



INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DO CEARÁ
IFCE *CAMPUS* MARACANAÚ
BACHAREL EM CIENCIA DA COMPUTAÇÃO

FRANCISCO YURI CARVALHO DE OLIVEIRA
JOÃO VICTOR DE FRANÇA LEITÃO
JOÃO VITOR MOREIRA DUARTE

INTELIGÊNCIA ARTIFICIAL: USO DE INTELIGÊNCIA ARTIFICIAL NO JOGO DE
XADREZ PARA COMPUTADOR

MARACANAÚ - CE

2022

FRANCISCO YURI CARVALHO DE OLIVEIRA

JOÃO VICTOR DE FRANÇA LEITÃO

JOÃO VITOR MOREIRA DUARTE

INTELIGÊNCIA ARTIFICIAL: USO DE INTELIGÊNCIA ARTIFICIAL NO JOGO DE
XADREZ PARA COMPUTADOR

Pré-projeto de pesquisa apresentado na cadeira de metodologia científica no curso de Bacharel em Ciencia da Computação do Instituto Federal de Educação, Ciência e Tecnologia do Ceará (IFCE) - *Campus* Maracanaú.

Linha de Pesquisa: Inteligencia Artificial e suas aplicações

MARACANAÚ - CE

2022

SUMÁRIO

| | | |
|--------------|---|-----------|
| 1 | INTRODUÇÃO | 2 |
| 1.1 | Tema | 2 |
| 1.2 | Delimitação do Tema | 2 |
| 1.3 | Problema | 2 |
| 1.4 | Objetivos | 2 |
| 1.4.1 | <i>Objetivo Geral</i> | 2 |
| 1.4.2 | <i>Objetivos Específicos</i> | 2 |
| 2 | JUSTIFICATIVA | 3 |
| 2.1 | Relevância | 3 |
| 3 | REFERENCIAL TEÓRICO | 5 |
| 3.1 | Teoria e Conceitos de Base | 5 |
| 3.1.1 | <i>Conceito de Inteligencia Artificial</i> | 5 |
| 3.1.2 | <i>Conceito do motor de xadrez</i> | 6 |
| 3.1.3 | <i>Algoritmo de busca min-max</i> | 7 |
| 3.1.4 | <i>Algoritmo alpha-beta pruning</i> | 8 |
| 3.1.5 | <i>Funções com uso de redes neurais</i> | 9 |
| 3.2 | Estado da Arte | 10 |
| 4 | METODOLOGIA | 12 |
| 5 | CRONOGRAMA | 13 |
| | REFERÊNCIAS | 14 |

1 INTRODUÇÃO

1.1 Tema: Inteligência Artificial

1.2 Delimitação do Tema: Uso de inteligência artificial no jogo de xadrez para computador.

1.3 Problema

O problema que norteará esta pesquisa está ligado à investigação da atuação dos melhores algoritmos de Inteligência artificial nos motores de jogos de xadrez para computador, visando procurar as melhores jogadas dentro do jogo a partir da representação matemática do tabuleiro e suas peças.

1.4 Objetivos

1.4.1 *Objetivo Geral*

Investigar e comparar os melhores algoritmos de inteligência artificial nos motores de jogos de xadrez para computador, classificando-os com diversos parâmetros, como velocidade, qualidade de movimento e porcentagem de vitórias, assim revelando no que cada motor pode dedicar-se para sua melhoria.

1.4.2 *Objetivos Específicos*

- Analisar a representação do tabuleiro e o valor das jogadas possíveis dentro do jogo
- Analisar as diferenças entre as implementações dos algoritmos de busca *min-max*, *alpha-beta pruning* e redes neurais nos motores de jogos de xadrez para computador.
- A partir dos dados coletados comparar os algoritmos e os motores de jogos de xadrez para computador.
- Especificar em que áreas ou algoritmos cada motor de xadrez pode melhorar.

2 JUSTIFICATIVA

A Inteligência Artificial (IA), foi escolhida como tema deste trabalho por sua grande importância atual e futura no desenvolvimento da humanidade, tendo aplicações claras em diversas áreas, como carros autônomos, assistentes digitais, algoritmos de reconhecimento de imagens, assim como qualquer projeto ou área que envolva a análise de uma grande base de dados.

