

UNIVERSIDADE FEDERAL DE UBERLÂNDIA

Isabella Paula Simão

**Otimização de agentes inteligentes no jogo
Ludo por meio de um Algoritmo Genético**

Uberlândia, Brasil

2026

UNIVERSIDADE FEDERAL DE UBERLÂNDIA

Isabella Paula Simão

**Otimização de agentes inteligentes no jogo Ludo por
meio de um Algoritmo Genético**

Trabalho de conclusão de curso apresentado
à Faculdade de Computação da Universidade
Federal de Uberlândia, como parte dos requi-
sitos exigidos para a obtenção título de Ba-
charel em Sistemas de Informação.

Orientador: Professora Christiane Regina Soares Brasil

Universidade Federal de Uberlândia – UFU

Faculdade de Computação

Bacharelado em Sistemas de Informação

Uberlândia, Brasil

2026

Isabella Paula Simão

Otimização de agentes inteligentes no jogo Ludo por meio de um Algoritmo Genético

Trabalho de conclusão de curso apresentado à Faculdade de Computação da Universidade Federal de Uberlândia, como parte dos requisitos exigidos para a obtenção título de Bacharel em Sistemas de Informação.

Trabalho aprovado. Uberlândia, Brasil, 14 de março de 2026:

Professora Christiane Regina Soares
Brasil
Orientadora

Paulo Henrique Ribeiro Gabriel

João Henrique de Souza Pereira

Uberlândia, Brasil
2026

“Não é o mais forte que sobrevive, nem o mais inteligente. Quem sobrevive é o mais disposto à mudança”. - Charles Darwin

Resumo

Algoritmos Genéticos (AGs) são técnicas de otimização inspiradas nos princípios da evolução natural, amplamente utilizadas na resolução de problemas complexos caracterizados por grandes espaços de busca e alta variabilidade. No contexto de jogos de tabuleiro, especialmente aqueles que envolvem aleatoriedade e interação entre múltiplos jogadores, a definição manual de estratégias eficazes torna-se um desafio significativo. Este trabalho investiga a aplicação de Algoritmos Genéticos na otimização dos parâmetros de decisão de um agente inteligente no jogo Ludo, um ambiente estocástico no qual fatores probabilísticos, como o lançamento de dados, influenciam diretamente o desenrolar das partidas. Para esse fim, foi implementado um ambiente de simulação lógica e um agente baseado em heurísticas ponderadas, cuja estratégia é codificada por um cromossomo composto por cinco genes reais. Esses genes representam pesos associados a diferentes aspectos estratégicos do jogo, como progresso, ataque, segurança, risco e finalização. O Algoritmo Genético é empregado para evoluir esses pesos, sendo realizadas análises experimentais sistemáticas sobre diferentes configurações evolutivas, incluindo tamanho da população, número de gerações, métodos de seleção, elitismo, cruzamento e mutação. Os resultados obtidos demonstraram que o Algoritmo Genético é capaz de evoluir agentes com desempenho consistente e superior ao de agentes aleatórios, apresentando convergência estável e baixa variabilidade entre execuções independentes. A análise dos DNAs campeões revela padrões estratégicos recorrentes, indicando a emergência de comportamentos coerentes e interpretáveis. Dessa forma, o estudo evidenciou o potencial dos Algoritmos Genéticos como uma abordagem eficaz para a otimização de agentes inteligentes em jogos estocásticos, além de contribuir para a compreensão prática do impacto dos operadores genéticos nesse tipo de problema.

Palavras-chave: Algoritmos Genéticos; Jogos de Tabuleiro; Agentes Inteligentes; Otimização; Ludo.

Resumo

Genetic Algorithms (GAs) are optimization techniques inspired by the principles of natural evolution, widely used in solving complex problems characterized by large search spaces and high variability. In the context of board games, especially those involving randomness and interaction between multiple players, the manual definition of effective strategies becomes a significant challenge. This work investigates the application of Genetic Algorithms in optimizing the decision parameters of an intelligent agent in the game Ludo, a stochastic environment in which probabilistic factors, such as dice rolls, directly influence the course of the games. To this end, a logic simulation environment and an agent based on weighted heuristics were implemented, whose strategy is encoded by a chromosome composed of five real genes. These genes represent weights associated with different strategic aspects of the game, such as progress, attack, safety, risk, and completion. The Genetic Algorithm is used to evolve these weights, and systematic experimental analyses are performed on different evolutionary configurations, including population size, number of generations, selection methods, elitism, crossover, and mutation. The results obtained demonstrated that the Genetic Algorithm is capable of evolving agents with consistent performance superior to that of random agents, exhibiting stable convergence and low variability between independent runs. The analysis of the winning DNAs reveals recurring strategic patterns, indicating the emergence of coherent and interpretable behaviors. Thus, the study highlighted the potential of Genetic Algorithms as an effective approach for optimizing intelligent agents in stochastic games, in addition to contributing to the practical understanding of the impact of genetic operators in this type of problem.

Palavras-chave: Genetic Algorithms; Board Games; Intelligent Agents; Optimization; Ludo.

Lista de ilustrações

Figura 1 – Fluxograma do AG. Adaptado de (POZO et al., 2005)	15
Figura 2 – Cruzamento de um ponto. Adaptado de (POZO et al., 2005)	16
Figura 3 – Mutação em um <i>bit</i> . Adaptado de (POZO et al., 2005)	17
Figura 4 – Tabuleiro de ludo. Fonte: Ideas (2026)	19
Figura 5 – Distribuição de vitórias (agente campeão vs aleatório).	63

Lista de tabelas

Tabela 1 – Exemplos de perfis estratégicos baseados no DNA.	25
Tabela 2 – Resultados das execuções e síntese dos dados (População = 30).	42
Tabela 3 – Resultados das execuções e síntese dos dados (População = 50).	42
Tabela 4 – Resultados das execuções e síntese dos dados (População = 100).	43
Tabela 5 – Resultados das execuções e síntese dos dados (População = 150).	43
Tabela 6 – Resultados das execuções e síntese dos dados (População = 200).	44
Tabela 7 – Resultados das execuções e síntese dos dados (Gerações = 100).	45
Tabela 8 – Resultados das execuções e síntese dos dados (Gerações = 150).	45
Tabela 9 – Resultados das execuções e síntese dos dados (Gerações = 200).	46
Tabela 10 – Resultados das execuções e síntese dos dados (Gerações = 250).	46
Tabela 11 – Resultados das execuções e síntese dos dados (Elitismo = 0.04).	47
Tabela 12 – Resultados das execuções e síntese dos dados (Elitismo = 0.10).	48
Tabela 13 – Resultados das execuções e síntese dos dados (Elitismo = 0.15).	48
Tabela 14 – Resultados das execuções e síntese dos dados (Elitismo = 0.20).	49
Tabela 15 – Resultados das execuções e síntese dos dados (Seleção: Roleta).	50
Tabela 16 – Resultados das execuções e síntese dos dados (Seleção: Torneio de 2).	50
Tabela 17 – Resultados das execuções e síntese dos dados (Seleção: Torneio de 3).	51
Tabela 18 – Resultados das execuções e síntese dos dados (Seleção: Torneio de 5).	51
Tabela 19 – Resultados das execuções e síntese dos dados (Seleção: Torneio de 7).	52
Tabela 20 – Resultados das execuções e síntese dos dados (Cruzamento: Blocos).	53
Tabela 21 – Resultados das execuções e síntese dos dados (Cruzamento: Uniforme).	53
Tabela 22 – Resultados das execuções e síntese dos dados (Cruzamento: Uniforme Ponderado).	54
Tabela 23 – Resultados das execuções e síntese dos dados (Mutaç�o: Alfa Aleat�rio Probabil�stico).	56
Tabela 24 – Resultados das execuções e síntese dos dados (Mutaç�o: Sigma Aleat�rio).	56
Tabela 25 – Resultados das execuções e síntese dos dados (Mutaç�o: Perturbaç�o Alfa Aleat�rio).	57
Tabela 26 – Resultados das execuções e síntese dos dados (Mutaç�o: Sigma baseado em Fase).	57
Tabela 27 – Tabela comparativa deste AG com o do trabalho de refer�ncia.	59
Tabela 28 – An�lise do agente campe�o das melhores configura��es nos experimentos.	61
Tabela 29 – Compara��o de desempenho do agente campe�o com um aleat�rio.	63

Lista de abreviaturas e siglas

AE	Algoritmo Evolutivo
AG	Algoritmo Genético
CE	Computação Evolutiva
IA	Inteligência Artificial

Sumário

1	INTRODUÇÃO	11
1.1	Objetivos	12
1.2	Justificativa	12
1.3	Estrutura do texto	13
2	FUNDAMENTAÇÃO TEÓRICA	14
2.1	Computação Evolutiva	14
2.1.1	Algoritmos Genéticos	14
2.1.1.1	Operadores Genéticos	15
2.2	Agentes Inteligentes	17
2.3	Agentes Inteligentes com Algoritmos Genéticos	18
2.4	Ludo	19
2.5	Trabalhos correlatos	20
2.5.1	Evolução de estratégias adaptativas em um jogo de adivinhação	20
2.5.2	Otimização da função de avaliação em jogo de dominó	20
2.5.3	Algoritmos genéticos para análise estratégica em jogos de tabuleiro	21
2.5.4	Algoritmo genético aplicado ao problema do 8-puzzle	21
2.6	Considerações finais	22
3	DESENVOLVIMENTO	24
3.1	Descrição do Algoritmo Genético aplicado ao jogo Ludo	24
3.1.1	Representação do indivíduo	25
3.2	Geração inicial da população	26
3.3	Função de aptidão	27
3.4	Métodos de seleção	29
3.4.1	Seleção por torneio de N	30
3.4.2	Seleção por roleta	30
3.4.3	Elitismo	30
3.5	Operadores de cruzamento	30
3.5.1	Cruzamento uniforme padrão	30
3.5.2	Cruzamento uniforme ponderado	31
3.5.3	Cruzamento entre blocos semânticos	32
3.6	Operações de mutação	34
3.6.1	Mutação com alfa aleatório probabilístico	34
3.6.2	Mutação com perturbação alfa aleatória	35
3.6.3	Mutação com sigma aleatório	35

3.6.4	Mutação com sigma baseada em fase	36
3.7	Ciclo evolutivo	37
3.8	Considerações finais	39
4	RESULTADOS	41
4.1	Experimentos com o tamanho da população	41
4.1.1	Análise dos resultados	44
4.1.1.1	Interpretação do DNA campeão	44
4.2	Experimentos com o tamanho das gerações	45
4.2.1	Análise dos resultados	46
4.2.1.1	Interpretação do DNA campeão	47
4.3	Experimentos com a taxa de elitismo	47
4.3.1	Análise dos resultados	49
4.3.1.1	Interpretação do DNA campeão	49
4.4	Experimentos com métodos de seleção	49
4.4.1	Análise dos resultados	52
4.4.1.1	Interpretação do DNA campeão	52
4.5	Experimentos com métodos de cruzamento	52
4.5.1	Análise dos resultados	54
4.5.1.1	Interpretação do DNA campeão	54
4.6	Experimentos com métodos de mutação	55
4.6.1	Análise dos resultados	57
4.6.1.1	Interpretação do DNA campeão	58
4.7	Análise da configuração final	59
4.8	Teste de validação do agente campeão	62
4.9	Considerações finais	64
5	CONCLUSÃO	65
	REFERÊNCIAS	67

1 Introdução

A Inteligência Artificial (IA) tem sido amplamente aplicada em jogos digitais com o objetivo de desenvolver agentes capazes de tomar decisões mais coerentes, desafiadoras e menos previsíveis. Tradicionalmente, o comportamento desses agentes é definido por regras fixas ou por parâmetros ajustados manualmente, o que pode limitar sua capacidade de adaptação a ambientes dinâmicos e estocásticos. Jogos que envolvem aleatoriedade, interação entre múltiplos agentes e decisões estratégicas sucessivas apresentam um desafio adicional, uma vez que o desempenho de um agente depende não apenas de suas ações, mas também de eventos probabilísticos e das estratégias adotadas pelos adversários.

Nesse contexto, os Algoritmos Genéticos surgem como uma alternativa eficiente para a otimização automática de parâmetros de agentes inteligentes. Conforme definido por [Pacheco et al. \(1999\)](#), esses algoritmos constituem técnicas de busca e otimização inspiradas nos princípios darwinianos de seleção natural e reprodução genética. A cada geração, as soluções são avaliadas por meio de uma função de aptidão, permitindo que os indivíduos com melhor desempenho tenham maior probabilidade de influenciar as gerações subsequentes.

