturado.

4.3 EXPERIMENTO 7

No Experimento 7, foram 22 vitórias dos cachorros ao todo. Além disso, as casas em que a onça foi encurralada, foram as casas 0, 1, 2, 3, 4, 5, 9, 15, 19 e 23, sendo cinco vezes na casa 19, quatro vezes na casa 3, quatro vezes na casa 15, duas vezes na casa 1 e duas vezes na casa 5, nas demais casas ela foi encurralada apenas uma vez, como mostra a Figura 33. Dentre as casas citadas, apenas a casa 2 possui cinco arestas, enquanto todas as outras possuem apenas três arestas.

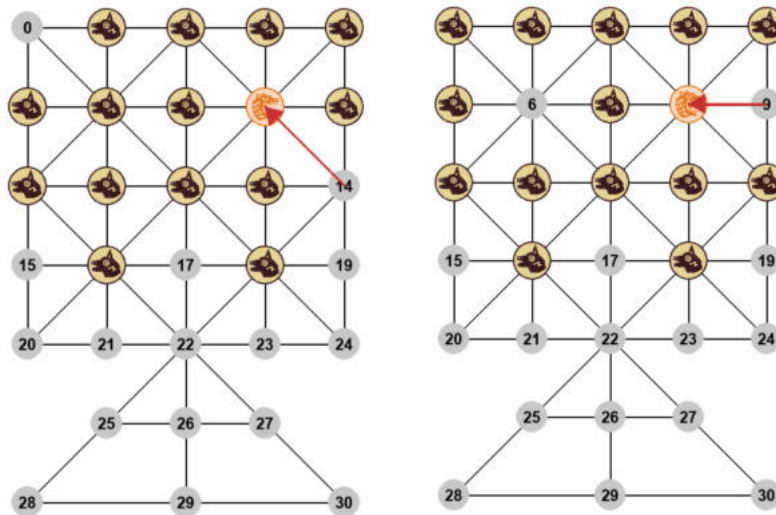
Para o Experimento 7, vamos analisar o jogo em que a onça ficou presa na casa 2, por

Figura 33 – Distribuição das posições em que a onça foi derrotada no Experimento 7



essa possuir o maior número de arestas em que a onça se encontrou presa até então, com cinco arestas, o que é um grau de dificuldade elevado. Além disso, as funções de utilidade que guiam os movimentos dos agentes são, a função $f_{o,0}$, de comportamento aleatório e novamente a função $f_{c,2}$ (Função 2.3).

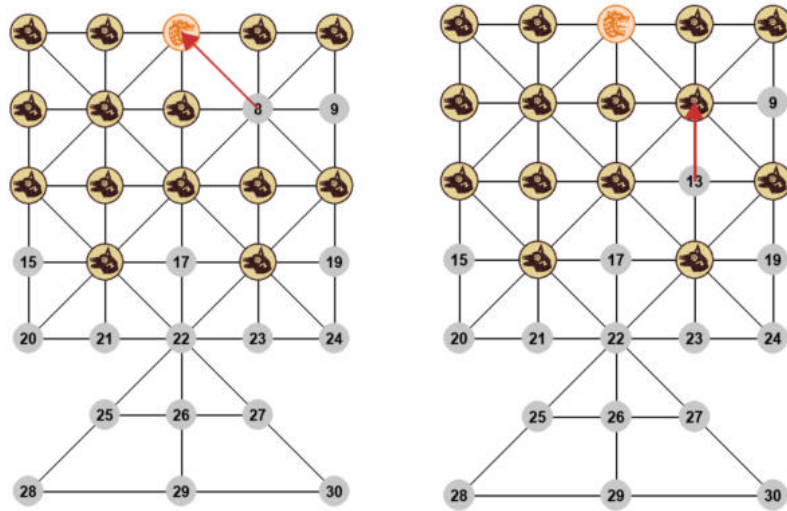
Figura 34 – Turno 15 (Onça) e Turno 19 (Onça)



No turno 15 (Figura 34), podemos ver que rapidamente a onça se encontra já bem restringida de movimentos pelos cachorros. No turno 19 (Figura 34), ela teria a possibilidade de escapar do cerco capturando o cachorro na casa 14, mas não realiza esse movimento, o que acaba levando a derrota nos turnos seguintes, se encerrando no turno 22 (Figura 35).