Optamos por analisar um tópico simplificado dentro do tema de Inteligência Artificial, que é o uso da mesma em algoritmos utilizados em motores do jogo de xadrez para computador, com o intuito de investigar e comparar os algoritmos e motores para classificá-los, revelando quais os melhores entre eles em diferentes quesitos e o porquê de assim serem, deste modo apresentaremos em quais áreas cada motor e implementação de algoritmos podem empenhar-se para seu aperfeiçoamento.

O xadrez é um jogo entre dois oponentes, um de cada lado de um tabuleiro de 64 casas de cores alternadas, cada jogador tem 16 peças. O objetivo é fazer um xeque mate, que acontece quando o rei está posicionado de forma que é possível fazer sua captura e não é possível escapar.

A importância dos jogos de tabuleiro no tema é exposta por Luger (2013),

(...) os jogos de tabuleiro tem certas propriedades que os tornaram objetos de estudo ideais para esses trabalhos iniciais. A maioria dos jogos utiliza um conjunto bem definido de regras: isso faz com que seja fácil gerar o espaço de busca e libera o pesquisador de muitas das ambiguidades e complexidades inerentes a problemas menos estruturados. As configurações do tabuleiro usadas nesses jogos são facilmente representáveis em um computador, dispensando o formalismo complexo necessário para capturar as sutilezas semânticas de domínios de problemas mais complexos. (LUGER, 2013, p.17)

2.1 Relevância

A inteligência artificial pode ser definida como sistemas ou máquinas que procuram imitar o raciocínio humano visando realizar alguma atividade podendo aprimorar seu desempenho de forma interativa com base nas informações que coletam.

Com o constante aumento do seu uso na resolução de problemas dentro da sociedade, indo da construção de trajetos mais eficientes para os serviços que proporcionam mapas até traçar um perfil de consumidor a partir da navegação de um usuário dentro da internet, a inteligência artificial e seus algoritmos são grandes responsáveis em como o mundo funciona atualmente.

A implementação desses programas para a análise de dados tradicionais permite descrever problemas em que ações que já aconteceram identificam novas oportunidades e implementam

estratégias baseadas nos dados para chegar em um resultado satisfatório.

Utilizando como referencial o jogo de xadrez para computador como exemplo, é possível exemplificar os algoritmos que fazem a base da inteligência artificial e mostrar como problemas podem ser representados de forma que uma máquina chegue a soluções seguindo um raciocínio matemático comparável a logica humana.

3 REFERENCIAL TEÓRICO

Nesta seção, serão apresentados os fundamentos teóricos que servirão de base para sustentar o estudo sobre a inteligência artificial dentro da área de algoritmos e motores do jogo de xadrez para computador. Primeiramente, iremos compreender as teorias e conceitos de base para esta pesquisa, que proveram as informações necessárias para a análise e comparação em questão. Depois, falaremos sobre o estado da arte deste tema.

3.1 Teoria e Conceitos de Base

Esta parte do projeto conterá as informações teóricas necessárias para compreensão do tema e sua problematização, assim veremos os conceitos de inteligência artificial, motor de xadrez, algoritmo de busca *min-max*, algoritmo *alpha-beta pruning* e as funções com uso de redes neurais.

3.1.1 Conceito de Inteligencia Artificial

O conceito de inteligência artificial surgiu da ideia de reproduzir nas máquinas a capacidade humana de usar das informações disponíveis para resolver problemas e tomar decisões com base na razão e lógica, o que resultou em dar aos computadores a capacidade de automatizar processos ou pelo menos minimizar consideravelmente o envolvimento humano nós mesmos, e com o avanço cada vez maior da velocidade de processamento das máquinas, elas alcançaram a capacidade de analisar dados em taxas extremamente mais rápidas do que a humana.

Hintze (2016) nos explica sobre os tipos mais básicos da inteligência artificial,

Os tipos mais básicos de sistemas de IA são puramente reativos e não têm a capacidade de formar memórias nem de usar experiências passadas para informar as decisões atuais. Deep Blue, o supercomputador de xadrez da IBM, que derrotou o grande mestre internacional Garry Kasparov no final dos anos 1990, é o exemplo perfeito desse tipo de máquina. (HINTZE, 2016, tradução nossa.)