No domínio de jogos digitais, diversos trabalhos investigaram a aplicação de Algoritmos Genéticos para o controle de agentes e a otimização de estratégias de decisão. [Ontanón et al. \(2013\)](#), por exemplo, demonstraram o uso de abordagens evolutivas no controle de agentes em jogos de estratégia em tempo real, evidenciando sua capacidade de lidar com ambientes complexos e competitivos. De forma complementar, [Togelius et al. \(2011\)](#) discutiram o papel desses algoritmos na geração procedural de conteúdo, destacando sua utilidade na adaptação dinâmica de níveis e desafios.

Pesquisas mais recentes exploraram a integração de Algoritmos Evolutivos com outras técnicas de aprendizado, como o Aprendizado por Reforço, ampliando o escopo de aplicação desses métodos em jogos e sistemas multiagente ([BLOEMBERGEN et al., 2015](#); [BAI; CHENG; JIN, 2023](#)). Adicionalmente, estudos contemporâneos abordaram a adaptação de funções de avaliação em jogos de tabuleiro ([SALLABERRY; LADEIRA, 2020](#)), o uso de hiper-heurísticas evolutivas para criação de perfis estratégicos ([MENDES; TOGELIUS; NEALEN, 2020](#)) e a aplicação de algoritmos de horizonte rolante para lidar com a estocasticidade do ambiente ([LIU; AL., 2022](#)). Outros trabalhos focaram na robustez de agentes evoluídos em cenários de incerteza e competição multiagente ([ZHANG; AL., 2021](#); [PEREZ-LIEBANA; AL., 2021](#)).

Este Trabalho de Conclusão de Curso concentra-se na aplicação de Algoritmos Genéticos para a otimização das propriedades de um agente inteligente no jogo Ludo, um

jogo de tabuleiro estocástico que envolve decisões estratégicas, interação entre múltiplos jogadores e eventos aleatórios, como o lançamento de dados. O Ludo configura-se como um ambiente particularmente adequado para essa investigação, pois combina regras simples a uma dinâmica de jogo complexa, na qual pequenas variações nos parâmetros de decisão podem gerar comportamentos significativamente distintos e impactar de forma relevante o desempenho do agente ao longo das partidas.

Os parâmetros otimizados correspondem a pesos atribuídos a diferentes aspectos estratégicos do jogo, como progresso no tabuleiro, ações ofensivas (ataque), segurança das peças, exposição ao risco e estratégias de finalização. Ao ajustar esses pesos por meio de Algoritmos Genéticos, busca-se equilibrar esses fatores de maneira adaptativa, visando maximizar o desempenho do agente em cenários variados e incertos.

1.1 Objetivos

O objetivo principal deste trabalho é investigar a aplicação de Algoritmos Genéticos na otimização dos parâmetros que definem o comportamento de um agente inteligente no jogo Ludo, por meio de pesos atribuídos a diferentes estratégias voltadas à conquista da vitória. Para tanto, busca-se compreender de que forma distintas configurações evolutivas influenciam o desempenho do agente em um ambiente estocástico, no qual os resultados são naturalmente sujeitos à variabilidade. A proposta deste estudo é fundamentada em pesquisas anteriores que aplicaram Algoritmos Genéticos em jogos e problemas de tomada de decisão, tendo como referência principal o trabalho de [Schmiele \(2023\)](#), no qual a metodologia evolutiva demonstrou eficácia na adaptação de agentes em ambientes controlados, motivando sua investigação em um contexto distinto, especificamente no jogo Ludo.

1.2 Justificativa

Do ponto de vista acadêmico, o trabalho se justifica pela análise experimental do comportamento de um Algoritmo Genético em um ambiente de jogo específico, no caso o Ludo, permitindo observar como as escolhas de operadores e parâmetros podem influenciar o processo de otimização. Em vez de propor um modelo genérico, o estudo visa aprofundar a compreensão do funcionamento dessas técnicas em um cenário concreto, contribuindo para a discussão sobre sua aplicação em jogos estocásticos e reforçando a importância de avaliações empíricas cuidadosas.

Do ponto de vista global, embora o Ludo seja tradicionalmente visto como um jogo simples e amplamente baseado na sorte, sua estrutura revela paralelos com diversos problemas de otimização do mundo real. Ao observar suas dinâmicas, foi possível identificar elementos que refletem desafios concretos enfrentados por empresas, governos e indivíduos

em contextos de incerteza e competição.

Um dos paralelos mais evidentes está na logística de entregas de mercadorias. No jogo, cada peça pode ser comparada a um veículo ou a uma entrega em andamento. O dado representa fatores imprevisíveis, como trânsito, clima ou falhas operacionais. As casas seguras funcionam como centros de distribuição ou pontos estratégicos protegidos, enquanto a captura de uma peça simboliza atrasos ou perdas que comprometem a eficiência do sistema. O objetivo final, levar todas as peças ao centro, equivale a minimizar o tempo total de operação sob condições incertas.

Deste modo, o Ludo pode ser interpretado como um modelo simplificado de decisões sob incerteza, envolvendo alocação de recursos limitados, competição estratégica e gestão de risco. Ainda que o elemento aleatório seja central no jogo, a forma como o jogador distribui seus movimentos influencia significativamente suas chances de sucesso, assim como no mundo real, em que não se controlam as variáveis externas, mas se pode otimizar as escolhas diante delas.

1.3 Estrutura do texto

Este trabalho está organizado da seguinte forma:

- O Capítulo 2 apresenta os conceitos fundamentais relacionados aos Algoritmos Genéticos, abordando sua origem, princípios de funcionamento e principais operadores, bem como aspectos relevantes da Inteligência Artificial aplicada a jogos.
- O Capítulo 3 descreve a metodologia adotada, detalhando o jogo Ludo, a modelagem do agente inteligente, a representação genética, a função de aptidão e a configuração dos Algoritmos Genéticos utilizados.
- O Capítulo 4 apresenta os experimentos realizados e a análise dos resultados obtidos, com foco na comparação entre diferentes estratégias evolutivas.
- Por fim, o Capítulo 5 discute as conclusões do trabalho e aponta possíveis direções para pesquisas futuras.

2 Fundamentação Teórica

Este capítulo apresenta os fundamentos teóricos que embasam o desenvolvimento deste Trabalho de Conclusão de Curso. São discutidos os principais conceitos relacionados aos Algoritmos Genéticos, incluindo seus princípios de funcionamento e operadores fundamentais, bem como a definição de agentes inteligentes e suas aplicações no contexto de jogos. Além disso, são analisados trabalhos correlatos que serviram de base metodológica e conceitual para a construção do agente inteligente aplicado ao jogo Ludo, permitindo a contextualização do problema investigado e a justificativa das escolhas adotadas ao longo da pesquisa.

2.1 Computação Evolutiva

A Computação Evolutiva (CE) mimetiza processos evolutivos que ocorrem na natureza por meio de heurísticas computacionais¹, a fim de encontrar soluções ótimas ou aproximadas de problemas de alta complexidade. A CE fundamenta-se na Teoria da Evolução das Espécies, proposta por Charles Darwin, cujo princípio central é a seleção natural (DARWIN, 1859). De acordo com Pila (2006), a seleção natural é o mecanismo essencial que garante a preservação dos indivíduos mais aptos ao longo das gerações. No ambiente computacional, isso se traduz em criar uma população de indivíduos (possíveis soluções) que vão reproduzir e competir; em que os melhores sobrevivem e transferem suas características às novas gerações.

2.1.1 Algoritmos Genéticos

Os Algoritmos Genéticos (AGs) são uma classe de busca estocástica fundamentada na teoria de seleção natural e nos operadores genéticos, tais como cruzamento e mutação (ANTONIO, 2011). Formalmente, os AGs foram introduzidos por Holland (1975), e a primeira aplicação prática desses algoritmos em problemas reais de otimização foi implementada e publicada por Goldberg (1989).

O funcionamento de um AG busca o equilíbrio entre os processos de exploração e exploração. A exploração é a atividade por meio da qual o algoritmo de busca tenta explorar o máximo possível do amplo espaço de busca para evitar cair em armadilhas de ótimos locais. Por outro lado, a exploração representa a atividade na qual o algoritmo de busca tenta desenvolver as melhores soluções descobertas por meio de abordagens direcionadas (EL-OMARI, 2020).

¹ Uma heurística computacional é uma técnica baseada em regras ou estimativas utilizada para buscar soluções ótimas ou aproximações, de modo eficiente, para problemas complexos (GOLDBERG, 1989).

Estabelecer este equilíbrio consiste em decidir adequadamente sobre a necessidade ou não de experimentar novas situações (explorar novas áreas do espaço de solução), em detrimento daquelas já vivenciadas (JUNIOR, 2009). Assim como na natureza, o algoritmo não tem a informação de qual problema esteja resolvendo, mas consegue, por meio de uma função de avaliação, fazer com que somente os indivíduos com maiores condições de chegar ao resultado (ou seja, os mais aptos) consigam sobreviver e se reproduzir (VILAS-BOAS, 2016).

Nas próximas subseções, são detalhados os operadores genéticos comumente utilizados pelos Algoritmos Genéticos.

2.1.1.1 Operadores Genéticos

Os operadores genéticos de AG buscam a melhor solução (ou uma aproximação da mesma) utilizando-se de um processo iterativo e evolutivo. Deste modo, a partir de uma população inicial, recombina-se os indivíduos pais que foram selecionados, a fim de gerar novas soluções que serão inseridas em uma nova população (VILAS-BOAS, 2016). A Figura 1 ilustra o funcionamento do AG, com a sequência da aplicação dos seus operadores genéticos.

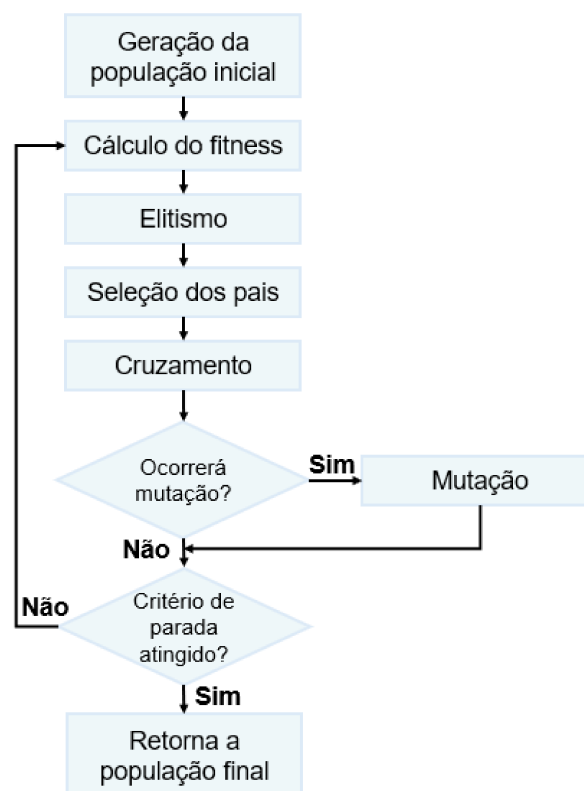


Figura 1 – Fluxograma do AG. Adaptado de (POZO et al., 2005)

A seguir, são apresentados os principais conceitos de um AG e os operadores genéticos correspondentes.

- **População Inicial:** A população inicial de um AG é o conjunto de indivíduos candidatos à solução de um dado problema, criada geralmente de modo aleatório. Estes indivíduos serão usados para construir a nova população para a próxima geração por meio dos operadores genéticos e da aptidão de cada possível solução (POZO et al., 2005).
- **Função de Aptidão (*Fitness*):** A função de aptidão é utilizada para determinar quão bem as soluções encontradas respondem ao problema proposto. Portanto, ela é usada para medir a habilidade do indivíduo de sobreviver e se reproduzir (VILAS-BOAS, 2016). Para cada problema, há uma função de objetivo elaborada para qualificar as possíveis soluções geradas pelo algoritmo.
- **Seleção:** O operador genético de seleção define quais indivíduos serão escolhidos para reprodução. Diferentemente de uma escolha determinística baseada apenas nos melhores valores de *fitness*, os métodos de seleção podem variar conforme a estratégia adotada, dentre os quais destacam-se: seleção por *ranking*, seleção por roleta, seleção por torneio e seleção uniforme (LUCAS, 2002). A seleção não garante necessariamente que apenas os indivíduos de maior aptidão sejam escolhidos, permitindo a manutenção da diversidade genética e evitando a convergência prematura (ANTONIO, 2011).
- **Cruzamento (*Crossover*):** O operador genético de cruzamento tem por finalidade combinar as informações genéticas de dois indivíduos (pais) selecionados, para gerar novos indivíduos (filhos), que serão inseridos na nova geração (VILAS-BOAS, 2016). Os mais conhecidos operadores tradicionais de cruzamento são: cruzamento de um ponto (1PX), cruzamento multiponto (MPX), cruzamento segmentado (SX), cruzamento uniforme (UX) e o cruzamento por combinação parcial (PMX) (LUCAS, 2002). A técnica de cruzamento ilustrada pelo exemplo da Figura 2 é chamada cruzamento de um ponto, ou cruzamento simples, pois é escolhido apenas um ponto de cruzamento (ANTONIO, 2011).

