

# Estudo Comparativa do Algoritmo Genético e Algoritmo de Colônia de Formigas na Solução do Problema do Caixeiro Viajante

Robson de Oliveira Miranda da Silva<sup>1</sup>, Vinicius Ponte Machado<sup>2</sup>

Universidade Federal do Piauí (UFPI)  
Centro de Ciências da Natureza (CCN)  
Curso de Bacharelado em Ciência da Computação  
robson.o.miranda@gmail.com, vinicius@ufpi.edu.br

**Abstract.** *The objective of this work is the comparative study of the genetic algorithm (GA) and the ant colony algorithm (ACA), where both were used to solve the problem of the traveling salesman (TSP), the algorithms used have great potential are many used for solve real-world problem co-mo for examples of optimization and routing. A comparative analysis of de-performance performed in the GA and ACA, to determine what better behaves to solve the TSP, thus shows the limitations of each of them. At the end we show the results made during the study done.*

**Keywords.** *Traveling Salesman Problem, Genetic Algorithm, Ant Colony Algorithm.*

**Resumo.** *O objetivo deste trabalho é o estudo comparativo do algoritmo genético (AG) e do algoritmo de colônia de formigas (ACF), onde ambos foram usados para resolver ao problema do caixeiro viajante (PCV), os algoritmos usados tem grande potencial são muitos utilizados para resolver problema mundo real como por exemplos de otimização e roteamento. Uma análise comparativa de desempenho feita no AG e ACF, para determinar qual melhor se comporta para resolver o PCV, assim mostra suas limitações de cada um deles. Ao final mostra os resultados feitas durante o estudo feito.*

**Palavras-chave.** *Problema do Caixeiro Viajante, Algoritmo Genético, algoritmo de colônia de formigas.*

## 1. Introdução

Durante muitos anos a inteligência artificial (IA), vem estudando varias técnicas de simular a inteligência humana e dar a maquina a capacidade de racionar e tomar decisões por si só, possuindo varias técnicas capazes de chegar a um resultado desejado [Russell 2009], partido de um conjunto de regras e informações logicas disponíveis, com isso ele seria capaz de simular a mete humana.

O jogo de damas será explorado neste trabalho, que consiste em um jogo de tabuleiro 8x8 com 24 peças, onde de um lado são 12 brancas e 12 pretas no outro. O objeti-

vo é capturar ou imobilizar as peças do adversário. O jogador que conseguir capturar todas as peças do inimigo ganha a partida.

Esse trabalho tem como objetivo estudar o uso de técnicas de inteligência artificial aplicado ao jogo, realizando um estudo comparativo através de implementação e testes práticos no jogo GG-Dama.

## **2. Motivação**

Motivação de explorar a área de inteligência artificial, dando ênfase na aplicação de um algoritmo de busca dentro de um jogo de damas. Também será apresentado um jogo de GG-Dama, o qual foi desenvolvido pelo autor, onde um dos jogadores tem o movimento de suas peças controlado pela máquina (sendo gerados e avaliados por meio do algoritmo Minimax), simulando assim a disputa entre um humano (usuário) e máquina.

## **3. Trabalhos relacionados**

### **3.1. CHINOOK – O Campeão Mundial**

O mais famoso agente jogador de damas e atual campeão mundial chama-se CHINOOK [Schaeffer et al. 1992] [Schaeffer et al. 1996]. É produto de esforços de pesquisas em estratégias de jogos iniciados em 1989. O projeto tinha dois objetivos que já foram alcançados: desenvolver, em curto prazo, um programa capaz de bater o campeão mundial humano e resolver, a longo prazo, a teoria do jogo de damas. Seu desempenho consiste de quatro aspectos fundamentais:

1. Busca: Eficiente algoritmo de poda alfa-beta paralelo.
2. Função de avaliação: Função ajustada manualmente capaz de decidir quão boa uma posição é.
3. Bases de dados de finais de jogos: informação perfeita de qual fim de jogo é vitória, derrota ou empate.
4. Opening book: base de dados de diretrizes para início de jogo.

