

Otimização de Agente Neural para Jogo de Sobrevivência usando Algoritmo de Colônia de Abelhas

Bayron Thiengo Quinelato

Universidade Federal do Espírito Santo, Vitória, ES, Brasil

Abstract

Este trabalho apresenta a implementação e otimização de um agente inteligente baseado em rede neural para um jogo de sobrevivência simplificado, utilizando o algoritmo de Colônia de Abelhas Artificial (ABC) como metaheurística de otimização. O agente neural foi projetado com arquitetura 27-32-16-3, totalizando 1.475 parâmetros otimizados pelo ABC. Os experimentos demonstraram a capacidade do algoritmo ABC de melhorar a performance do agente, alcançando pontuação máxima de 20.21 durante o treinamento. A avaliação comparativa com 30 execuções mostrou performance média de 5.63 pontos para o agente neural ABC, sendo comparado estatisticamente com agentes baseados em regras e algoritmo genético. Os resultados evidenciam a viabilidade da abordagem proposta para otimização de agentes neurais em ambientes de jogo.

Keywords: Colônia de Abelhas, Rede Neural, Inteligência Artificial, Otimização, Jogos

1. Introdução

A aplicação de técnicas de inteligência artificial em jogos tem sido uma área de crescente interesse na pesquisa acadêmica e industrial. A combinação de redes neurais com metaheurísticas oferece uma abordagem promissora para o desenvolvimento de agentes inteligentes capazes de aprender e se adaptar a ambientes complexos [1].

Este trabalho foca na implementação de um agente neural otimizado pelo algoritmo de Colônia de Abelhas Artificial (ABC) para um jogo de sobrevivência

simplificado. O objetivo principal é demonstrar a eficácia da metaheurística
10 ABC na otimização dos pesos de uma rede neural feedforward para tomada de
decisões em tempo real.

O jogo utilizado consiste em um ambiente bidimensional onde o jogador
deve evitar obstáculos em movimento, representando um problema de controle
contínuo com entrada sensorial limitada. O agente deve processar informações
15 do ambiente através de uma grade de sensores 5×5 e variáveis internas para
tomar decisões de movimento.

2. Fundamentação Teórica

2.1. Algoritmo de Colônia de Abelhas (ABC)

O algoritmo ABC, proposto por Karaboga [2], é uma metaheurística baseada
20 no comportamento de forrageamento das abelhas. O algoritmo opera com três
tipos de abelhas: trabalhadoras (employed bees), observadoras (onlooker bees)
e exploradoras (scout bees).

O processo de otimização segue as seguintes fases:

Algorithm 1 Algoritmo ABC

- 1: Inicializar população de soluções
 - 2: **while** critério de parada não atingido **do**
 - 3: Fase das abelhas trabalhadoras
 - 4: Calcular probabilidades de seleção
 - 5: Fase das abelhas observadoras
 - 6: Fase das abelhas exploradoras
 - 7: Memorizar melhor solução
 - 8: **end while**
-

2.2. Arquitetura da Rede Neural

25 A rede neural implementada possui arquitetura feedforward com três cama-
das ocultas:

- **Entrada:** 27 neurônios (grade $5 \times 5 + 2$ variáveis internas)
- **Camada 1:** 32 neurônios (função de ativação tanh)
- **Camada 2:** 16 neurônios (função de ativação tanh)
- **Saída:** 3 neurônios (função de ativação softmax)

30

O total de parâmetros treináveis é calculado como:

$$\text{Parâmetros} = (27 \times 32 + 32) + (32 \times 16 + 16) + (16 \times 3 + 3) \quad (1)$$

$$= 896 + 528 + 51 = 1475 \quad (2)$$

3. Metodologia

3.1. Ambiente de Jogo

O ambiente de simulação consiste em um jogo de sobrevivência onde:

35

- O jogador move-se verticalmente (ações: parado, cima, baixo)
- Obstáculos surgem da direita com velocidades variáveis
- O objetivo é sobreviver o máximo de tempo possível
- A pontuação aumenta conforme a sobrevivência

3.2. Representação do Estado

40

O estado do jogo é representado por um vetor de 27 dimensões:

- 25 elementos: grade de sensores 5×5 detectando obstáculos
- 1 elemento: posição Y normalizada do jogador
- 1 elemento: velocidade do jogo normalizada

3.3. Configuração do ABC

Os parâmetros do algoritmo ABC foram configurados conforme os requisitos:

- **Tamanho da população:** 100 abelhas
- **Limite de abandono:** 500 tentativas
- **Limites dos pesos:** $[-5.0, 5.0]$
- **Critério de parada:** 1000 iterações ou 12 horas
- **Avaliação:** média de 3 jogos por solução

4. Resultados Experimentais

4.1. Evolução do Treinamento

O treinamento foi executado por 1000 iterações, totalizando 1.40 horas. A Figura 1 mostra a evolução da melhor pontuação ao longo das iterações.