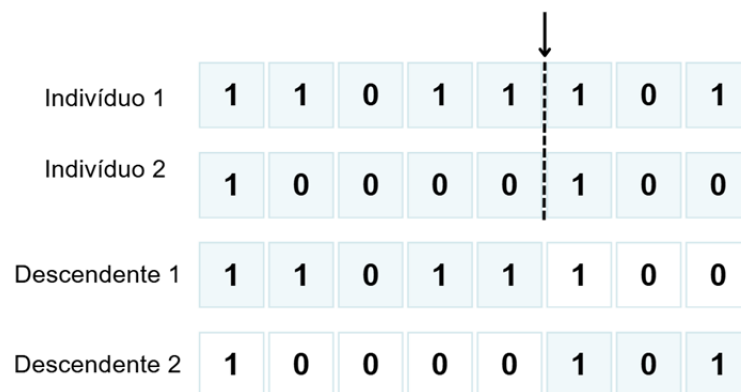


Figura 2 – Cruzamento de um ponto. Adaptado de (POZO et al., 2005)

- **Mutação:** A mutação é um operador genético que tem a função de facilitar a busca local e introduzir novos elementos, garantindo a diversidade genética da população (ANTONIO, 2011). Por meio deste operador, busca-se recuperar boas características, eventualmente perdidas nos processos de seleção e cruzamento, e também evitar ficar preso em determinadas regiões do espaço de busca ou em pontos de ótimo locais (VILAS-BOAS, 2016). Existem vários operadores para o processo de mutação: mutação em um ponto, mutação em múltiplos pontos, mutação probabilística, mutação Lamarckiana e mutação nula são alguns exemplos (ANTONIO, 2011). Na Figura 3 é ilustrada a mutação em um ponto, que modifica apenas um valor do cromossomo em uma posição gerada aleatoriamente (ANTONIO, 2011).

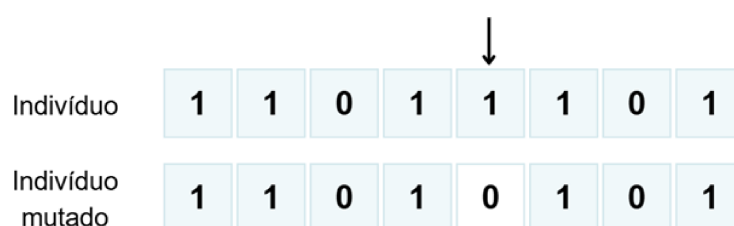


Figura 3 – Mutação em um *bit*. Adaptado de (POZO et al., 2005)

- **Elitismo:** O elitismo é um recurso no processo evolutivo que preserva a informação genética de uma taxa dos melhores indivíduos dentre os ascendentes, repassando-a diretamente para a próxima geração (VILAS-BOAS, 2016).

Nas próximas seções, são definidos os agentes inteligentes e apresentados dentro do contexto dos Algoritmos Genéticos.

2.2 Agentes Inteligentes

Computacionalmente, um agente é um sistema fundamentado em uma base de conhecimento, que percebe um ambiente, seja este um mundo físico, representação gráfica, uma coleção de outros agentes ou ambientes complexos; interpreta estas percepções; desenvolve inferências; resolve problemas e determina ações, agindo sobre o ambiente para realizar um conjunto de metas ou tarefas para as quais foi projetado (GOEBLER, 2004).

Portanto, pode-se definir como agente inteligente aquele que toma uma decisão racional no momento de efetuar uma escolha. Esta decisão deve ser tomada considerando o que o agente constatou durante sua pesquisa atual e suas pesquisas anteriores (GOEBLER, 2004).

Em jogos digitais, a Inteligência Artificial permite que a interação com estes agentes seja mais dinâmica, passando ao jogador a sensação de que a máquina está “pen-

sando” e se comportando como um ser humano na tentativa de vencê-lo (PEGORIN; MENECELLI; SILVA, 2022).

2.3 Agentes Inteligentes com Algoritmos Genéticos

A partir da integração entre os conceitos de agentes inteligentes e Algoritmos Genéticos, torna-se possível desenvolver sistemas autônomos capazes de adaptar seu comportamento de forma automática. Em vez de um programador detalhar o comportamento do agente, o processo evolutivo descobre as melhores estratégias por si mesmo (CAMPOS; PEREIRA, 2019).

A combinação de técnicas evolutivas e agentes tem se mostrado promissora para a criação de sistemas que se adaptam e evoluem em ambientes complexos e dinâmicos (NETO, 2023). Como exemplos de trabalhos que fazem o uso dessa combinação, tem-se a investigação de Pegorin, Menechelli e Silva (2022), que propõe uma metodologia para a evolução de pesos em agentes de jogos de tabuleiro, evidenciando como o processo de seleção natural pode refinar heurísticas de decisão. Na mesma linha de pesquisa, Martins et al. (2021) investiga a aplicação de algoritmos genéticos em agentes de busca em ambientes incertos, demonstrando que a evolução populacional supera abordagens baseadas em regras rígidas no quesito de adaptabilidade. Complementarmente, Medina e Müller (2009) aplica esses conceitos em jogos de estratégia clássicos, utilizando o AG para gerar comportamentos autônomos que aprendem a reagir a táticas adversárias por meio da evolução de suas funções de avaliação.

No domínio de jogos, a área central deste trabalho, essa combinação é particularmente proeminente. A computação evolutiva tem sido aplicada para desenvolver a inteligência de agentes em estilos de jogo específicos, como em cenários do tipo presa-predador (SILVA, 2019). Além disso, em jogos de tabuleiro com grandes espaços de busca, como o Xadrez e as Damas, algoritmos genéticos são utilizados para otimizar as funções de avaliação heurística, permitindo que os agentes determinem a importância relativa de fatores como controle de centro e mobilidade das peças (FOGEL, 2002). No jogo de Go, técnicas evolutivas têm sido integradas para aprimorar a seleção de jogadas em árvores de decisão complexas (RICHARDS; MORIARTY; MIIKKULAINEN, 1998).

A integração com o aprendizado por reforço profundo tem permitido a criação de experiências mais complexas e desafiadoras (WANG et al., 2025) e a otimização de estratégias em jogos de furtividade (SILVA, 2024). Essa linha de pesquisa se estende ao uso de agentes em ambientes de aprendizado baseados em jogos, onde agentes colaboram com o usuário em processos de ensino (DYULICHEVA; GLAZIEVA, 2021).

A seguir, é apresentada a descrição do jogo de tabuleiro Ludo, que é a aplicação do AG neste Trabalho de Conclusão de Curso.

2.4 Ludo

O Ludo é um jogo de tabuleiro de percurso para dois a quatro jogadores, cujo objetivo é conduzir quatro peões da base inicial até a área final do tabuleiro, por meio de lançamentos de um dado de seis faces. O jogo apresenta regras simples, porém um espaço de estados complexo, resultante da interação entre múltiplos jogadores, da aleatoriedade dos dados e das decisões estratégicas envolvidas.

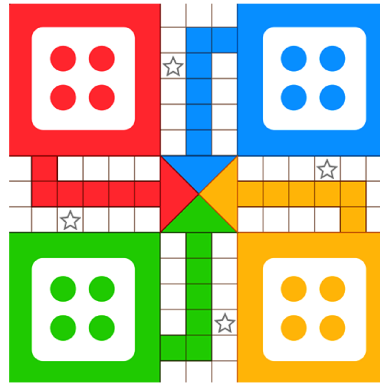


Figura 4 – Tabuleiro de ludo. Fonte: [Ideas \(2026\)](#)

A configuração visual e funcional do tabuleiro pode ser observada na Figura 4:

- **Bases (Ponto de Partida):** Representadas pelos quatro grandes quadrados coloridos nos cantos do tabuleiro (verde, vermelho, azul e amarelo). Cada jogador inicia com suas quatro peças posicionadas em sua respectiva base, onde permanecem no estado base até que um valor específico no dado (o número 6) permita que a peça entre no percurso principal.
- **Percurso Principal:** É o caminho composto pelas casas brancas e as casas de saída. É neste trajeto que ocorre a interação direta entre os agentes, permitindo a captura de oponentes caso uma peça finalize seu movimento em uma casa ocupada por um adversário.
- **Casas Seguras:** No tabuleiro, são identificadas pelo símbolo de uma estrela. Estas casas impedem que a peça seja capturada e enviada de volta à base.
- **Reta Final:** A reta final é caracterizada pelas colunas coloridas que levam ao centro do tabuleiro. Uma peça entra nesta reta após completar (quase) uma volta inteira no percurso principal.
- **Triângulo Central (Ponto de Chegada):** O centro do tabuleiro é o objetivo final. Para que uma peça atinja o estado final, o jogador deve obter o valor exato no dado para alcançar a última casa da reta final. O vencedor é o primeiro a levar todas as suas quatro peças para esta área.

Durante uma partida, o jogador deve decidir qual peça mover a cada turno, considerando fatores como progressão no tabuleiro, risco de captura por adversários, possibilidade de capturar peças oponentes e bloqueio de caminhos. Essas decisões impactam diretamente o desempenho ao longo do jogo, tornando o Ludo um ambiente propício para o estudo de estratégias adaptativas.

Além da aleatoriedade inerente ao lançamento dos dados, o jogo exige a avaliação contínua do estado do tabuleiro, o que envolve a análise de múltiplas variáveis estratégicas. Essa combinação de fatores torna o Ludo um problema desafiador para agentes inteligentes, justificando sua utilização como ambiente de aplicação neste trabalho.

Na próxima seção, são descritos brevemente alguns trabalhos relacionados à área de pesquisa deste Trabalho de Conclusão de Curso.

2.5 Trabalhos correlatos

Esta seção tem como objetivo apresentar trabalhos da literatura científica, no qual seus autores buscaram aplicar Algoritmos Genéticos para solucionar problemas de diferentes tipos de jogos. Essas pesquisas foram selecionadas em razão de sua proximidade temática em relação com o presente trabalho, e sua análise proporcionou uma sólida base teórica para a compreensão e o adequado desenvolvimento deste TCC.

2.5.1 Evolução de estratégias adaptativas em um jogo de adivinhação

O trabalho de [Wittkamp e Barone \(2006\)](#) aplicou estratégias adaptativas para o jogo de adivinhação “Spoof” com objetivo de desenvolver um agente que pudesse se adaptar às táticas dos jogadores humanos, mantendo um nível de desafio constante. Para tal, os autores utilizaram a Programação Genética² para criar e evoluir um programa que definisse o comportamento do agente. A evolução do programa, que representava a estratégia do agente, foi baseada em um processo de seleção e variação que favorecia os programas mais bem-sucedidos. Os resultados mostraram que o agente foi capaz de evoluir sua estratégia ao longo do tempo, adaptando-se às jogadas do oponente e demonstrando um comportamento inteligente e imprevisível.

2.5.2 Otimização da função de avaliação em jogo de dominó

O trabalho de [Antonio \(2011\)](#) tem como objetivo principal a otimização da função de avaliação de um agente para o jogo de Dominó de 4 pontas, utilizando um Algoritmo Genético. A motivação para o trabalho foi a necessidade de criar um agente com um

² Dentro da grande área da CE, existe a classe dos Algoritmos Evolutivos, que abrange as subclasses: Programação Genética, Algoritmos Genéticos e Estratégias Evolutivas ([GABRIEL; DELBEM, 2008](#)).

comportamento mais estratégico, superando o oferecido por abordagens de força bruta. Para isso, o autor modelou a função de avaliação como um conjunto de pesos que são ajustados pelo AG, buscando encontrar a combinação que gere um melhor desempenho do agente em partidas simuladas. O método empregado consistiu na criação de um ambiente de jogo virtual onde o AG foi aplicado para evoluir os pesos da função de avaliação do agente. A evolução ocorreu por meio de sucessivas gerações, onde a população de indivíduos foi avaliada pela sua aptidão (desempenho em partidas contra outros agentes). Os resultados obtidos demonstraram que o AG foi capaz de otimizar a função de avaliação de forma satisfatória, resultando em um agente com um comportamento estratégico e eficiente.