Nessa partida, podemos ver que a função $f_{c,2}$ (Função 2.3) conseguiu se manter em bloco e sendo beneficiada pela movimentação da onça na hora de realizar o cerco. Lem-

Figura 35 – Turno 21 (Onça) e Turno 22 (Cachorros)

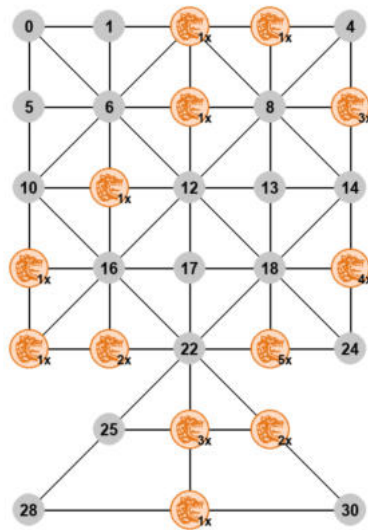


brando que, dessa vez, ela tinha um adversário com comportamento aleatório.

4.4 EXPERIMENTO 8

No Experimento 8, foram 26 vitórias dos cachorros ao todo. Além disso, as casas em que a onça foi encurralada, foram as casas 2, 3, 7, 9, 11, 15, 19, 20, 21, 23, 26, 27 e 29, sendo encurralada cinco vezes na casa 23, quatro vezes na casa 19, três vezes na casa 9, três vezes na casa 26, duas vezes na casa 21 e duas vezes na casa 27, nas demais casas ela foi encurralada apenas uma vez, como mostra a Figura 36. Dentre as casas citadas, a casa 2 possui cinco arestas; as casas 7, 11 e 26 possuem quatro arestas; e as restantes possuem três arestas.

Figura 36 – Distribuição das posições em que a onça foi derrotada no Experimento 8



Para o Experimento 8, vamos observar mais detalhadamente um dos jogos em que a onça terminou presa na casa 26, que pertence a toca da onça, que possui 4 arestas. Além disso, as funções de utilidade que guiam os movimentos dos agentes são, a função $f_{o,1}$ (Função 2.1) e função $f_{c,2}$ (Função 2.3).

Figura 37 – Turno 5 (Onça) e Turno 17 (Onça)

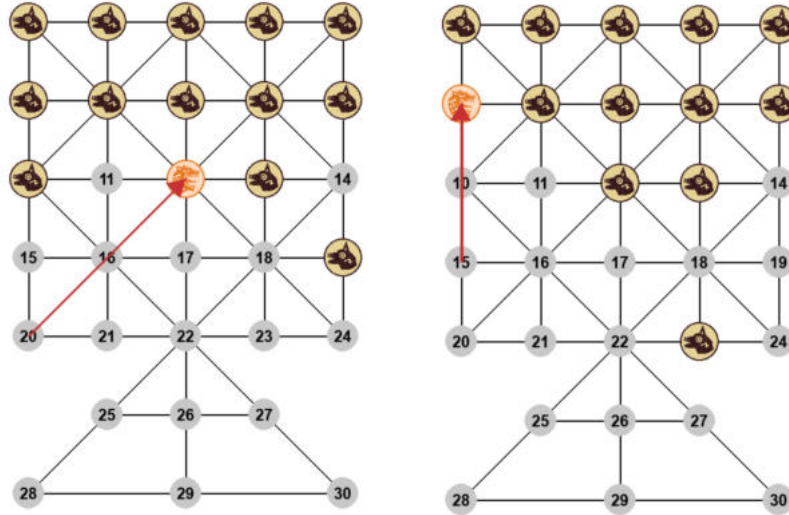
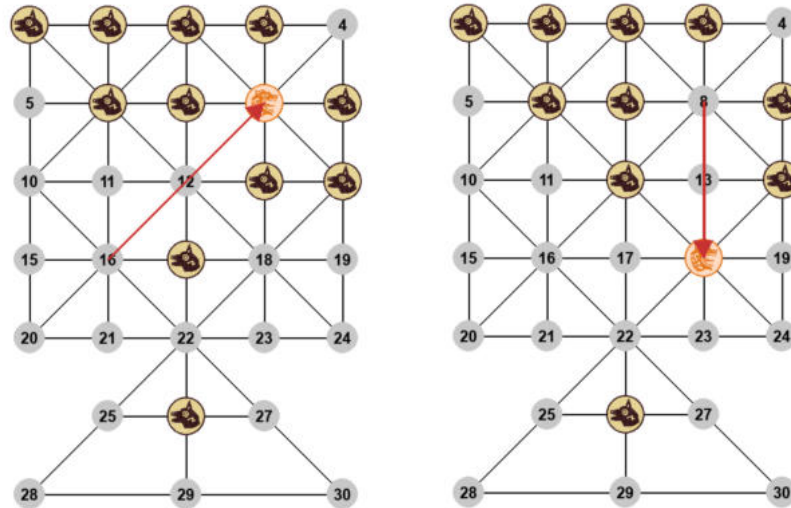
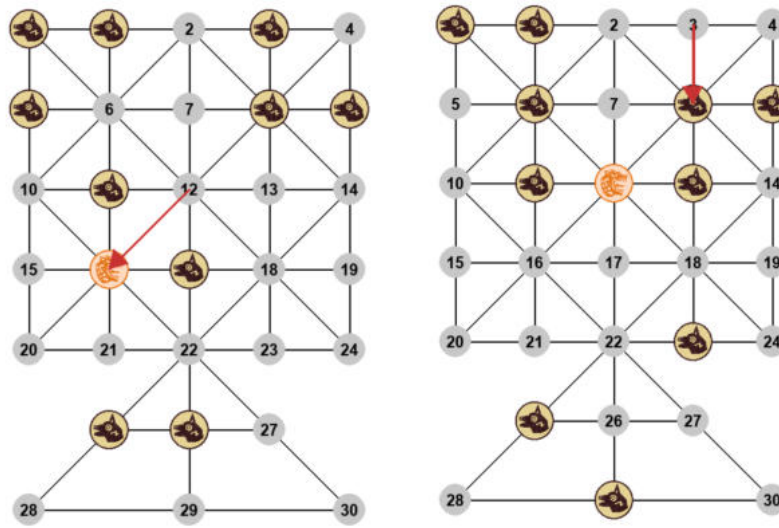