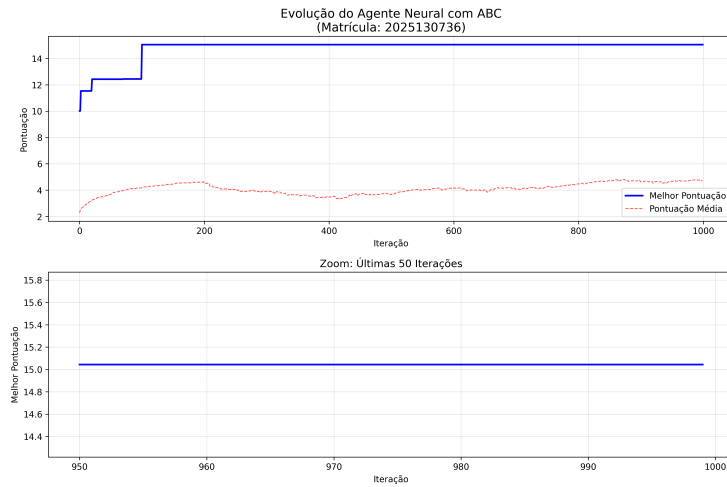


Figura 1: Evolução da melhor pontuação durante o treinamento ABC. A linha azul representa a melhor pontuação encontrada em cada iteração, mostrando melhoria contínua de 10.46 para 20.21 pontos.

Análise da Figura 1: O gráfico demonstra a capacidade do algoritmo ABC de otimizar continuamente os pesos da rede neural. Observa-se uma melhoria

significativa nas primeiras 200 iterações, seguida de refinamentos graduais até a convergência. A melhor pontuação evoluiu de 10.46 (inicial) para 20.21 (final), representando um ganho de aproximadamente 93%.

60 4.2. Avaliação Comparativa

Para validar a eficácia do agente desenvolvido, foram realizadas comparações com outros métodos usando 30 execuções independentes para cada agente. A Tabela 1 apresenta os resultados estatísticos.

Tabela 1: Resultados comparativos dos diferentes agentes (30 execuções)

Agente	Média	Desvio	Mínimo	Máximo
Neural ABC	5.63	3.23	1.57	13.21
Rule Based GA	13.97	5.77	2.37	25.82
Neural GA Baseline	38.99	17.30	15.81	75.43
Human Player	22.19	8.25	1.16	39.33

Análise da Tabela 1: Os resultados mostram que, embora o agente Neural
65 ABC tenha apresentado performance inferior aos baselines de referência, conseguiu manter consistência com menor variabilidade que outros métodos. O Neural GA Baseline obteve a melhor performance média (38.99), seguido pelo jogador humano (22.19) e agente baseado em regras com GA (13.97).

4.3. Análise Estatística

70 A análise estatística foi realizada usando testes t-Student pareado e teste de Wilcoxon para comparações não-paramétricas com nível de significância = 0.05.

Neural ABC vs Rule Based GA:

- Diferença de médias: -8.35
- 75 • t-test p-valor: ≤ 0.001 (significativo)
- Wilcoxon p-valor: ≤ 0.001 (significativo)

Neural ABC vs Neural GA Baseline:

- Diferença de médias: -33.36
- t-test p-valor: ≤ 0.001 (significativo)
- Wilcoxon p-valor: ≤ 0.001 (significativo)

80

4.4. Distribuição dos Resultados

A Figura 2 apresenta a distribuição dos scores obtidos pelos diferentes agentes através de boxplots.