2.5.3 Algoritmos genéticos para análise estratégica em jogos de tabuleiro

O trabalho de [Schmiele \(2023\)](#) teve como objetivo aplicar AGs para a análise estratégica de jogos de tabuleiro, otimizando os parâmetros de um agente de jogo, visando aprimorar as estratégias mais eficazes em um ambiente de regras bem definidas e complexas. A metodologia do trabalho envolveu a utilização de um ambiente de simulação onde o AG explorava o espaço de busca de estratégias, testando diferentes configurações para o agente. Os resultados do estudo demonstraram que o AG foi capaz de encontrar soluções otimizadas que superaram o desempenho de abordagens não evolutivas, mostrando a capacidade dos AGs de explorar o espaço de busca de estratégias e encontrar soluções eficientes.

2.5.4 Algoritmo genético aplicado ao problema do 8-*puzzle*

O trabalho de [Muniz \(2025\)](#) apresentou a implementação e análise de um AG para a solução do problema do quebra-cabeça de 8 peças, visando investigar a eficiência desse método em problemas de busca combinatória. O objetivo principal foi desenvolver uma abordagem que combinasse operadores de seleção, *crossover* e mutação para encontrar a solução ótima ou próximas da ótima em diferentes configurações do quebra-cabeça. Os experimentos demonstraram que o AG foi capaz de encontrar soluções viáveis com número limitado de movimentos e apresentou desempenho consistente em diferentes taxas de mutação, *crossover* e elitismo, evidenciando a importância do equilíbrio entre exploração e diversidade da população. Os resultados obtidos mostraram-se bastante satisfatórios quando comparados à literatura, confirmando a relevância do ajuste adequado dos parâmetros.

As principais referências metodológicas deste trabalho foram: inicialmente as primeiras publicações da área de [Holland \(1975\)](#) e [Goldberg \(1989\)](#) que auxiliaram no entendimento do Algoritmo Genético; e posteriormente o estudos de [Antonio \(2011\)](#) e [Schmiele \(2023\)](#), que serviram como base conceitual e estrutural para a modelagem do agente in-

teligente e para a definição da função de avaliação utilizada no processo evolutivo do AG em questão.

A relevância dos trabalhos de [Antonio \(2011\)](#) e [Schmiele \(2023\)](#) para esta pesquisa está, sobretudo, na abordagens adotadas para representar estratégias de jogo por meio de um vetor de parâmetros, cuja combinação influencia diretamente a tomada de decisão do agente. Essas modelagens inspiraram diretamente a estrutura da representação do indivíduo implementado neste trabalho, bem como a utilização de partidas simuladas como mecanismo de avaliação da aptidão dos indivíduos. A semelhança fundamental entre ambas as obras, que se conecta com o propósito deste trabalho, reside na tradução de heurísticas subjetivas em valores numéricos (genes), permitindo que o processo evolutivo quantifique a importância de cada ação.

Além disso, o estudo de [Schmiele \(2023\)](#) forneceu um referencial claro sobre a importância do equilíbrio entre exploração e exploração ao longo do processo evolutivo, evidenciando como a escolha adequada de operadores genéticos, taxas de mutação e estratégias de seleção impacta o desempenho final do agente. Esses aspectos foram considerados e adaptados ao contexto específico do jogo Ludo, respeitando suas particularidades estratégicas e dinâmicas de estado, o que reforça a aderência metodológica entre os trabalhos.

2.6 Considerações finais

Este capítulo apresentou os fundamentos teóricos necessários para a compreensão e contextualização do trabalho desenvolvido. Foram discutidos os conceitos centrais da Computação Evolutiva, com ênfase nos Algoritmos Genéticos, seus princípios de funcionamento e operadores genéticos, como seleção, cruzamento, mutação e elitismo. Esses conceitos forneceram o embasamento conceitual para o desenvolvimento do AG ao longo deste estudo.

As principais referências que sustentaram este Trabalho de Conclusão de Curso foram os estudos clássicos de [Holland \(1975\)](#) e [Goldberg \(1989\)](#), responsáveis por estabelecer a fundamentação teórica dos Algoritmos Genéticos, incluindo os conceitos de seleção, cruzamento, mutação, função de aptidão e representação genética das soluções. Essas obras construíram a base conceitual necessária para compreender o funcionamento dos Algoritmos Genéticos como métodos de busca e otimização inspirados nos princípios da evolução natural.

Complementarmente, as pesquisas de [Antonio \(2011\)](#) e [Schmiele \(2023\)](#) foram cruciais para este trabalho por apresentarem aplicações práticas dos Algoritmos Genéticos no contexto de jogos de tabuleiro, demonstrando como essas técnicas podem ser utilizadas para otimizar funções de avaliação e estratégias de tomada de decisão de agentes artificiais.

Desse modo, o referencial teórico apresentado estabeleceu uma base sólida para o desenvolvimento do capítulo subsequente, no qual são detalhados o desenvolvimento do agente evolutivo, a modelagem do indivíduo, os operadores genéticos implementados e as escolhas de projeto realizadas para adaptar os Algoritmos Genéticos ao contexto específico do jogo Ludo.

3 Desenvolvimento

Este capítulo descreve de forma detalhada o desenvolvimento do Algoritmo Genético proposto para a otimização de um agente inteligente no jogo de tabuleiro Ludo. Além disso, também é apresentada uma justificativa sobre as escolhas metodológicas adotadas, especialmente no que se refere aos operadores genéticos de cruzamento e mutação.

Este trabalho foi implementado em um computador *Dell Inspiron 15 3520*, equipado com processador *Intel® Core™ i5-1235U* de 12ª geração (1.30 GHz), 16 GB de memória RAM e sistema operacional *Windows 64 bits*. A plataforma de desenvolvimento utilizada foi o Google Colab, que possibilitou a execução dos códigos e experimentos em ambiente de nuvem, garantindo praticidade e execuções simultâneas. O algoritmo foi implementado em Python 3.10, utilizando bibliotecas amplamente consolidadas para computação científica, como *NumPy*, além de ferramentas próprias para simulação do ambiente do jogo.

A modelagem e o desenvolvimento do agente inteligente propostos neste capítulo têm como inspiração principal o trabalho de (SCHMIELE, 2023), que aplicou Algoritmos Genéticos (AGs) para a otimização de parâmetros de agentes no jogo *Monopoly* (Banco Imobiliário). Assim como no Ludo, o ambiente explorado no estudo de referência compartilha a alta complexidade da estocasticidade, onde o estado do jogo e o planejamento a curto prazo são fortemente influenciados pela incerteza do lançamento de dados. Inspirando-se nessa premissa — na qual um ambiente de simulação lógica, sem interface gráfica, é construído para que o AG explore o espaço de busca e encontre estratégias resilientes ao acaso —, esta pesquisa adota uma estrutura metodológica estruturalmente similar. Contudo, a arquitetura de decisão e a função de avaliação são inteiramente projetadas e modeladas para as dinâmicas e regras específicas de conflito e movimentação do jogo Ludo.

O código-fonte completo está disponível publicamente no GitHub, com acesso por meio do seguinte link: <https://github.com/isabellapaulaS/TCC-Ludo-GA/tree/main>

3.1 Descrição do Algoritmo Genético aplicado ao jogo Ludo

O Algoritmo Genético implementado neste trabalho, voltado para o jogo Ludo, segue a estrutura clássica descrita na literatura: inicialização da população, avaliação de aptidão, seleção, aplicação de operadores de cruzamento e mutação, elitismo e geração da nova população. O ciclo evolutivo deve ser repetido até que seja atingido um número máximo de gerações.

Cada indivíduo da população representa uma estratégia de jogo, codificada por um

vetor de cinco genes reais. Esses genes correspondem aos pesos atribuídos a heurísticas de decisão do agente durante o jogo, influenciando diretamente a escolha das jogadas realizadas. A seguir, a representação do indivíduo é descrita mais detalhadamente.

3.1.1 Representação do indivíduo

Neste AG, o cromossomo de cada indivíduo é composto por cinco genes reais (ou seja, o DNA do agente), cada um associado a um aspecto estratégico do jogo Ludo:

1. **Peso de progresso:** prioriza o avanço das peças ao longo do tabuleiro.
2. **Peso de ataque:** valoriza jogadas que resultam na captura de peças adversárias.
3. **Peso de segurança:** penaliza movimentos que expõem as próprias peças a riscos.
4. **Peso de risco:** controla a tolerância do agente a jogadas potencialmente perigosas.
5. **Peso de finalização:** prioriza a conclusão do percurso das peças.

Durante uma partida, todas as jogadas possíveis são avaliadas com base nesses pesos, sendo selecionada aquela que apresenta a maior pontuação agregada.

Os valores de cada gene são limitados ao intervalo contínuo $[-5.0, 5.0]$, garantindo estabilidade numérica e evitando comportamentos extremos não realistas. Esses valores foram estabelecidos por meio de uma análise empírica durante o desenvolvimento do sistema.

Tabela 1 – Exemplos de perfis estratégicos baseados no DNA.

Perfil	Progresso	Ataque	Segurança	Risco	Chegada
Corredor	5.0	0.2	5.0	3.5	4.8
Predador	2.0	5.0	3.3	4.0	1.0
Defensivo	3.2	0.7	5.0	-5.0	3.8
Agressivo	5.0	5.0	1.0	4.7	4.9
Cauteloso	3.9	3.2	3.1	-2.0	2.8

A lógica de decisão baseia-se na premissa de que o agente sempre escolherá o movimento com a maior pontuação agregada. A seguir, apresenta-se uma análise detalhada de como os pesos configuram cada comportamento estratégico:

- **Corredor:** A predominância dos valores máximos nos genes de progresso (5.0) e chegada (4.8) justifica a descrição de um perfil focado no avanço linear e na conclusão do jogo. O valor de segurança (5.0) atua como um contrapeso crítico; mesmo visando a velocidade, o agente atribui uma pontuação altíssima a movimentos que terminam

em casas seguras para evitar o retrocesso punitivo de ser capturado. A tolerância a riscos é moderada-alta (3.5), indicando que ele aceita perigos se o ganho de terreno for expressivo.

- **Predador:** Este perfil é definido pelo peso máximo no gene de ataque (5.0), o que faz com que qualquer jogada que resulte em captura receba a maior pontuação possível na avaliação do agente. O valor de segurança (3.3) é mantido em nível intermediário para garantir que o “caçador” não se torne vulnerável de forma trivial, enquanto o baixo peso em chegada (1.0) confirma que seu objetivo primário não é terminar o percurso rapidamente, mas sim controlar o tabuleiro através da eliminação de oponentes.
- **Defensivo:** A característica marcante deste perfil é o valor extremo negativo de risco (-5.0) combinado com a segurança máxima (5.0). Na prática, isso significa que qualquer movimento que termine em uma casa vulnerável sofrerá uma penalidade tão severa na pontuação que o agente preferirá movimentos curtos ou até a permanência na base se não houver um trajeto totalmente seguro. O foco no progresso (3.2) é apenas secundário à sua integridade física.
- **Agressivo:** Diferente do predador, o agressivo busca o domínio total. Os valores de 5.0 em progresso e ataque, aliados a um peso de chegada de 4.9, criam um agente que avança batendo. O peso de 1.0 em segurança mostra que ele negligencia a própria defesa, enquanto o valor de risco (4.7) demonstra que ele ignora ameaças iminentes se houver chance de progredir ou capturar, priorizando ganhos imediatos e agressivos sobre a sobrevivência a longo prazo.
- **Cauteloso:** Este perfil apresenta uma distribuição mais uniforme, com pesos em torno de 3.0 para os fatores positivos. O valor negativo de risco (-2.0) é fundamental para caracterizar o "perfil padrão": ele busca o avanço e o ataque, mas possui um filtro de cautela que o impede de realizar jogadas imprudentes, buscando uma média ponderada que maximize a vitória sem exposição desnecessária.

3.2 Geração inicial da população

Neste trabalho, a população inicial é gerada de forma aleatória, garantindo alta diversidade genética no início do processo evolutivo. Cada gene de cada indivíduo é inicializado com um valor aleatório uniforme dentro do intervalo fechado de -5 a 5.

Essa abordagem evita vieses iniciais e permite que o algoritmo explore um amplo espaço de estratégias possíveis desde as primeiras gerações.

3.3 Função de aptidão

A função de aptidão (*fitness*) é o principal mecanismo de avaliação do desempenho dos indivíduos. Devido à natureza estocástica do jogo Ludo, onde um único lançamento de dados pode alterar drasticamente o resultado, o *fitness* de cada indivíduo é calculado como a média do desempenho em 5 partidas simuladas. Esta abordagem estabiliza o valor de aptidão, permitindo que o algoritmo diferencie estratégias consistentemente boas de vitórias acidentais. Esse valor foi determinado por meio de uma análise empírica durante o desenvolvimento do sistema.

Em cada avaliação, o agente evolutivo compete contra três oponentes fixos, cujas estratégias são baseadas nos perfis agressivo, cauteloso e corredor. A escolha destes três perfis específicos, em detrimento da multiplicidade exemplificada na Tabela 1, justifica-se pela necessidade de expor o agente a paradigmas táticos extremos e bem definidos.