Figura 38 – Turno 35 (Onça) e Turno 37 (Onça)



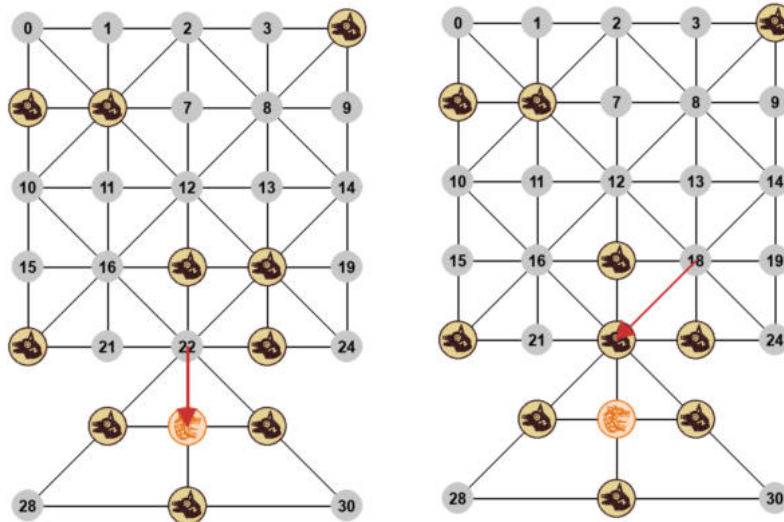
Logo de início, podemos ver que a onça realiza rapidamente quatro capturas até o turno 37 da partida, como pode ser observado nas Figuras 37 e 38. Após rapidamente se colocar numa posição favorável contra os cachorros, a onça apresenta um comportamento em que evita capturas, como mostra a Figura 39, onde ela tem três possibilidades de movimento que lhe concederiam a vitória naquele momento, mas que ela opta por não realizar naquele momento, postergando a partida. Situação essa que volta a acontecer no turno 144 (Figura 39), onde existem novamente três possibilidades de captura para encerrar a partida com vitória da onça, que ela não realiza.

Figura 39 – Turno 105 (Onça) e Turno 144 (Onça)



Esse comportamento de evitar capturas se repete ao longo do restante da partida que só vai se encerrar no turno 776 (Figura 40), com a vitória dos cachorros. Dessa forma, a onça só entra na toca da onça justamente no fim da partida no turno 775 (Figura 40). Isso acontece devido ao aumento da profundidade para a onça, fazendo com que ela tenha um comportamento de adiar a última captura, o que causa um alongamento na duração da partida em relação aos testes anteriores. Isso pode ser observado também pelo aumento geral na média de turnos no Experimento 8.

Figura 40 – Turno 775 (Onça) e Turno 776 (Cachorros)



4.5 EXPERIMENTO 9

No Experimento 9, foram 282 vitórias dos cachorros ao todo, sendo o experimento com o maior número de vitórias dos cachorros. Além disso, as casas em que a onça foi

Figura 42 – Turno 7 (Onça) e Turno 15 (Onça)