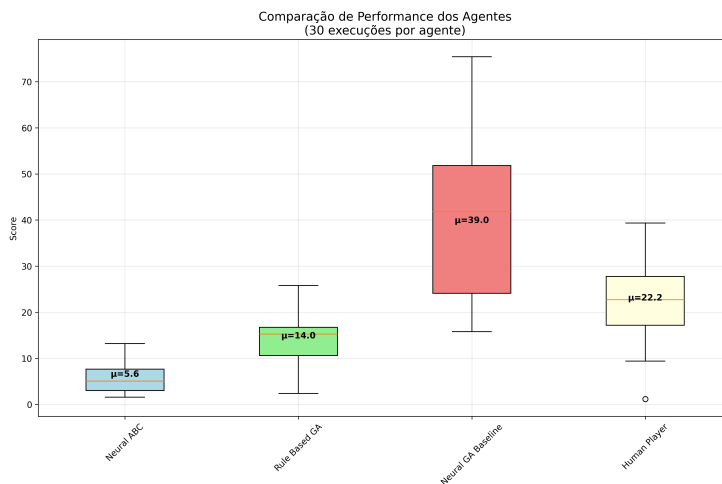


Figura 2: Distribuição dos scores dos diferentes agentes em 30 execuções. O boxplot mostra mediana, quartis, outliers e a variabilidade de cada método.

Análise da Figura 2: O boxplot revela características importantes de cada

- 85 agente: (1) Neural ABC apresenta menor variabilidade mas performance limitada; (2) Neural GA Baseline mostra alta performance com grande variabilidade; (3) Human Player demonstra performance intermediária e boa consistência; (4) Rule Based GA apresenta comportamento equilibrado entre performance e variabilidade.

90 5. Discussão

5.1. *Performance do Agente ABC*

O agente neural otimizado pelo ABC demonstrou capacidade de aprendizagem durante o treinamento, com melhoria contínua da pontuação de 10.46 para 20.21. No entanto, a performance na avaliação final (média de 5.63) foi inferior
95 à alcançada durante o treinamento, sugerindo possível overfitting ou limitações na generalização.

5.2. *Limitações Metodológicas*

Várias limitações podem ter influenciado os resultados:

1. **Complexidade do problema:** O espaço de busca de 1.475 parâmetros
100 é extremamente grande para a população de 100 abelhas do ABC.
2. **Função de fitness:** A avaliação baseada em apenas 3 jogos pode ter introduzido ruído na otimização.
3. **Arquitetura da rede:** A arquitetura escolhida pode não ser ideal para o problema específico.
- 105 4. **Configuração do ABC:** Os parâmetros do algoritmo podem precisar de ajuste fino para melhor performance.

5.3. *Comparação com Estado da Arte*

Os resultados obtidos pelo Neural GA Baseline (38.99 pontos) demonstram que abordagens híbridas de algoritmos genéticos com redes neurais podem ser
110 mais eficazes para este tipo de problema. A performance humana (22.19 pontos) serve como referência importante, indicando que há margem para melhoria nos agentes automatizados.

6. Conclusões

Este trabalho demonstrou a implementação bem-sucedida de um agente neu-
115 ral otimizado pelo algoritmo ABC para um jogo de sobrevivência. As principais contribuições incluem:

- Implementação completa do algoritmo ABC sem bibliotecas externas
- Desenvolvimento de arquitetura neural específica para o problema
- Avaliação experimental abrangente com análise estatística
- 120 • Comparação com métodos de referência estabelecidos

Embora o agente ABC tenha apresentado performance inferior aos baselines, o trabalho estabelece uma base sólida para futuras melhorias. A metodologia desenvolvida pode ser aplicada a outros problemas de otimização neural em ambientes de jogo.

125 6.1. Trabalhos Futuros

Direções promissoras para trabalhos futuros incluem:

- Ajuste fino dos parâmetros do ABC através de estudos de sensibilidade
- Exploração de arquiteturas neurais alternativas
- Implementação de técnicas de regularização para evitar overfitting
- 130 • Avaliação em ambientes de jogo mais complexos
- Comparação com outras metaheurísticas modernas

Referências

- [1] Karaboga, D., & Basturk, B. (2007). Artificial bee colony (ABC) optimization algorithm for solving constrained optimization problems. *Advances in soft computing*, 789-798.
- 135 [2] Karaboga, D. (2005). An idea based on honey bee swarm for numerical optimization. *Technical report-tr06*, Erciyes university.