Também é dito por Jajal (2018) que:

A Inteligência Artificial Estreita (AIE) também conhecida como IA “fraca” é a IA que existe em nosso mundo hoje. AIE é a IA programada para realizar uma única tarefa – seja verificar o clima, jogar xadrez ou analisar dados brutos para escrever relatórios jornalísticos.(JAJAL, 2018, tradução nossa.)

É importante diferenciar o tipo mais básico de inteligência artificial dos mais complexos, pois o pensamento mais comum quando falamos neste tema é a criação de máquinas semelhantes

aos humanos que assim como nós possam pensar e agir por conta própria, possuindo a capacidade de aprender e até mesmo possuir sentimentos e consciência, mas tais feitos só podem ser alcançados utilizando-se de tecnologias de áreas como machine learning e redes neurais, que são ramos da inteligência artificial.

De acordo com Allende-Cid(2019),

Machine Learning é a área ideal para a automatização de processos, os quais podem ser "simples", como reconhecer padrões visuais, ou complexos, tais quais decisões de especialistas da área da saúde. Quando seres humanos lidam com problemas complexos, muitas vezes é impossível explicar o raciocínio que levou a tomar determinadas decisões. Por outro lado, é menos complexo realizarmos a coleta dos exemplos de decisões tomadas por seres humanos e usá-los como fonte para que o sistema aprenda a resolver o mesmo problema.(ALLENDE-CID, 2019, tradução dos editores, p.16.)

3.1.2 Conceito do motor de xadrez

A primeira aplicação de um motor de xadrez foi criada entre os anos de 1950 a 1953 por Alan Turing juntamente com a ideia base de Claude Shannon, criando assim o primeiro algoritmo para o jogo de xadrez de computador, que pela falta de uma máquina adequada teve que ter cada movimento calculado manualmente por Turing via o algoritmo.

Santana (2014) define um motor de xadrez como:

(...) um programa de computador capaz de decidir um movimento em uma partida de xadrez. Tal programa recebe uma configuração de um tabuleiro, isto é, o conjunto de casas contendo a informação de qual peça está ocupando cada casa, analisa esta informação considerando somente os movimentos válidos e retorna um movimento que é o melhor possível de acordo com algum critério.(SANTANA, 2014, p.4.)

Os motores de xadrez em geral não possuem interface gráfica própria, eles apenas recebem comandos e devolvem o próximo movimento, assim para que haja interação com outros programas se faz necessário um protocolo de comunicação, a partir da padronização de comandos utilizados no protocolo, dois motores diferentes podem interagir com uma mesma interface sem conflitos.

É necessário que o motor faça a representação do tabuleiro do jogo, mantendo dados como a posição de todas as peças no tabuleiro, a regra de 50 movimentos, entre outros. Uma boa representação do tabuleiro torna o cálculo de movimentos e a avaliação do tabuleiro muito rápidas, de forma geral, existem três métodos diferentes para representar um tabuleiro: centrado nas casas, centrado nas peças e soluções híbridas.

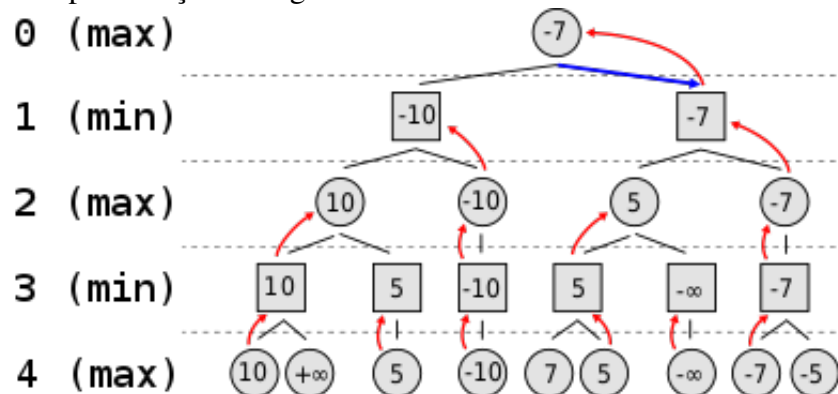
Bijil e Tiet (2021) ressaltam a importância desta fase, pois qualquer algoritmo, esteja relacionado a pesquisa ou avaliação, tem como base a implementação da representação do tabuleiro.