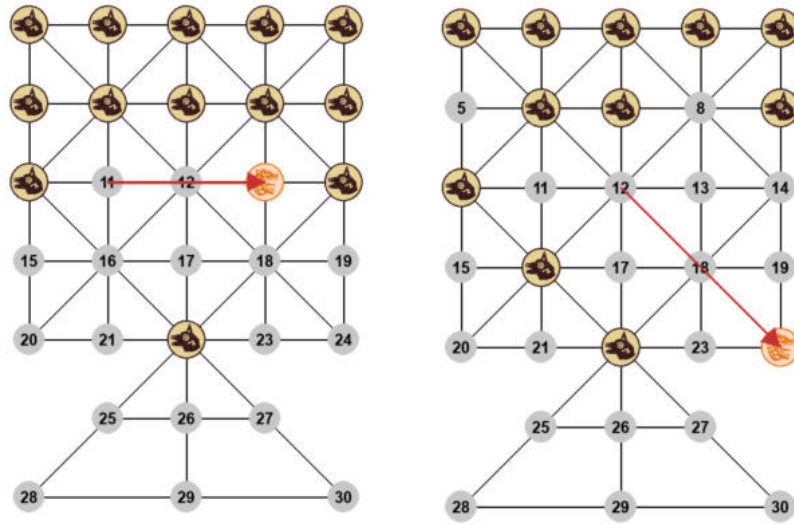
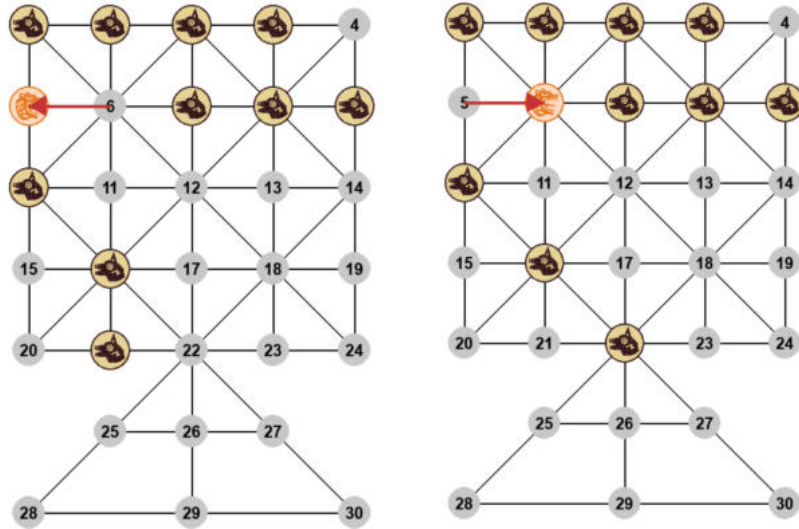


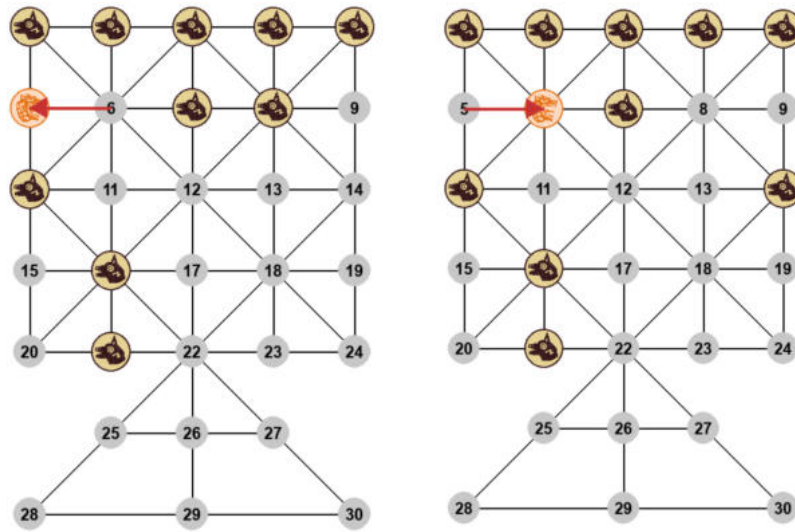
Figura 43 – Turno 21 (Onça) e Turno 23 (Onça)



portamento se mantém até o final da partida (Figura 44). Durante o restante da partida, os cachorros executam movimentos distintos, porém sem efetivamente colocar a onça em uma situação arriscada.

Esse comportamento da onça se repete em outras partidas, sendo provavelmente influenciado pela definição da função $f_{o,2}$ (Função 3.1), que leva em consideração as casas com maior número de movimentos possíveis. Além disso, a baixa profundidade limita a previsão dos estados futuros da onça, restringindo a análise a poucos turnos à frente. Dessa forma, a onça tende a privilegiar configurações localmente vantajosas, ainda que isso resulte na formação de ciclos e na ausência de progressão efetiva da partida.