Enquanto a tabela anterior visou demonstrar a vasta gama de comportamentos que a modelagem genética permite criar, a seleção dos três oponentes fixos para o ambiente de treino focou em cobrir os eixos fundamentais do jogo: o combate direto (ataque), a preservação tática (segurança) e a velocidade de deslocamento (progresso). Ao confrontar simultaneamente essas três vertentes polares, o Algoritmo Genético é induzido a evoluir indivíduos que desenvolvam uma estratégia robusta e versátil, capaz de performar com eficácia contra as táticas mais distintas e desafiadoras que podem emergir no tabuleiro.

O *fitness* final é composto pelos seguintes componentes:

1. Colocação no *ranking* final

Ao final de cada partida, o agente recebe uma pontuação base de acordo com sua posição no *ranking*: 1º lugar: 50 pontos; 2º lugar: 30 pontos; 3º lugar: 10 pontos; 4º lugar: 0 pontos

Esse valor é multiplicado pelo gene de progresso, reforçando a importância do desempenho global.

2. Captura de peças adversárias

Cada vez que o agente captura uma peça adversária, recebe uma pontuação positiva (+2.0), ponderada pelo gene de ataque. Esse critério incentiva estratégias ofensivas e o controle do tabuleiro.

3. Perda de peças (Penalização)

Sempre que uma peça do agente é capturada por um adversário, é aplicada uma penalização ao *fitness* (-3.0). Essa penalização é ponderada pelo gene de segurança, incentivando estratégias mais defensivas quando esse gene assume valores elevados.

4. Finalização de peças

Cada peça que atinge a área final do tabuleiro gera uma recompensa adicional (+5.0), ponderada pelo gene de finalização, reforçando comportamentos orientados à conclusão do jogo.

5. Jogadas de risco

Movimentos que deixam uma peça vulnerável à captura nas próximas jogadas são penalizados (-1.0). Essa penalização é modulada pelo gene de risco, permitindo que o agente aprenda diferentes níveis de tolerância ao risco.

O *fitness* final de uma partida é calculado como a soma ponderada de todos esses componentes, e o *fitness* do indivíduo corresponde à média dos valores obtidos ao longo das partidas simuladas.

A arquitetura de decisão do agente baseia-se em uma função de utilidade ponderada, onde cada indivíduo da população é composto por um conjunto de genes que atuam como multiplicadores de peso (moduladores de prioridade) para as heurísticas de decisão, permitindo que o agente avalie cada jogada legal sob uma perspectiva única.

Durante uma partida, para cada movimento possível, o agente calcula um valor de utilidade multiplicando o valor base da ação (como a distância avançada ou o bônus de captura) pelo peso correspondente derivado de seu DNA. Por exemplo, um indivíduo com gene de ataque elevado atribuirá uma utilidade superior a movimentos que resultem em capturas, mesmo que isso implique em maior exposição ao risco. O agente seleciona a jogada que resulta no maior valor somado, permitindo que o processo evolutivo ajuste, ao longo das gerações, quais comportamentos devem ser priorizados para maximizar o desempenho e a sobrevivência no tabuleiro, conforme detalhado no Algoritmo 1.

Algoritmo 1: Cálculo de *fitness* e lógica de decisão do agente

Entrada: DNA do indivíduo $[G_{prog}, G_{ataq}, G_{seg}, G_{risco}, G_{cheg}]$

Saída : Aptidão final (*fitness*) do indivíduo

```

/* Conversão dos genes em pesos multiplicadores de comportamento
*/
1 foreach gene  $G_i \in DNA$  do
    /* Ajuste de escala conforme a lógica de 10% do código Python
    */
2      $W_i \leftarrow \max(0.1, 1.0 + (G_i \times 0.1))$ 
3 end

/* Ciclo de avaliação estatística (5 partidas por indivíduo) */
4 for partida  $\leftarrow 1$  to 5 do
5     Inicializar ambiente de jogo contra oponentes fixos
6     while partida não finalizada do
7         /* Tomada de decisão baseada na Heurística Ponderada */
8         foreach movimento  $m$  legal do
9              $U(m) \leftarrow$ 
10              $(Dist_m \cdot W_{prog}) + (Cap_m \cdot W_{ataq}) + (Seg_m \cdot W_{seg}) - (Risco_m \cdot W_{risco})$ 
11         end
12         Executar movimento  $m$  com maior utilidade  $U(m)$ 
13     end

    /* Cálculo do Fitness da Partida utilizando constantes do
    sistema */
14      $Pts_{Rank} \leftarrow BaseRank(Posição) \times W_{prog}$  // Rankings: 50.0, 30.0 ou 10.0
15      $Pts_{Cap} \leftarrow Total_{Capturas} \times (2.0 \times W_{ataq})$ 
16      $Pts_{Morte} \leftarrow Total_{Mortes} \times (-3.0 \times W_{seg})$ 
17      $Pts_{Final} \leftarrow Total_{Finalizações} \times (5.0 \times W_{cheg})$ 
18      $Pts_{Risco} \leftarrow Turnos_{Exposição} \times (-1.0 \times W_{risco})$ 
19      $F_{partida} \leftarrow Pts_{Rank} + Pts_{Cap} + Pts_{Morte} + Pts_{Final} + Pts_{Risco}$ 
20 end

19 AptidãoFinal  $\leftarrow$  Média( $F_{partidas}$ )
20 return AptidãoFinal

```

3.4 Métodos de seleção

O método de seleção define quais indivíduos serão escolhidos para reprodução, e neste trabalho foram implementados os seguintes métodos:

3.4.1 Seleção por torneio de N

Neste trabalho, um método utilizado foi a seleção por torneio de N . Neste procedimento, um subconjunto aleatório de N indivíduos é selecionado da população, e aquele com melhor *fitness* é escolhido como um progenitor. Esse processo é repetido para outro grupo aleatório de N indivíduos, tendo-se escolhido o segundo progenitor (MILLER; GOLDBERG et al., 1995). O valor de N foi definido empiricamente durante o processo de implementação e testes do algoritmo.

3.4.2 Seleção por roleta

Como método alternativo, também foi considerada a seleção por roleta, na qual a probabilidade de seleção de um indivíduo é proporcional ao seu valor de *fitness*. Esse método favorece indivíduos mais aptos, mas mantém a possibilidade de seleção de soluções subótimas, contribuindo para a diversidade (GOLDBERG, 1989).

3.4.3 Elitismo

O elitismo foi aplicado preservando uma taxa de porcentagem com os $X\%$ melhores indivíduos de cada geração, sendo X um valor definido empiricamente durante o processo de implementação e testes do algoritmo. Neste método, estes indivíduos são copiados diretamente para a próxima geração sem sofrer *crossover* ou mutação. Essa estratégia garante que soluções de alta qualidade não sejam perdidas (JONG, 1975).

3.5 Operadores de cruzamento

O operador de cruzamento (*crossover*) tem papel central na recombinação das características dos indivíduos mais aptos, sendo responsável por gerar novas combinações de estratégias a partir da população atual. Considerando que os genes do cromossomo possuem significados estratégicos distintos, foram testadas diferentes abordagens de cruzamento, cada uma com objetivos específicos de exploração e preservação de informação genética. Nesta implementação, o cruzamento ocorre em 90% das vezes após a seleção dos indivíduos pais. Este valor foi definido empiricamente durante o processo de implementação e testes do algoritmo.

3.5.1 Cruzamento uniforme padrão

O cruzamento uniforme, proposto por Holland (1975), é a abordagem mais tradicional utilizada neste trabalho. Para cada gene do cromossomo, é realizado um sorteio com probabilidade de 50%, definindo se o valor será herdado do Pai A ou do Pai B, de forma totalmente independente dos demais genes.

O principal diferencial desse método é a máxima diversidade genética gerada, sendo particularmente eficaz nas fases iniciais da evolução, quando a exploração global do espaço de busca é desejável.

Para detalhar o mecanismo de sorteio independente que rege a recombinação equitativa dos genes, o Algoritmo 2 apresenta o pseudocódigo desta abordagem.

Algoritmo 2: Cruzamento uniforme padrão

Entrada: Pai_A, Pai_B (vetores de 5 genes reais)

Saída : $Filho$ resultante do cruzamento

```

1  $Filho \leftarrow$  Inicializar cromossomo vazio com 5 posições
2 for  $i \leftarrow 1$  to 5 do
3    $Sorteio \leftarrow random(0, 1)$ 
4   if  $Sorteio < 0.50$  then
5      $Filho[i] \leftarrow Pai_A[i]$ 
6   end
7   else
8      $Filho[i] \leftarrow Pai_B[i]$ 
9   end
10 end
11 return  $Filho$ 

```

3.5.2 Cruzamento uniforme ponderado

Esta variação do cruzamento uniforme introduz um viés probabilístico baseado na Proporção Áurea (61,8%). Diferente do cruzamento uniforme convencional, onde cada gene tem 50% de chance de vir de qualquer progenitor, este método fundamenta-se no conceito de cruzamento uniforme parametrizado explorado por [Spears e Jong \(1991\)](#).

A lógica deste operador é a priorização da aptidão individual de forma equilibrada. No momento da recombinação, o algoritmo define, de forma aleatória, um dos progenitores como “dominante”. Como ambos os pais já foram selecionados previamente por meio do método de torneio, garantindo que possuam alta aptidão, a escolha aleatória do dominante serve para manter a diversidade genética.

Ao definir um dos pais como dominante, o algoritmo assume que a “receita” daquele indivíduo é bem-sucedida e deve ser mantida como base, permitindo que apenas 38,2% do código genético seja alterado pelo segundo pai, visando manter a diversidade. O uso da razão áurea busca um equilíbrio funcional que mimetiza padrões de eficiência encontrados na natureza, promovendo uma exploração intensiva das melhores soluções encontradas até o momento, sem descaracterizar as estruturas comportamentais que provaram ser eficazes.

A fim de explicitar a lógica de definição do progenitor dominante e a aplicação matemática da proporção áurea na herança genética, o fluxo de execução deste método é formalizado no Algoritmo 3.

Algoritmo 3: Cruzamento uniforme ponderado

Entrada: Pai_A, Pai_B (vetores de 5 genes reais)
Saída : *Filho* resultante do cruzamento

```

1 Filho  $\leftarrow$  Inicializar cromossomo vazio com 5 posições
2 Sorteio_Dominante  $\leftarrow$  random(0,1)
   /* Definição aleatória do progenitor dominante */
3 if Sorteio_Dominante < 0.50 then
4   | Pai_Dominante  $\leftarrow$   $Pai_A$ 
5   | Pai_Recessivo  $\leftarrow$   $Pai_B$ 
6 end
7 else
8   | Pai_Dominante  $\leftarrow$   $Pai_B$ 
9   | Pai_Recessivo  $\leftarrow$   $Pai_A$ 
10 end
11 for  $i \leftarrow 1$  to 5 do
12   | Sorteio_Heranca  $\leftarrow$  random(0,1)
      /* Probabilidade baseada na Proporção Áurea */
13   | if Sorteio_Heranca < 0.618 then
14     | Filho[ $i$ ]  $\leftarrow$  Pai_Dominante[ $i$ ]
15   | end
16   | else
17     | Filho[ $i$ ]  $\leftarrow$  Pai_Recessivo[ $i$ ]
18   | end
19 end
20 return Filho

```

3.5.3 Cruzamento entre blocos semânticos

Diferente do método anterior, que trata cada gene como uma unidade independente, o cruzamento entre blocos opera sobre grupos de genes com dependência funcional, baseando-se na hipótese dos blocos de construção de Goldberg. Em uma de suas pesquisas, Harik e Goldberg (1999) definem que genes com dependência funcional (*linkage*) devem ser transmitidos em conjunto para evitar a fragmentação de estratégias.

Neste trabalho, a diferença fundamental é a preservação de comportamentos. Enquanto no uniforme ponderado o descendente pode herdar o peso de ataque de um pai e

o peso de risco de outro, o cruzamento por blocos garante que todo o conjunto venha da mesma fonte.

O agrupamento de genes em blocos foi feito conforme a seguir:

- Movimentação: genes de progresso e finalização.
- Combate: genes de ataque, segurança e risco.

A decisão de herança ocorre por bloco: se o bloco de “combate” é selecionado, todos os seus componentes (ataque, segurança e risco) são herdados integralmente de um único pai. Essa técnica visa preservar estilos de jogo consistentes (agressivos ou defensivos), evitando o surgimento de descendentes funcionalmente desconexos.

Para ilustrar como a transferência de características ocorre de forma conjunta, preservando a dependência funcional entre as heurísticas de movimentação e combate, a estrutura computacional deste operador está descrita no Algoritmo 4.