3.1.3 Algoritmo de busca min-max

O minimax é um algoritmo de força bruta, isso significa que seu objetivo é enumerar todos os possíveis candidatos de uma solução e verificar se cada um satisfaz o problema, no caso do minimax ele divide as possibilidades de ações de cada um dos jogadores em uma árvore de jogadas para conseguir a melhor jogada possível.

Essa árvore vai ser definida em etapas de minimização(min) e maximização(max), sendo cada uma destas etapas representadas por uma jogada do adversário ou da máquina respectivamente, o objetivo das jogadas de minimização é minimizar as chances de uma boa jogada da máquina e as de maximização é maximizar as chances de uma boa jogada da máquina ocorrer, cada nó representa uma configuração do jogo, e cada aresta de um nó representa uma jogada que leva a uma determinada configuração.

Figura 1 – Representação do algoritmo minimax



Fonte: <<https://wblog.wiki/pt/Minimax>>. Acessado em: 24 de jun. 2022

De acordo com THÉ(1992):

Jogos de soma-zero com dois jogadores que exigem inteligência humana como Xadrez, damas, go e othello podem ser todos resolvidos computacionalmente programando uma procurando uma busca computacional de árvore de estado-espaco onde os nós representam o as posições do problema ou os estados (o tabuleiro do jogo) e os ramos representam as operações que transformam um estado em outro.(THÉ, 1992, tradução nossa.)

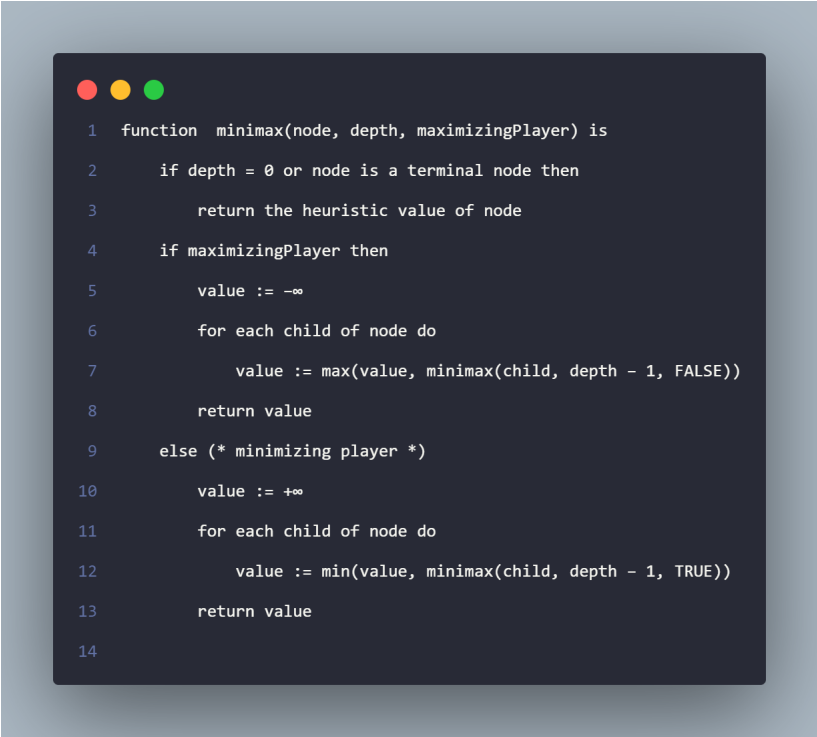
Nós que não possuem arestas são denominados de nó folha, esses nós são configurações de fim de jogo, são aplicados nesses nós um fator avaliativo para validar se aquele nó possui um

resultado positivo ou negativo para a máquina, o nó em questão recebe um valor com base no seu resultado.

No caso do xadrez o valor avaliativo é dado com base em um valor específico para cada peça, caso haja mais peças da máquina com um valor significativo aquele possui um valor positivo, caso contrário possui um valor negativo.

Após a avaliação todos as arestas de um nó atribuímos um valor ao mesmo, se o nó em questão for um nó de min, ou seja, uma jogada do adversário, atribuímos ao nó o valor mínimo entre os valores de suas arestas, caso seja o max, uma jogada da máquina, atribuímos o valor máximo entre suas arestas.