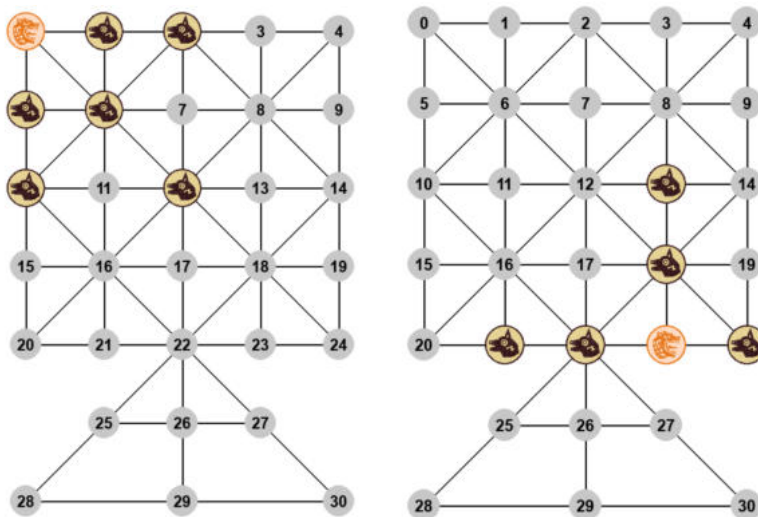
Figura 44 – Turno 997 (Onça) e Turno 999 (Onça)



4.6 VISÃO GERAL

No geral, observando os experimentos, a maioria das casas em que a onça termina o jogo derrotada são em casas que possuem apenas três arestas, sendo as casas com o segundo menor número de arestas, ficando atrás apenas das casas 28 e 30, com duas arestas cada. Suspeitamos que isso ocorra devido ao número de cachorros necessários para prender a onça em uma determinada casa, que varia conforme o número de arestas, também impactando no número de saltos para captura que a onça pode realizar.

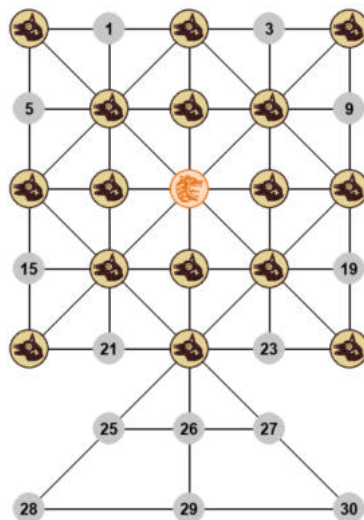
Figura 45 – Configuração do Tabuleiro – Exemplo 3



Por exemplo, no tabuleiro mais à esquerda da Figura 45, a onça está na casa 0 que possui três arestas, sendo necessário seis cachorros para conseguir bloquear seus movimentos, com eles posicionados nas casas 1, 2, 5, 6, 10 e 12. Entretanto, esse número varia nas casas que possuem três arestas, como mostra o tabuleiro mais à direita da Figura 45,

com a onça encurralada na casa 23, sendo necessário 5 cachorros posicionados nas casas 13, 18, 21, 22, e 24. Isso acontece pelo formato do tabuleiro, pois na direção da casa 23 para a casa 24 só existe 1 espaço disponível para ocupar, diminuindo assim o número de cachorros necessários para prender a onça.

Figura 46 – Configuração do Tabuleiro – Exemplo 4



Seguindo esse raciocínio, se onça estiver na casa 12, casa inicial da onça, que possui um total de oito arestas (sendo, junto com as casas 6, 8, 16, 18 e 22, a casa com maior número de arestas), seriam necessários dois cachorros em cada direção para conseguir encurralá-la. Dessa forma, seriam necessários dezesseis cachorros para encurralá-la, como mostra a Figura 46, mas isso não é possível, pois o número máximo de cachorros no jogo é quatorze, tornando impossível para os cachorros prenderem a onça enquanto ela se encontra em uma dessas casas.

Em um panorama da análise qualitativa pudemos observar em diversos experimentos, um comportamento por vezes hesitante da onça em realizar algumas capturas e que esse comportamento por vezes a levava a derrota. Por parte dos cachorros podemos ver que eles se colocam em posições muito ruins, fazendo com que a onça pudesse ter vários movimentos de captura.

Em geral é possível notar que a função $f_{c,2}$ (Função 2.3) se destaca em relação a função $f_{c,1}$ (Função 2.2) sendo responsável por 37% das vitórias analisadas neste capítulo. Enquanto a função $f_{c,1}$ (Função 2.2) foi responsável por 2% e a junção das duas na função $f_{c,3}$ (Função 3.3) foi responsável por 61% das vitórias. Foi possível notar que a função $f_{c,2}$ (Função 2.3) até certo ponto consegue manter os cachorros em bloco, evitando sua captura. Porém, ao tentar uma estratégia mais agressiva para fechar o cerco, acaba se tornando vulnerável, sendo refletido no número baixo de vitórias em relação as funções

da onça.