Algoritmo 4: Cruzamento entre blocos semânticos

Entrada: Pai_A, Pai_B (vetores de 5 genes reais)

Saída : $Filho$ resultante do cruzamento

```

1  $Filho \leftarrow$  Inicializar cromossomo vazio
  /* Herança do Bloco de Movimentação (Progresso, Finalização) */
2  $Sorteio\_Movimentacao \leftarrow random(0, 1)$ 
3 if  $Sorteio\_Movimentacao < 0.50$  then
4   |  $Filho[Progresso, Finalizacao] \leftarrow Pai_A[Progresso, Finalizacao]$ 
5 end
6 else
7   |  $Filho[Progresso, Finalizacao] \leftarrow Pai_B[Progresso, Finalizacao]$ 
8 end

  /* Herança do Bloco de Combate (Ataque, Segurança, Risco) */
9  $Sorteio\_Combate \leftarrow random(0, 1)$ 
10 if  $Sorteio\_Combate < 0.50$  then
11   |  $Filho[Ataque, Seguranca, Risco] \leftarrow Pai_A[Ataque, Seguranca, Risco]$ 
12 end
13 else
14   |  $Filho[Ataque, Seguranca, Risco] \leftarrow Pai_B[Ataque, Seguranca, Risco]$ 
15 end

16 return  $Filho$ 
```

3.6 Operações de mutação

A mutação é responsável por introduzir diversidade genética e permitir a exploração de novas regiões do espaço de busca. Considerando a natureza contínua dos genes, foram avaliados diferentes métodos de mutação baseados em distribuições gaussianas e variações adaptativas de intensidade.

3.6.1 Mutação com alfa aleatório probabilístico

A mutação gaussiana tradicional altera um gene somando-lhe um valor aleatório. Neste método de alfa aleatório probabilístico, foi introduzido o fator $\alpha \in [0, 1]$ como um regulador de impacto. Baseado na *Non-Uniform Mutation* de [Michalewicz \(1996\)](#), o α funciona como uma “válvula de segurança”: se o sorteio de α for baixo, a mutação será um ajuste fino; se for alto, será uma mudança drástica.

Primeiro, verifica-se se o gene será mutado com base na taxa de mutação global. Caso positivo, é sorteado um fator $\alpha \in [0, 1]$, que escala o valor gerado por uma distribuição gaussiana.

Esse mecanismo permite que o algoritmo ganhe a capacidade de realizar microajustes de precisão na maioria das vezes, reservando grandes saltos evolutivos para casos raros. Isso torna o processo de busca mais estável e evita que o agente perca subitamente um comportamento bom que demorou gerações para aprender.

A demonstração exata de como o fator regulador α atua em conjunto com a distribuição gaussiana, estritamente condicionado à taxa de mutação global do indivíduo, pode ser observada no pseudocódigo do Algoritmo 5.

Algoritmo 5: Mutação com alfa aleatório probabilístico

Entrada: *Cromossomo*, Taxa de Mutação $T_m = 0.05$

Saída : *Cromossomo* mutado

```

1  for  $i \leftarrow 1$  to 5 do
2      Sorteio_Mutacao  $\leftarrow$  random(0, 1)
3      if Sorteio_Mutacao  $< T_m$  then
4           $\alpha \leftarrow$  random(0, 1)
5          Ruido  $\leftarrow$  gaussiana( $\mu = 0, \sigma = 0.5$ )
6          Perturbacao  $\leftarrow$  Ruido  $\times \alpha$ 
7          Cromossomo[ $i$ ]  $\leftarrow$  Cromossomo[ $i$ ] + Perturbacao
8          Limitar Cromossomo[ $i$ ] no intervalo  $[-5.0, 5.0]$ 
9      end
10 end
11 return Cromossomo

```

3.6.2 Mutação com perturbação alfa aleatória

Ao contrário do método anterior, esta abordagem aplica o fator α de forma direta e frequente, sem filtros probabilísticos adicionais. Esta técnica reflete os princípios das Estratégias Evolutivas de Schwefel (1981), onde a mutação é o motor principal de descoberta.

A diferença lógica com a mutação anterior reside na agressividade da exploração. Enquanto a mutação probabilística é cautelosa e “teme” estragar o que já está bom, a perturbação direta força variações constantes em todos os genes. O objetivo neste caso não é o ajuste fino, mas garantir que a população nunca pare de se movimentar no espaço de busca. Segundo Fogel (2006), este método evita que o algoritmo fique preso em uma estratégia mediana, forçando o agente a testar constantemente variações laterais de seus pesos.

Com o intuito de transparecer a aplicação direta e ininterrupta desta perturbação estocástica sobre o espaço de busca, a mecânica deste operador contínuo é detalhada no Algoritmo 6.

Algoritmo 6: Mutação com perturbação alfa aleatória

Entrada: *Cromossomo*

Saída : *Cromossomo* mutado

```

/* Aplicada de forma direta sem filtro probabilístico */
1 for i ← 1 to 5 do
2    $\alpha \leftarrow \text{random}(0, 1)$ 
3    $Ruido \leftarrow \text{gaussiana}(\mu = 0, \sigma = 0.5)$ 
4    $Perturbacao \leftarrow Ruido \times \alpha$ 
5    $Cromossomo[i] \leftarrow Cromossomo[i] + Perturbacao$ 
6   Limitar  $Cromossomo[i]$  no intervalo  $[-5.0, 5.0]$ 
7 end
8 return Cromossomo

```

3.6.3 Mutação com sigma aleatório

Neste método, a intensidade da mutação não é fixa. O desvio padrão (σ) da distribuição gaussiana é sorteado individualmente para cada gene selecionado, dentro do intervalo $[0.0, 0.5]$. A escolha deste limite superior foi definida considerando que a amplitude total dos genes no sistema é de 10.0 (variando de -5.0 a 5.0); assim, um σ máximo de 0.5 permite que, em um único evento de mutação, o gene possa saltar até 5% da escala total em um desvio padrão.

A lógica deste operador é a heterogeneidade adaptativa. Segundo Bäck e Schwefel

(1993), em sistemas onde não se conhece previamente a sensibilidade de cada gene, a mutação com parâmetros aleatórios permite que o próprio processo evolutivo encontre diferentes magnitudes de ajuste. Enquanto um gene pode sofrer um microajuste quase imperceptível, outro, na mesma geração, pode ser drasticamente alterado.

Para clarificar como o algoritmo sorteia e aplica diferentes intensidades de desvio padrão (σ) simultânea e independentemente em um mesmo cromossomo, o processo computacional encontra-se esquematizado no Algoritmo 7.

Algoritmo 7: Mutação com sigma aleatório

Entrada: *Cromossomo*, Taxa de Mutação $T_m = 0.05$

Saída : *Cromossomo* mutado

```

1 for  $i \leftarrow 1$  to 5 do
2    $Sorteio\_Mutacao \leftarrow random(0, 1)$ 
3   if  $Sorteio\_Mutacao < T_m$  then
4      $\sigma_{dinamico} \leftarrow random(0.0, 0.5)$ 
5      $Perturbacao \leftarrow gaussiana(\mu = 0, \sigma = \sigma_{dinamico})$ 
6      $Cromossomo[i] \leftarrow Cromossomo[i] + Perturbacao$ 
7     Limitar  $Cromossomo[i]$  no intervalo  $[-5.0, 5.0]$ 
8   end
9 end
10 return Cromossomo

```

3.6.4 Mutação com sigma baseada em fase

Este método adota uma abordagem temporal e determinística, ajustando o valor de σ conforme o progresso das gerações. O processo é dividido em três estágios estratégicos:

- Início (0–30%): $\alpha = 0.2$, focado em estabilizar a população inicial gerada aleatoriamente
- Meio (30–70%): $\alpha = 1.0$, aumenta-se o passo para saltar de máximos locais e explorar novas regiões do espaço de busca
- Final (70–100%): $\alpha = 0.1$, reduz-se a intensidade para lapidar as melhores soluções e garantir a convergência

A diferenciação lógica é o controle do balanço entre exploração e exploração. Esta estratégia fundamenta-se na técnica de *Step Size Adaptation* descrita por Eiben e Smith (2003), que sugere que a mutação deve ser ampla no início (para descobrir caminhos de busca) e pequena no final (para não perder os melhores indivíduos). Os valores foram definidos com base na literatura de Estratégias Evolutivas (RECHENBERG, 1973), que

demonstra que o sucesso de um algoritmo depende de uma transição suave entre a busca global e a otimização local, mimetizando um processo de aprendizado que começa cauteloso, torna-se ousado para aprender o novo e termina refinando o conhecimento adquirido.

A modelagem lógica da transição temporal, que ajusta deterministicamente a magnitude da mutação conforme o avanço percentual das gerações, encontra-se estruturada no Algoritmo 8.

Algoritmo 8: Mutação com sigma baseada em fases

Entrada: *Cromossomo*, *Geracao_Atual*, *Total_Geracoes*, $T_m = 0.05$

Saída : *Cromossomo* mutado

```

1  Progresso  $\leftarrow$  Geracao_Atual/Total_Geracoes
   /* Definição temporal do Sigma */
2  if Progresso < 0.30 then
3      |  $\sigma \leftarrow 0.2$                                 /* Início: Estabilização */
4  end
5  else if Progresso < 0.70 then
6      |  $\sigma \leftarrow 1.0$                                 /* Meio: Exploração profunda */
7  end
8  else
9      |  $\sigma \leftarrow 0.1$                                 /* Final: Refinamento e lapidação */
10 end

   /* Aplicação da Mutação */
11 for i  $\leftarrow$  1 to 5 do
12     Sorteio_Mutacao  $\leftarrow$  random(0, 1)
13     if Sorteio_Mutacao <  $T_m$  then
14         | Perturbacao  $\leftarrow$  gaussian( $\mu = 0, \sigma = \sigma$ )
15         | Cromossomo[i]  $\leftarrow$  Cromossomo[i] + Perturbacao
16         | Limitar Cromossomo[i] no intervalo [-5.0, 5.0]
17     end
18 end
19 return Cromossomo

```

3.7 Ciclo evolutivo

No AG elaborado para este trabalho, o ciclo do algoritmo é executado por N gerações e repetido em 10 execuções independentes, para garantir a validade estatística dos resultados, em que N é um valor definido pelo usuário. Um possível fluxo lógico do AG está detalhado no pseudocódigo a seguir:

Algoritmo 9: Ciclo evolutivo

Entrada: Tamanho da população $N = 100$, Número de gerações $G = 200$

Saída : Melhor indivíduo encontrado

```

1 Inicializar população com 100 indivíduos aleatórios
2 for geração  $\leftarrow 1$  to 200 do
    /* Ajuste dinâmico da taxa de mutação */
3     Definir  $\sigma$  de mutação conforme o progresso da geração
4     foreach indivíduo  $\in$  População do
5         | Avaliar fitness (conforme Algoritmo 1)
6     end
7     Ordenar população decrescentemente por fitness
8     Nova_população  $\leftarrow$  Selecionar 15 melhores indivíduos (Elitismo)
9     while tamanho(Nova_população)  $< 100$  do
10        |  $Pai_1, Pai_2 \leftarrow$  Seleção por torneio ( $k = 5$ )
11        | if random()  $< 0.9$  then
12            |  $Filho_1, Filho_2 \leftarrow$  Cruzamento_uniforme( $Pai_1, Pai_2$ )
13        | end
14        | else
15            |  $Filho_1, Filho_2 \leftarrow$  Clonar( $Pai_1, Pai_2$ )
16        | end
17        | Aplicar mutação_gaussiana( $Filho_1, Filho_2, \sigma$ )
18        | Adicionar Filhos à nova_população
19    end
20    População  $\leftarrow$  Nova_população
21 end
22 return Melhor indivíduo de todas as gerações

```

O fluxo operacional do AG implementado neste trabalho segue um ciclo iterativo de evolução e seleção, e o pseudocódigo apresentado ilustra uma possibilidade de execução do mesmo. Nesta execução, foi utilizada uma população de 100 indivíduos ao longo de 200 gerações. O processo é descrito a seguir:

- **Inicialização:** O ciclo inicia-se com a criação de uma população inicial composta por 100 vetores de pesos (genes), preenchidos de forma aleatória dentro do intervalo contínuo de -5.0 a 5.0 .
- **Ajuste paramétrico de mutação:** No início de cada geração, o algoritmo define o valor do desvio padrão (σ) para a mutação gaussiana. Este parâmetro é ajustado dinamicamente conforme o progresso das gerações: $\sigma = 0.2$ até atingir 30% do ciclo;

$\sigma = 1.0$ entre 30% e 70%; e $\sigma = 0.1$ para o restante das gerações.