Figura 2 – Pseudo Código do *minimax*



```

1  function minimax(node, depth, maximizingPlayer) is
2      if depth = 0 or node is a terminal node then
3          return the heuristic value of node
4      if maximizingPlayer then
5          value := -∞
6          for each child of node do
7              value := max(value, minimax(child, depth - 1, FALSE))
8          return value
9      else (* minimizing player *)
10         value := +∞
11         for each child of node do
12             value := min(value, minimax(child, depth - 1, TRUE))
13         return value
14

```

Fonte: De autoria própria.

No fim de todas as arestas o algoritmo escolhe a aresta com o maior valor pois esta é a melhor alternativa para se seguir.

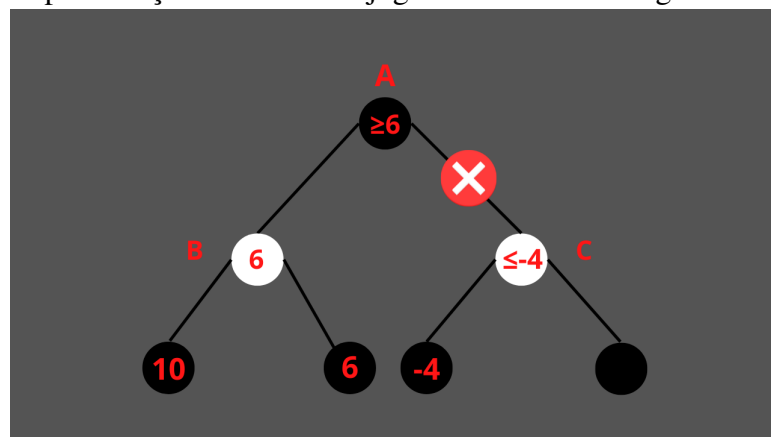
3.1.4 Algoritmo *alpha-beta pruning*

O *alpha-beta pruning* é um algoritmo de otimização do minimax onde excluimos as arestas que possuem um valor pior do que um valor encontrado anteriormente, isso é feito para economizar tempo de processamento da máquina já que o algoritmo minimax por si só é muito custoso.

O algoritmo funciona da seguinte forma, existem duas variáveis chamadas *alpha* e *beta* respectivamente, *alpha* é o maior valor que o maximizador pode garantir naquele nível da árvore ou abaixo e *beta* é o menor valor que o minimizador possui naquele nível da árvore ou abaixo, a cada subida na árvore esses valores são atualizados com base nos maiores valores encontrados, logo, quando o algoritmo está decidindo o valor que irá atribuir ao nó em questão ele verifica as variáveis para verificar se uma das arestas tem chances de possuir um valor maior ou menor do que ele já possui nas variáveis *alpha* e *beta*.

Pegando por exemplo um nó maximizador A com dois nós filhos minimizadores, B e C, caso o valor de B seja 6 saberemos que o nó A maximizador escolherá um valor maior ou igual a 6, se avaliarmos o nó C e ele possuir um nó filho com valor 4, saberemos que aquele nó só escolherá um valor 4 ou menor, logo não há necessidade de fazer mais nenhuma avaliação de seus filhos pois o maior valor para o nó A já foi escolhido.

Figura 3 – Representação da árvore de jogadas utilizando o algoritmo alpha-beta



Fonte: De autoria própria.

3.1.5 Funções com uso de redes neurais

Redes neurais ou redes neurais artificiais é um dos tipos de *machine learning* que se inspira no cérebro humano imitando a forma como os nossos neurônios conversam entre si usando um algoritmo de aprendizagem que tem como objetivo modificar pesos sinápticos da rede de uma forma ordenada para alcançar uma solução esperada.