Também foi constatado alguns casos em que ocorre um número elevado de turnos e que a onça capturou rapidamente 4 de 5 cachorros, mas evita finalizar o jogo. Acreditamos que esse comportamento seja fruto do aumento da profundidade e da paridade da profundidade, profundidades ímpares aparentam ter um comportamento melhor com as funções da onça, enquanto profundidades pares aparentam ter um comportamento melhor com as funções dos cachorros. Acreditamos que esse comportamento em relação as paridades tenha a ver com o caráter assimétrico dos agentes, com objetivos distintos dentro do jogo.

Figura 47 – Graus dos vértices

```
Grau 2: [28, 30]
Grau 3: [0, 1, 3, 4, 5, 9, 15, 19, 20, 21, 23, 24, 25, 27, 29]
Grau 4: [7, 11, 13, 17, 26]
Grau 5: [2, 10, 14]
Grau 8: [6, 8, 12, 16, 18, 22]
```

Além disso, é importante ressaltar o impacto do grau do vértice nas casas em que a onça termina capturada. Ao todo foram 356 vitórias dos cachorros analisadas, das quais 53% destas vitórias foram em casas com três arestas, 33% em casas com quatro arestas, 12% em casas com 2 arestas e somente 2% em casas com cinco arestas. Na Figura 47, estão representados o grau de cada vértice agrupado pelo valor dos graus existentes no grafo(tabuleiro).

5 CONCLUSÃO

Este trabalho teve como propósito realizar uma análise exploratória sobre a aplicação do algoritmo Minimax no jogo da onça, um jogo de origem indígena brasileiro, analisando funções de utilidade buscando um equilíbrio no desempenho entre os agentes. Foram propostas diferentes funções de utilidade, além de maneiras de se utilizar o Minimax no jogo da onça, variando parâmetros como a profundidade da árvore de busca do jogo. Ao longo do texto foram detalhados a implementação do jogo e os experimentos desenvolvidos assim como os resultados obtidos.

A hipótese inicial deste trabalho, considera que há um equilíbrio entre os agentes, de modo que, apesar de possuírem objetivos distintos, não exista uma vantagem consistente para nenhum dos lados. Contudo, os resultados obtidos ao longo do nosso trabalho indicam que a onça leva uma vantagem em relação aos cachorros, considerando-se as funções aqui analisadas. Enquanto por parte dos cachorros, apesar de também apresentarem uma melhoria pequena para algumas das funções propostas, essa não se faz satisfatória para equilibrar as disputas contra a onça, possivelmente devido a complexidade em se traduzir estratégias coletivas para os cachorros que fossem ao mesmo tempo seguras, evitando capturas, e agressivas, ao ponto de encurralar a onça.

Os resultados obtidos, mostram que a função $f_{o,2}$ (Função 3.1), se mostrou ligeiramente superior a função $f_{o,1}$ (Função 2.1). Melhorando o desempenho da onça contra os cachorros. Enquanto do ponto de vista dos cachorros, a função $f_{c,2}$ (Função 2.3), se destaca entre as demais, apresentando os melhores resultados, mesmo que estes não sejam os resultados esperados. Estes resultados evidenciam a dificuldade de se conceber estratégias realmente eficazes para os cachorros, devido ao caráter complexo do controle coletivo de peças e demonstram melhorias no desempenho da onça.

Acreditamos que a forma de combinar diferentes funções contribuiu positivamente para formar uma estratégia para os cachorros, buscando representar um comportamento coletivo e de dividir o objetivo dos cachorros em dois, uma estratégia voltada para cercar a onça e outra voltada para se defender de capturas. Julgamos que essa contribuição possa auxiliar na descoberta e implementações de outras funções que possam vir a ser mais adequadas para representar bem o papel dos cachorros. Notamos que uma das dificuldades presentes neste trabalho é o caráter assimétrico dos agentes, que causa um impacto nas profundidades utilizadas, que dependendo da paridade da profundidade, tornava o jogo mais benéfico para um dos agentes, assim como no desinteresse da onça em realizar a captura final em algumas partidas observadas durante a análise qualitativa.

Consideramos, que uma investigação mais aprofundada a respeito do grau do vértice da casa em que a onça é cercada pelos cachorros, temos fortes indícios de que essa seja uma das estratégias a se seguir na hora de definir novas estratégias ou aprimorar as estratégias

presentes neste trabalho.