CHINOOK usa um algoritmo paralelo de busca alfa-beta. Em 1992, teve uma profundidade mínima média de 19 jogadas. Embora buscas com tal profundidade pareçam impressionantes, ainda são insuficientes para igualar as habilidades humanas, especialmente em jogos. Extensivo trabalho tem sido feito na intenção de melhorar a qualidade da árvore de busca construída pelo jogador.

Base de dados de finais de jogos são coleções de posições com valores provados pela teoria do jogo. CHINOOK tem acesso a bases de dados completas de 10 ou menos peças no tabuleiro, aproximadamente 39 trilhões de possibilidades. Para cada uma dessas possibilidades, existe uma informação perfeita se o jogo leva a uma derrota, empate ou vitória.

O conhecimento da função de avaliação do CHINOOK é uma combinação linear de aproximadamente duas dúzias de componentes, cada um contendo muitos pesos heurísticos ajustados manualmente. Esforços têm sido feitos na tentativa de automatizar o processo de ajuste dos pesos utilizando equações lineares, redes neurais e algoritmos genéticos. Embora essas abordagens tradicionais pareçam promissoras na literatura, a experiência mostra que elas não podem competir com os resultados do ajuste manual.

Em damas, é fundamental evitar algumas posições nas aberturas dos jogos. Por isso, torna-se necessário a utilização de bases de dados de início de jogos – opening books. A falta de um opening book em damas pode fazer com que, antes do quarto movimento, seja encontrada uma situação de derrota forçada. São armadilhas que necessitam de buscas profundas para serem descobertas e, por outro lado, especialistas humanos sabem como enfrentar. No início de seu desenvolvimento, CHINOOK não possuía bases de dados iniciais mas, obviamente, não poderia competir no topo, por muito tempo, sem um bom opening book.

### **3.2. Samuel – O Primeiro agente jogador**

Samuel [Samuel 1959] foi o primeiro grande e conhecido esforço em desenvolver um algoritmo capaz de jogar damas. Seu método consistiu no uso de uma função de avaliação polinomial abrangendo um subconjunto de características do tabuleiro escolhidas de uma grande lista de possibilidades. Samuel obteve um bom jogador de damas, mas ele próprio identificou o primeiro grande defeito de seu programa [Samuel 1967]: a ausência de um procedimento de máquina, efetivo, gerando novos parâmetros para o procedimento de avaliação.

### **3.3. ANACONDA – Pressionar o oponente sem depender de perícia humana**

Outro bem sucedido programa jogador é o ANACONDA [Chellapilla and Fogel 2000] [Fogel and Chellapilla 2002]. Seus autores exploraram o potencial de aprender jogar damas sem depender da usual inclusão de perícia humana [Chellapilla and Fogel 2001], na forma de características do jogo, que acreditam ser importantes para se jogar bem. Foi utilizada uma população de redes neurais na qual cada indivíduo serve como uma função de avaliação, para descrever a qualidade da posição corrente do jogo. Depois de pouco mais de 800 gerações, o processo evolucionário gerou uma rede neural que pode jogar damas no nível especialista.

Em vez de pesquisar em tabelas, bases de dados perfeitas de finais de jogos, aberturas de grandes mestres ou mesmo características sobre posições do tabuleiro que parecem ser importantes, ANACONDA considera a possibilidade de redes neurais aprenderem a jogar damas sem tais conhecimentos.

Segundo Fogel [Chellapilla and Fogel 2000], os esforços despendidos com técnicas de aprendizagem automática devem minimizar o montante de intervenção humana no processo.

Em seu trabalho foi utilizada a técnica de co-evolução para evoluir uma rede neural que pode avaliar tabuleiros de damas e jogar em nível de especialistas humanos.

### **3.4. Algoritmo Minimax**

Segundo [Fortes 2009], o algoritmo minimax é uma função recursiva de busca (semelhante ao método de busca em profundidade), criado por Von Neumann em 1928, que examina várias sequências de movimentos, a fim de encontrar uma que leve à vitória. O princípio básico dessa técnica é retornar uma jogada, de modo que ela reduza as possibilidades de seu adversário realizar uma boa jogada.