- **Avaliação de aptidão (*Fitness*):** Cada indivíduo da população é submetido ao cálculo de aptidão, onde atua como o agente principal em 5 partidas simuladas contra oponentes fixos. O valor de *fitness* é atribuído com base na média do desempenho estatístico nessas partidas.
- **Elitismo:** Após a avaliação, a população é ordenada de forma decrescente. Os 15 melhores indivíduos (15% da população) são selecionados e copiados integralmente para a composição da próxima geração, garantindo a preservação das melhores soluções encontradas até o momento.
- **Seleção:** Para o preenchimento das 85 vagas restantes na nova população, utiliza-se a seleção por torneio com tamanho $k = 5$. Neste processo, cinco indivíduos são sorteados aleatoriamente, e aquele com maior aptidão é escolhido como progenitor. O procedimento é repetido para selecionar o segundo progenitor.
- **Recombinação (*Crossover*):** Com uma taxa de ocorrência de 90% (0.9), aplica-se o cruzamento uniforme entre os pares de pais selecionados, gerando dois descendentes que herdam genes de ambos os progenitores. Caso o sorteio seja superior à taxa de cruzamento, os descendentes são clones idênticos aos pais.
- **Mutação:** A mutação ocorre apenas mediante validação probabilística. O mecanismo opera gene a gene: primeiramente, verifica-se a ocorrência do evento (com 5% de chance); somente nos casos positivos é que o valor do gene sofre a adição do ruído gaussiano (sigma). Nos 95% restantes, o processo de alteração é ignorado e o gene preserva seu valor original.
- **Finalização:** O ciclo se repete até que a 200ª geração seja processada. Ao término, o algoritmo retorna o indivíduo que apresentou o maior valor de *fitness* histórico entre todas as execuções e a população final.

3.8 Considerações finais

Neste capítulo, foram apresentados os detalhes da implementação do Algoritmo Genético, com ênfase nos diferentes operadores genéticos desenvolvidos (seleção, cruzamento e mutação). A diversidade de métodos testados permitiu investigar como diferentes estratégias de recombinação e perturbação genética impactam a estabilidade, a convergência e a qualidade das soluções.

Essa base metodológica fundamenta os experimentos apresentados no Capítulo 4, nos quais os operadores e parâmetros descritos foram comparados a partir dos valores de *fitness* obtidos nas soluções geradas, bem como dos comportamentos observados nos

agentes campeões, permitindo identificar a configuração mais adequada para o problema abordado.

4 Resultados

Este capítulo apresenta os resultados obtidos a partir dos experimentos realizados com o Algoritmo Genético aplicado ao jogo de tabuleiro Ludo, desenvolvido neste Trabalho de Conclusão de Curso. A análise dos resultados teve ênfase na comparação dos valores de *fitness* em diferentes configurações do AG, e também na interpretação estratégica dos comportamentos observados nos agentes evoluídos. O objetivo foi analisar o que os resultados revelam sobre o processo evolutivo, de que forma os parâmetros influenciaram o comportamento do agente e em que medida o agente campeão demonstrou superioridade estratégica no ambiente de jogo.

Todos os experimentos são executados em 10 repetições independentes por configuração, permitindo a análise do melhor resultado, média dos melhores *fitness* e desvio padrão dos melhores. Essa abordagem visou reduzir o impacto da aleatoriedade inerente ao jogo e ao próprio AG, garantindo maior robustez às conclusões apresentadas. Todos os testes foram realizados em um computador *Dell Inspiron 15 3520*, equipado com processador *Intel® Core™ i5-1235U* de 12^a geração (1.30 GHz), 16 GB de memória RAM e sistema operacional *Windows 64 bits*.

Em relação ao tempo de execução, cada teste de parâmetro demandou aproximadamente 1 hora e 30 minutos para ser concluído, considerando as 10 repetições independentes adotadas para reduzir o impacto da aleatoriedade do algoritmo genético e do próprio ambiente de jogo.

A seguir, é descrita cada etapa de experimentação do Algoritmo Genético implementado neste trabalho.

4.1 Experimentos com o tamanho da população

O primeiro conjunto de experimentos teve como meta avaliar o impacto do tamanho da população inicial no desempenho do AG aplicado ao jogo Ludo. Foram testados cinco valores distintos de população: 30 (Tabela 2), 50 (Tabela 3), 100 (Tabela 4), 150 (Tabela 5) e 200 (Tabela 6) indivíduos, mantendo fixo o número de gerações em 200. Para este teste, o método de seleção utilizado foi o torneio com $k = 5$, com cruzamento uniforme sob uma taxa fixa de 90% (0.9) e mutação gaussiana com sigma baseada em fases operando sob uma taxa fixa de 5% (0.05).

Tabela 2 – Resultados das execuções e síntese dos dados (População = 30).

Execução	Melhor <i>fitness</i>
1	121.55
2	122.54
3	120.59
4	117.95
5	120.21
6	120.54
7	120.21
8	117.06
9	117.99
10	117.96
Desvio Padrão: 1.76	
Média de <i>fitness</i> : 119.66	
Melhor DNA (2): [4.50, 5.00, 0.84, 0.19, 5.00]	

Tabela 3 – Resultados das execuções e síntese dos dados (População = 50).

Execução	Melhor <i>fitness</i>
1	121.32
2	120.61
3	120.32
4	119.24
5	118.90
6	123.67
7	119.67
8	120.22
9	116.87
10	120.46
Desvio Padrão: 1.74	
Média de <i>fitness</i> : 120.12	
Melhor DNA (6): [4.83, 4.15, 0.84, 0.87, 4.97]	

Tabela 4 – Resultados das execuções e síntese dos dados (População = 100).

Execução	Melhor <i>fitness</i>
1	121.42
2	121.36
3	124.61
4	123.97
5	122.05
6	122.86
7	121.72
8	123.82
9	121.64
10	122.68
Desvio Padrão: 1.17	
Média de <i>fitness</i>: 122.61	
Melhor DNA (3): [5.0, 5.0, 1.62, 0.41, 4.88]	

Tabela 5 – Resultados das execuções e síntese dos dados (População = 150).

Execução	Melhor <i>fitness</i>
1	122.33
2	126.59
3	120.93
4	120.33
5	122.35
6	121.57
7	120.11
8	121.01
9	122.30
10	122.77
Desvio Padrão: 1.84	
Média de <i>fitness</i>: 122.02	
Melhor DNA (2): [5.0, 4.75, 1.26, 0.32, 5.00]	

Tabela 6 – Resultados das execuções e síntese dos dados (População = 200).

Execução	Melhor <i>fitness</i>
1	122.41
2	127.62
3	122.80
4	121.27
5	120.23
6	121.73
7	122.73
8	120.68
9	122.58
10	122.74
Desvio Padrão: 2.02	
Média de <i>fitness</i>: 122.47	
Melhor DNA (2): [4.98, 5.00, 0.65, 0.21, 4.61]	

4.1.1 Análise dos resultados

Observou-se que populações muito reduzidas apresentam maior variabilidade entre execuções, indicando sensibilidade elevada à inicialização aleatória. Por outro lado, populações maiores aumentam o custo computacional sem ganhos proporcionais de desempenho.

A configuração com população intermediária (tamanho igual a 100) apresentou o melhor equilíbrio entre desempenho médio elevado e baixo desvio padrão, sugerindo que o algoritmo conseguiu explorar adequadamente o espaço de soluções sem comprometer a estabilidade evolutiva.

4.1.1.1 Interpretação do DNA campeão

O DNA campeão [5.0, 5.0, 1.62, 0.41, 4.88] associado à configuração de população igual a 100 apresentou pesos elevados para progresso (5.0) e ataque (5.0), ambos com o valor máximo permitido. Isso indicou que, ao longo do processo evolutivo, o agente aprendeu que avançar peças de forma consistente e capturar adversários sempre que possível são estratégias altamente vantajosas.

Em contrapartida, os pesos associados à segurança (1.62) e ao risco (0.41) assumiram valores significativamente menores. Essa combinação revelou que o agente não evitou completamente situações perigosas, mas também não se expôs de forma excessiva. O peso alto da finalização (4.88), neste caso, sinalizou que o agente priorizou finalizar o percurso das peças. Em outras palavras, o comportamento emergente foi o de um agente agressivo, porém calculado, que aceitou riscos moderados quando houve potencial de ganho estratégico.

4.2 Experimentos com o tamanho das gerações

Após a análise do tamanho da população, foram realizados testes variando o número de gerações, mantendo a população fixa em 100 indivíduos. Foram avaliados os valores de 100 (Tabela 7), 150 (Tabela 5), 200 (Tabela 6) e 250 gerações (Tabela 10). Para estas avaliações, fixou-se a seleção por torneio ($k = 5$), o cruzamento uniforme sob uma taxa fixa de 90% (0.9) e a mutação gaussiana com sigma baseada em fases operando sob uma taxa fixa de 5% (0.05).

Tabela 7 – Resultados das execuções e síntese dos dados (Gerações = 100).

Execução	Melhor <i>fitness</i>
1	122.30
2	121.24
3	117.26
4	120.62
5	123.62
6	121.38
7	121.89
8	122.45
9	113.24
10	121.88
Desvio Padrão: 3.07	
Média de <i>fitness</i> : 120.58	
Melhor DNA (5): [4.96, 5.0, 1.50, 0.49, 4.96]	

Tabela 8 – Resultados das execuções e síntese dos dados (Gerações = 150).

Execução	Melhor <i>fitness</i>
1	121.96
2	122.01
3	122.86
4	121.52
5	120.87
6	124.29
7	124.31
8	120.55
9	120.98
10	122.33
Desvio Padrão: 1.32	
Média de <i>fitness</i> : 122.16	
Melhor DNA (7): [5.00, 5.00, 2.07, 0.04, 4.90]	

Tabela 9 – Resultados das execuções e síntese dos dados (Gerações = 200).

Execução	Melhor <i>fitness</i>
1	124.85
2	121.50
3	122.57
4	125.10
5	123.22
6	122.75
7	121.36
8	122.16
9	122.79
10	122.98
Desvio Padrão: 1.23	
Média de <i>fitness</i>: 122.92	
Melhor DNA (4): [4.97, 5.00, 0.69, 1.39, 5.00]	

Tabela 10 – Resultados das execuções e síntese dos dados (Gerações = 250).

Execução	Melhor <i>fitness</i>
1	125.72
2	124.16
3	122.63
4	121.94
5	120.79
6	122.69
7	122.18
8	122.68
9	120.92
10	121.38
Desvio Padrão: 1.50	
Média de <i>fitness</i>: 122.50	
Melhor DNA (1): [5.00, 5.00, 0.85, 0.82, 5.00]	

4.2.1 Análise dos resultados

A variação do número de gerações permitiu analisar o processo de convergência do AG. Observa-se que, para valores baixos de gerações, o algoritmo atuava em fase exploratória, não sendo capaz de estabilizar estratégias eficazes.

Em contrapartida, a partir de um determinado número de gerações, os ganhos de desempenho tornaram-se marginais, enquanto o desvio padrão tendeu a se manter estável ou até aumentar levemente. Isso sugere que o algoritmo atingiu um ponto de maturidade evolutiva com a população de tamanho igual a 200, no qual a exploração adicional do espaço de soluções não resulta em melhorias, analisando os valores de melhor *fitness*, média e desvio padrão.

4.2.1.1 Interpretação do DNA campeão

O DNA campeão obtido nessa fase [4.97, 5.00, 0.69, 1.39, 5.00] mantém o padrão observado anteriormente: pesos elevados para ataque e finalização, indicando que o agente aprendeu a não apenas capturar peças adversárias, mas também a priorizar a conclusão do percurso das próprias peças. Esse comportamento evidencia uma estratégia orientada à vitória global, e não apenas a ganhos locais durante a partida.

4.3 Experimentos com a taxa de elitismo

A fim de garantir a continuidade das características mais aptas da população, adotou-se o elitismo como estratégia de sobrevivência. A sensibilidade do algoritmo a este parâmetro foi testada sob quatro percentuais de preservação: 0.04 (Tabela 11), 0.10 (Tabela 12), 0.15 (Tabela 13) e 0.20 (Tabela 14), com parâmetros fixados em 100 para tamanho de população e 200 para número de gerações. O método de seleção foi o torneio ($k = 5$), com o cruzamento uniforme sob uma taxa fixa de 90% (0.9) e a mutação gaussiana com sigma baseada em fases operando sob uma taxa fixa de 5% (0.05).

Tabela 11 – Resultados das execuções e síntese dos dados (Elitismo = 0.04).