Outra possibilidade de trabalho futuro seria tentar uma abordagem diferente, utilizando o aprendizado por reforço, que não foi abordado neste trabalho por considerarmos mais apropriado compreender as dificuldades ao utilizar uma abordagem mais tradicional, com os agentes desenvolvendo suas estratégias por meio de tentativa e erro, explorando diferentes estratégias e recebendo recompensas com base nos resultados obtidos.

REFERÊNCIAS

CAMPBELL, M.; JR, A. J. H.; HSU, F.-h. Deep blue. **Artificial intelligence**, Elsevier, v. 134, n. 1-2, p. 57–83, 2002.

COMMUNITY, P. **Pygame**. 2026. Disponível em: <https://www.pygame.org>. Acesso em: 27 jan. 2026.

FARIA, R. E. O. de; ARAÚJO, R. B.; BAUMANN, A. P. P. Jogo da onça: Um relato de experiência em aulas de matemática para turma de eaja. **Encontro Goiano de Educação Matemática**, v. 6, n. 6, p. 545–555, 2017.

FROZZA, R.; SORIO, F.; ALVARES, L. O. C. Simula-ambiente de simulação em sistemas multiagentes reativos. In: **IV Congreso Argentina de Ciencias de la Computación**. [S.l.: s.n.], 1998.

HSU, F.-H. **Behind Deep Blue: Building the computer that defeated the world chess champion**. [S.l.]: Princeton University Press, 2002.

NASCIMENTO, F. do; GUEDES, E. Análise comparativa de funções de utilidade para o algoritmo minimax com poda alfa-beta aplicadas ao jogo da onça. In: SBC. **Anais do XVI Encontro Nacional de Inteligência Artificial e Computacional**. [S.l.], 2019. p. 49–59.

RUSSELL, S. J.; NORVIG, P. **Artificial intelligence: a modern approach**. [S.l.]: Pearson, 2016.

SARDINHA, A. G. de O.; GASPAR, M. T. J. Jogos indígenas aplicados ao ensino de matemática. **Anais do X Encontro Nacional de Educação Matemática**, p. 1–10, 2010.

SILVER, D. et al. Mastering the game of go with deep neural networks and tree search. **nature**, Nature Publishing Group, v. 529, n. 7587, p. 484–489, 2016.

SILVER, D. et al. Mastering the game of go without human knowledge. **nature**, Nature Publishing Group UK London, v. 550, n. 7676, p. 354–359, 2017.

APÊNDICE A – HISTOGRAMAS UTILIZADOS NO EXPERIMENTO 3

Figura 48 – Histograma - Onça ($f_{o,0}$) x Cachorros ($f_{c,1}$) - Profundidade 5

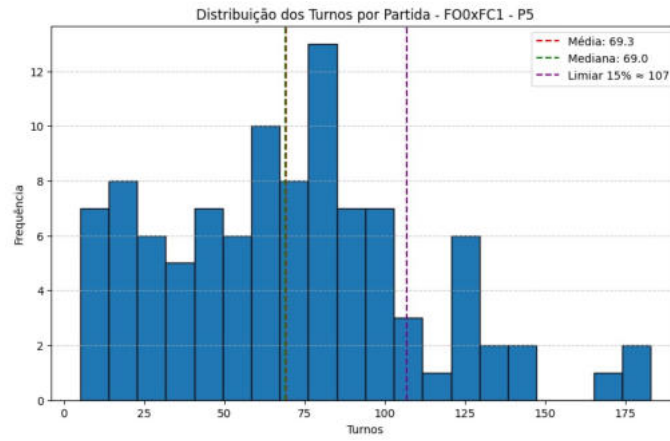


Figura 49 – Histograma - Onça ($f_{o,0}$) x Cachorros ($f_{c,2}$) - Profundidade 5

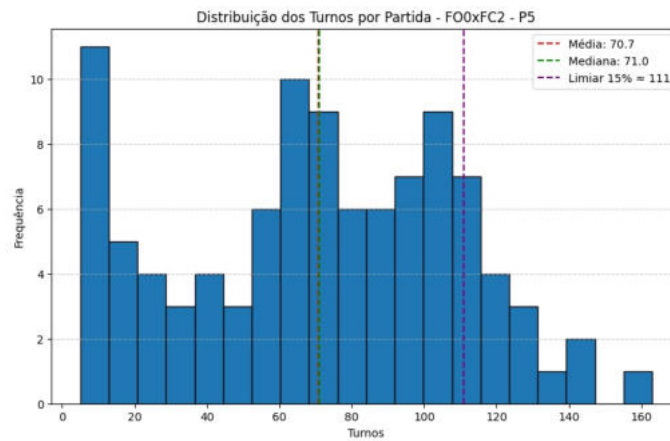


Figura 50 – Histograma - Onça ($f_{o,1}$) x Cachorros ($f_{o,0}$) - Profundidade 5