Conforme [Russell 2009], o algoritmo minimax toma uma decisão, partindo do estado atual da partida, analisando jogadas futuras e as avaliando. É usado um gerador de movimentos que cria uma árvore de jogadas possíveis (onde a posição atual do jogo é a raiz da árvore e as jogadas futuras são os nós filhos dessa raiz). Em seguida, a árvore

Este é um método que auxilia na tomada de decisão e é representado por dois agentes: MAX retrata as possíveis jogadas de um jogador controlado pela máquina e MIN, que representa os possíveis movimentos do jogador controlado pelo usuário. Esse algoritmo toma como verdade que o oponente (MIN) sempre fará a melhor jogada possível. Assim, a função de MAX é escolher a melhor jogada, supondo que MIN executará sempre sua melhor jogada.

Ao desenvolver o projeto espera-se atingir de forma eficiente, a melhor opção na árvore geradora, a melhor jogada que apresenta as seguintes maneiras:

Aumentando o nível da árvore de jogadas, o que melhorará a análise de cada movimento possível inserido nessa árvore, resultado no retorno de melhores jogadas.

A melhoria dos dois aspectos citados acima também proporcionará o aumento da efetividade do algoritmo Minimax, o qual terá mais possibilidades de jogadas a analisar. Reforça-se o cuidado para que essas oportunidades de melhoria não tornem o jogo lento, uma vez que a árvore de jogadas será maior, o tempo de resposta também será maior e consumo de memória aumentará consideravelmente.

Este trabalho propôs o desenvolvimento de uma ferramenta voltada ao jogo damas, onde o principal objetivo era a realização de uma partida entre humano máquina.

A principal dificuldade do projeto é gerar uma quantidade limitada de jogadas possíveis, de modo que as jogadas geradas fossem boas, que a análise fosse eficiente que o tempo de resposta do algoritmo fosse rápido.

O algoritmo Minimax é um bom algoritmo de busca, pois consegue analisar todas as possibilidades da árvore de jogo, entretanto seu tempo de execução pode ser muito alto, como resultado esperado da pesquisa é fazer uma busca mais eficiente com melhor escolha.

O planejamento dos procedimentos descritos ao longo do tempo são realizados no Cronograma de Execução. Geralmente o cronograma é disposto em uma tabela com as atividades e os meses.

[illegible]

Estudo dos artigos			x	x	x						
Pré-projeto				x	x						
Estudo das regras de damas					x	x					
Estudo do algoritmo Minimax					x	x					
Implementação do jogo						x	x	x			
Testes do jogo							x	x			
Redação do TCC II								x	x	x	
Revisão do TCC II									x	x	
Apresentação											x

## Referências

Schaeffer, J., Culberson, J. C., Treloar, N., Knight, B., Lu, P., and Szafron, D. (1992). A world championship caliber checkers program. *Artificial Intelligence*, 53(2-3):273–289.

Schaeffer, J., Lake, R., Lu, P., and Bryant, M. (1996). CHINOOK: The world man-machine checkers champion. *AI Magazine*, 17(1):21–29.

Samuel, A. L. (1959). Some studies in machine learning using the game of checkers.

Samuel, A. L. (1967). Some studies in machine learning using the game of checkers ii – recent progress.

RUSSELL, S.; NORVIG, P. *Artificial Intelligence: A Modern Approach*. 3. ed. [S.l.]: Prentice Hall, 2009.

Chellapilla, K. and Fogel, D. B. (2000). Anaconda defeats hoyle 6-0: A case study competing an evolved checkers program against commercially available software. In *Proceedings of the 2000 Congress on Evolutionary Computation CEC00*, pages 857–863, La Jolla Marriott Hotel La Jolla, California, USA. IEEE Press.

Fogel, D. B. and Chellapilla, K. (2002). Verifying anaconda’s expert rating by competing against chinook: experiments in co-evolving a neural checkers player. *Neuro-computing*, 42(1-4):69–86.

FORTES, R. P. *Web API para disponibilização de jogo educativos em ambientes virtuais*. Santa Catarina: Universidade do Vale do Itajaí, 2009.