Haykin(2007) define redes neurais e suas semelhanças com o cérebro humano da seguinte forma:

Uma rede neural é um processador maciçamente paralelamente distribuído constituído de unidades de processamento simples, que têm a propensão natural para armazenar conhecimento experimental e torna-lo disponível para uso. Ela se assemelha ao cérebro humano em dois aspectos:

Figura 4 – Pseudo Código do *alpha-beta*

```

1  function alphabeta(node, depth,  $\alpha$ ,  $\beta$ , maximizingPlayer) is
2      if depth = 0 or node is a terminal node then
3          return the heuristic value of node
4      if maximizingPlayer then
5          value :=  $-\infty$ 
6          for each child of node do
7              value := max(value, alphabeta(child, depth - 1,  $\alpha$ ,  $\beta$ , FALSE))
8              if value  $\geq \beta$  then
9                  break (*  $\beta$  cutoff *)
10              $\alpha$  := max( $\alpha$ , value)
11         return value
12     else
13         value :=  $+\infty$ 
14         for each child of node do
15             value := min(value, alphabeta(child, depth - 1,  $\alpha$ ,  $\beta$ , TRUE))
16             if value  $\leq \alpha$  then
17                 break (*  $\alpha$  cutoff *)
18              $\beta$  := min( $\beta$ , value)
19         return value

```

Fonte: De autoria própria.

1. O conhecimento é adquirido pela rede a partir de seu ambiente através de um processo de aprendizagem.
2. Forças de conexão de neurônios, conhecidas como pesos sinápticos, são utilizados para armazenar o conhecimento adquirido. (HAYKIN, 2007, p.28, tradução por Paulo Martins Engel)

Fausett(1993) complementa dizendo que:

(...) um sistema de processamento de informações que possui certas características de desempenho em comum com as redes neurais biológicas. Redes neurais artificiais foram desenvolvidas como generalização de modelos matemáticos da cognição humana ou biologia neural (...) (FAUSETT, 1993, p.3, tradução nossa.)

3.2 Estado da Arte

Existem vários trabalhos envolvendo a criação de motores de xadrez e descrição de seus algoritmos, iremos aqui citar alguns dos quais tomamos como base para este projeto.

Santana (2014) apresenta teorias e algoritmos relacionados ao xadrez para computador e mostra possíveis implementações para um motor de xadrez, no final ele analisa a implementação

dentro do motor de xadrez de código aberto Pulse.

Abreu *et al.* (2006) também apresenta teorias e implementações de algoritmos, mas este descreve técnicas mais avançadas, como: Tabelas de transposição, extensões e reduções de busca, livro de aberturas e banco de dados finais. Além disso, este trabalho faz a análise resumida de um motor de xadrez de propriedade dos autores chamado ICE.

Block *et al.* (2008) apresenta a teoria matemática e lógica para implementar aprendizado por reforço por meio de um método chamado Diferença Temporal (DT), o qual otimiza as funções de avaliação e seus coeficientes, e tem a capacidade de aumentar sua própria compreensão do xadrez após cada partida.

Bijl e Tiet (2021) nos expõem uma visão mais atual quanto ao uso de alguns algoritmos em motores de xadrez. Eles fazem a comparação de diferentes técnicas tanto no fator de velocidade quanto no de qualidade de movimentos.

Como pode ser visto, existem vários trabalhos explicando diferentes tipos de implementação dos algoritmos para motores de xadrez, mas não muitos fazem a comparação entre eles e suas implementações nestes, assim buscamos ampliar a visão neste tema, mostrando caminhos no qual os algoritmos podem ser melhorados.

4 METODOLOGIA

Com base em Marconi e Lakatos (2001), como estaremos utilizando métodos matemáticos, lógicos e estatísticos para esta pesquisa, determinamos que esta terá caráter quantitativo.

Para esta pesquisa, estaremos utilizando os motores de xadrez chamados *Stockfish*, *Leela Chess Zero* (LCZ), *RubiChess*, *Nemorino*, *Igel*, *Xifos*, *Laser*, *Defen chess*, *Andscacs*, *Halogênio*, *Arasan* e o *Combusken*, todos de código aberto.

A comparação que faremos entre os motores vai ser pela sua velocidade em calcular as melhores jogadas, a qualidade de movimentos, sua porcentagem de vitórias e por seu ranking nos campeonatos de motores online, sendo o mais famoso deles o *Top Chess Engine Championship*.

Primeiro começaremos avaliando a implementação da representação do tabuleiro de cada um dos motores de xadrez selecionados, descrevendo como é a estrutura de dados que guarda: as posições de todas as peças, a informação de qual dos jogadores é a vez, o direito de *roque*, a casa com possibilidades de captura *en passant* e o número de movimentos relacionado a regra de 50 movimentos. Também mediremos quão boa a representação do tabuleiro com base nas três principais formas de representação: centrada nas peças, centrada nas casas e a híbrida.