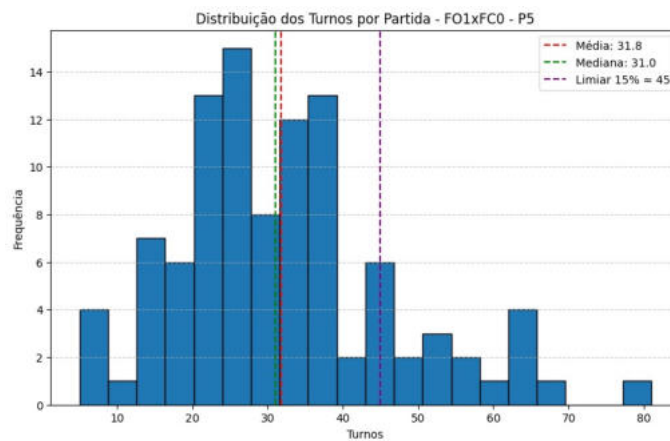


Figura 51 – Histograma - Onça ($f_{o,1}$) x Cachorros ($f_{c,2}$) - Profundidade 5

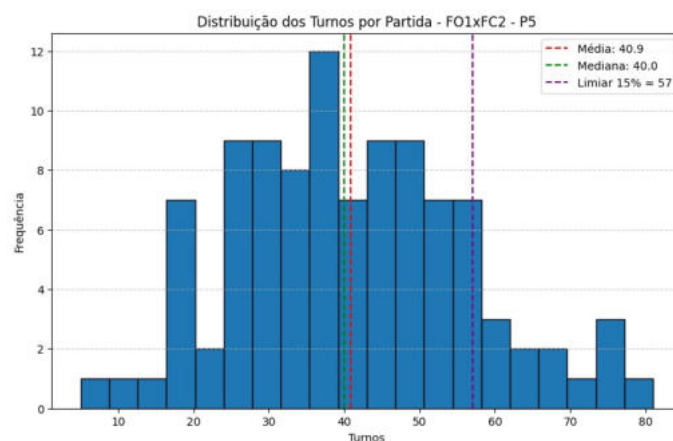


Figura 52 – Histograma - Onça ($f_{o,2}$) x Cachorros ($f_{c,0}$) - Profundidade 5

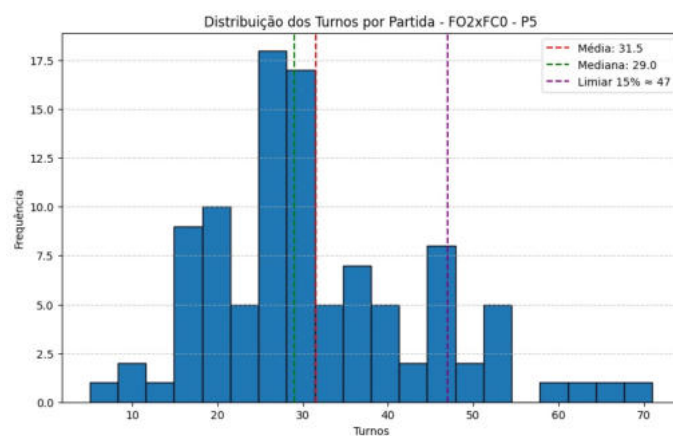


Figura 53 – Histograma - Onça ($f_{o,2}$) x Cachorros ($f_{c,1}$) - Profundidade 5

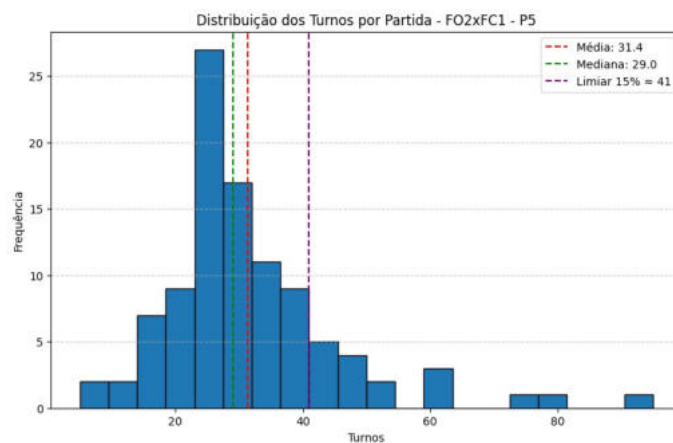


Figura 54 – Histograma - Onça ($f_{o,2}$) x Cachorros ($f_{c,2}$) - Profundidade 5