Depois de analisarmos a base do motor, iremos calcular a velocidade mínima, média e de pior caso de cada um dos algoritmos apontados neste trabalho, levando em conta as variáveis de quantidade de peças em jogo, possíveis movimentos de *roque* e captura *en passant*. Além de a partir das referências já citadas e seus diferentes métodos de cálculo de avaliação de movimentos, verificar quais dos algoritmos produzem os melhores movimentos.

Criaremos uma estrutura de teste para verificar a porcentagem de vitórias de cada motor de xadrez, investigando a participação de cada algoritmo nesta porcentagem. Levaremos em conta o elo ou *ranking* dos motores em campeonatos, seus feitos já realizados, e possíveis erros e *bugs* encontrados.

Por fim, analisaremos quais possíveis mudanças e efeitos diferentes algoritmos de comunicação entre dois motores e uma interface podem produzir dentro dos fatores aos quais iremos comparar os motores.

5 CRONOGRAMA

| | 2023 | | 2024 | |
|---|--------|--------|--------|--------|
| Atividades | 2023.1 | 2023.2 | 2024.1 | 2024.2 |
| Créditos a serem cursados | | | | |
| Levantamento bibliográfico e revisão da literatura | | | | |
| Criação de sistema de testes para avaliar os algoritmos e motores do jogo de xadrez para computador | | | | |
| Análise dos resultados obtidos pelos testes realizados | | | | |
| Redação da dissertação | | | | |
| Revisão e redação final da dissertação | | | | |
| Defesa dissertação | | | | |

REFERÊNCIAS

- ABREU, C. M.; LAZOSKI, I. N.; PINTO, E. R. W. P. E. D. Técnicas computacionais para o jogo de xadrez. **Cadernos do IME**, v. 21, n. 1, 2006.
- ALLENDE-CID, H. Machine learning: Catalisador da ciência. **Revista da Sociedade Brasileira de Computação**, v. 39, n. 1, 2019.
- BIJL, P.; TIET, A. P. **Exploring modern chess engine Architectures**. Monografia (Tese de Bacherelado) — Vrije Universiteit Amsterdam, Amsterdam, 2021.
- BLOCK, M.; BADER, M.; TAPIA, E.; RAMÍREZ, M.; GUNNARSSON, K.; CUEVAS, E.; ZALDIVAR, D.; ROJAS, R. Using reinforcement learnig in chess engines. **CONCIBE**, 2008.
- FAUSETT, V. L. **Fundamentals of Neural Networks**: Architectures, algorithms and applications. 1. ed. Upper Saddle River: Pearson, 1993.
- HAYKIN, S. **Redes Neurais**: Princípios e prática. 2. ed. Porto Alegre: Artmed Editora S.A, 2007.
- HINTZE, A. **The Conversation**: Understanding the four types of AI, from reactive robots to self-aware beings. 2016. <<https://theconversation.com/understanding-the-four-types-of-ai-from-reactive-robots-to-self-aware-beings-67616>>. Acessado em: 21 de jun de 2022.
- JAJAL, D. T. **Distinguishing between Narrow AI, General AI and Super AI**: A simple but necessary guide. 2018. <<https://medium.com/mapping-out-2050/distinguishing-between-narrow-ai-general-ai-and-super-ai-a4bc44172e22>>. Acessado em: 23 de jun de 2022.
- LUGER, G. F. **Inteligência Artificial**. 6. ed. São Paulo: Pearson, 2013.
- MARCONI, M. A.; LAKATOS, E. M. **Fundamentos de metodologia científica**. 6. ed. São Paulo: Atlas, 2001.
- SANTANA, H. V. M. D. **Anatomia de um Motor de Xadrez**. Monografia (Trabalho de Conclusão de Curso) — Instito de Matemática e Estatística Universidade de São Paulo, São Paulo, 2014.
- THÉ, E. **An Analysis of Move Ordering on the Efficiency of Alpha-Beta Search**. Monografia (Tese de Mestrado) — School of Computer Science McGill University, Montreal, 1992.