Execução	Melhor <i>fitness</i>
1	123.39
2	125.31
3	127.39
4	127.12
5	127.22
6	125.49
7	124.71
8	127.04
9	128.06
10	123.82
Desvio Padrão: 1.61	
Média de <i>fitness</i>: 125.65	
Melhor DNA (9): [5.00, 5.00, 0.97, 0.27, 5.00]	

Tabela 12 – Resultados das execuções e síntese dos dados (Elitismo = 0.10).

Execução	Melhor <i>fitness</i>
1	124.59
2	124.21
3	126.64
4	124.82
5	125.18
6	124.29
7	123.74
8	124.58
9	126.00
10	125.21
Desvio Padrão: 0.86	
Média de <i>fitness</i> : 124.92	
Melhor DNA (3): [4.94, 5.00, 0.62, 1.20, 4.75]	

Tabela 13 – Resultados das execuções e síntese dos dados (Elitismo = 0.15).

Execução	Melhor <i>fitness</i>
1	125.93
2	125.71
3	124.99
4	129.88
5	126.33
6	126.32
7	124.86
8	125.81
9	125.45
10	125.17
Desvio Padrão: 1.44	
Média de <i>fitness</i> : 126.04	
Melhor DNA (4): [5.00, 4.95, 2.08, 1.25, 5.00]	

Tabela 14 – Resultados das execuções e síntese dos dados (Elitismo = 0.20).

Execução	Melhor <i>fitness</i>
1	124.13
2	126.44
3	124.52
4	124.12
5	123.46
6	124.62
7	126.64
8	126.62
9	124.75
10	126.22
Desvio Padrão: 1.23	
Média de <i>fitness</i>: 124.75	
Melhor DNA (7): [5.00, 4.96, 1.28, 0.53, 4.85]	

4.3.1 Análise dos resultados

Os testes com diferentes taxas de elitismo demonstraram que valores muito baixos resultaram em perda frequente de boas soluções, enquanto valores mais altos reduziram a diversidade genética da população.

A taxa intermediária (0.15) apresentou maior estabilidade e melhor média de *fitness*, além de obter o melhor valor de *fitness* até o momento (129.88), indicando que a preservação controlada dos melhores indivíduos contribuiu para a consolidação de estratégias eficazes sem comprometer a capacidade exploratória do algoritmo.

4.3.1.1 Interpretação do DNA campeão

O DNA campeão nessa configuração [5.00, 4.95, 2.08, 1.25, 5.00] reforçou o padrão estratégico observado: forte valorização do ataque e da finalização, acompanhada de um controle moderado do risco. Isso sugere que o elitismo contribuiu para a consolidação de um estilo de jogo consistente, evitando oscilações bruscas entre execuções.

4.4 Experimentos com métodos de seleção

Outro aspecto avaliado foi o método de seleção dos indivíduos para reprodução. Foram comparados o método da roleta (Tabela 15) e o método de torneio (Tabelas ??, variando o tamanho do torneio, com parâmetros fixados em 100 para tamanho de população, 200 para número de gerações e 0.15 para taxa de elitismo. Nestes experimentos, foram utilizados o cruzamento uniforme sob uma taxa fixa de 90% (0.9) e a mutação gaussiana com sigma baseada em fases operando sob uma taxa fixa de 5% (0.05).

Tabela 15 – Resultados das execuções e síntese dos dados (Seleção: Roleta).

Execução	Melhor <i>fitness</i>
1	115.75
2	115.99
3	116.67
4	114.94
5	115.41
6	115.35
7	115.82
8	117.99
9	115.12
10	114.10
Desvio Padrão: 1.05	
Média de <i>fitness</i> : 115.71	
Melhor DNA (8): [5.00, 5.00, 1.12, 0.36, 4.99]	

Tabela 16 – Resultados das execuções e síntese dos dados (Seleção: Torneio de 2).

Execução	Melhor <i>fitness</i>
1	125.64
2	125.76
3	128.99
4	124.01
5	125.11
6	124.61
7	125.36
8	124.60
9	124.58
10	126.70
Desvio Padrão: 1.47	
Média de <i>fitness</i> : 125.23	
Melhor DNA (3): [4.99, 4.98, 1.45, 0.53, 4.80]	

Tabela 17 – Resultados das execuções e síntese dos dados (Seleção: Torneio de 3).

Execução	Melhor <i>fitness</i>
1	126.12
2	125.06
3	124.01
4	125.88
5	125.05
6	123.06
7	127.48
8	130.06
9	123.01
10	128.37
Desvio Padrão: 2.30	
Média de <i>fitness</i> : 125.11	
Melhor DNA (8): [5.00, 5.00, 0.77, 0.19, 4.13]	

Tabela 18 – Resultados das execuções e síntese dos dados (Seleção: Torneio de 5).

Execução	Melhor <i>fitness</i>
1	123.56
2	127.71
3	125.34
4	125.82
5	125.18
6	125.32
7	125.68
8	123.54
9	125.52
10	125.73
Desvio Padrão: 1.18	
Média de <i>fitness</i> : 125.34	
Melhor DNA (2): [5.00, 5.00, 1.03, 0.80, 5.00]	

Tabela 19 – Resultados das execuções e síntese dos dados (Seleção: Torneio de 7).

Execução	Melhor <i>fitness</i>
1	124.46
2	124.55
3	124.66
4	129.23
5	123.57
6	127.83
7	127.09
8	124.93
9	124.11
10	122.87
Desvio Padrão: 2.03	
Média de <i>fitness</i>: 125.33	
Melhor DNA (4): [5.00, 5.00, 0.84, 2.51, 5.00]	

4.4.1 Análise dos resultados

A comparação entre os métodos de seleção evidenciou que a seleção por torneio apresentou desempenho consideravelmente superior à roleta, focando no melhor *fitness* e média dos melhores. Note que na seleção por torneio de 3 e 7, foram obtidos os melhores valores de *fitness*, no entanto, o desvio padrão foi mais alto nestes testes. O controle da pressão seletiva proporcionado pelo torneio de 5 favoreceu a propagação de estratégias de jogo eficazes, apresentando o menor desvio padrão entre os testes com o torneio. Portanto, esta configuração foi escolhida como a mais adequada para este experimento.

4.4.1.1 Interpretação do DNA campeão

O DNA campeão [5.00, 5.00, 1.03, 0.80, 5.00] obtido por esse método apresenta pesos elevados para ataque, indicando que o agente aprendeu que capturar peças adversárias gera vantagens acumulativas significativas ao longo da partida. Simultaneamente, o peso reduzido associado ao risco demonstra que o agente evita jogadas excessivamente perigosas, privilegiando ataques com maior probabilidade de sucesso.

4.5 Experimentos com métodos de cruzamento

Além dos parâmetros estruturais do AG, também foram realizados experimentos com diferentes métodos de cruzamento sob uma taxa fixa de 90% (0.9): de bloco (Tabela 20), uniforme (Tabela 21) e uniforme ponderado (Tabela 22). Os parâmetros utilizados nestes testes foram 100 para tamanho da população, 200 para número de gerações, 0.15 para taxa de elitismo e seleção com torneio de 5. Nestes testes, foi utilizada a mutação gaussiana com sigma baseada em fases operando sob uma taxa fixa de 5% (0.05).

Tabela 20 – Resultados das execuções e síntese dos dados (Cruzamento: Blocos).

Execução	Melhor <i>fitness</i>
1	123.66
2	124.87
3	125.62
4	124.93
5	126.18
6	124.49
7	129.20
8	123.66
9	125.14
10	127.54
Desvio Padrão: 1.73	
Média de <i>fitness</i> : 125.52	
Melhor DNA (7): [4.82, 5.00, 2.12, 0.33, 4.86]	

Tabela 21 – Resultados das execuções e síntese dos dados (Cruzamento: Uniforme).

Execução	Melhor <i>fitness</i>
1	127.91
2	125.91
3	126.17
4	125.49
5	124.57
6	124.77
7	128.41
8	125.86
9	124.74
10	124.41
Desvio Padrão: 1.37	
Média de <i>fitness</i> : 125.82	
Melhor DNA (7): [5.00, 5.00, 0.96, 0.25, 5.00]	

Tabela 22 – Resultados das execuções e síntese dos dados (Cruzamento: Uniforme Ponderado).

Execução	Melhor <i>fitness</i>
1	114.47
2	117.80
3	117.27
4	118.05
5	117.11
6	117.83
7	114.34
8	114.72
9	116.64
10	116.05
Desvio Padrão: 1.45	
Média de <i>fitness</i>: 116.42	
Melhor DNA (4): [5.00, 5.00, 0.77, 0.58, 4.89]	

4.5.1 Análise dos resultados

Foram avaliados diferentes métodos de cruzamento com o objetivo de identificar qual abordagem seria mais adequada à natureza contínua e semântica dos genes utilizados neste trabalho. Cada gene do cromossomo representa um peso associado a uma heurística específica de decisão, de modo que alterações bruscas ou estruturalmente incoerentes poderiam comprometer estratégias eficazes já aprendidas.

Entre os métodos testados, o cruzamento uniforme apresentou maior média dos melhores (125.52) e menor desvio padrão (1.37) deste experimento, indicando desempenho superior aos demais e maior estabilidade entre as execuções. Esse resultado pode ser explicado pelo fato de que o cruzamento uniforme permite a recombinação independente de cada gene, preservando parcialmente a contribuição estratégica individual de cada heurística. Deste modo, estratégias bem-sucedidas não são destruídas abruptamente, mas gradualmente refinadas ao longo das gerações.

O cruzamento uniforme ponderado não obteve um desempenho satisfatório pois, ao introduzir um viés excessivo baseado na proporção áurea para um dos progenitores, o algoritmo tendeu a limitar a diversidade genética necessária para o ajuste fino. No contexto do Ludo, a recombinação equilibrada é vital para fundir táticas ofensivas e defensivas; o peso desproporcional do método ponderado acabou “viciando” a população em soluções estáveis mas sub-ótimas, impedindo a exploração de nichos de fitness mais elevados.

4.5.1.1 Interpretação do DNA campeão

No DNA campeão final [5.00, 5.00, 0.96, 0.25, 5.00], observou-se que valores extremos, como os pesos máximos atribuídos ao ataque, ao progresso e a finalização, coexistem

com pesos mais moderados associados à segurança e ao risco. Essa combinação sugeriu que o cruzamento uniforme permitiu ao algoritmo combinar agressividade elevada com controle seletivo de risco, sem que um aspecto anulasse o outro.

Deste modo, o cruzamento uniforme mostrou-se especialmente adequado para problemas em que o significado individual de cada gene é claro e independente, contribuindo para a formação de um DNA campeão coerente, equilibrado e estrategicamente interpretável.

O DNA campeão do cruzamento uniforme ponderado [5.00, 5.00, 0.77, 0.58, 4.89] demonstrou que, embora o algoritmo tenha identificado a importância da agressividade, ele não conseguiu refinar os genes de suporte de forma eficaz. O *fitness* médio de 116.42 evidencia que a estratégia resultante é menos resiliente do que a obtida pelo cruzamento uniforme, pois a herança enviesada dificultou a calibração precisa entre o risco e a segurança, resultando em um agente que, apesar de agressivo, é mais vulnerável a variações táticas dos oponentes.

4.6 Experimentos com métodos de mutação

Concluída a análise do cruzamento, esta seção foi dedicada ao operador de mutação, responsável por fornecer a inovação necessária ao processo evolutivo. Foram testadas diversas taxas e métodos de mutação (Tabelas ?? para identificar como a inserção de perturbações estocásticas influencia a fuga de ótimos locais e a evolução do *fitness*. Os parâmetros utilizados nestes testes foram 100 para tamanho da população, 200 para número de gerações, 0.15 para taxa de elitismo, seleção com torneio de 5 e cruzamento uniforme sob uma taxa fixa de 90% (0.9). A taxa de mutação aplicada em todos os métodos testados nesta seção foi de 5% (0.05).

Tabela 23 – Resultados das execuções e síntese dos dados (Mutações: Alfa Aleatório Probabilístico).

Execução	Melhor <i>fitness</i>
1	129.19
2	127.29
3	123.17
4	126.23
5	122.63
6	125.72
7	128.91
8	125.68
9	123.33
10	125.67
Desvio Padrão: 2.27	
Média de <i>fitness</i>: 125.78	
Melhor DNA (1): [4.98, 5.00, 0.96, 0.85, 4.68]	

Tabela 24 – Resultados das execuções e síntese dos dados (Mutações: Sigma Aleatório).

Execução	Melhor <i>fitness</i>
1	125.37
2	124.77
3	124.12
4	125.89
5	122.92
6	126.39
7	127.14
8	123.90
9	125.23
10	123.96
Desvio Padrão: 1.28	
Média de <i>fitness</i>: 124.96	
Melhor DNA (7): [4.95, 4.96, 1.03, 1.53, 4.59]	

Tabela 25 – Resultados das execuções e síntese dos dados (Mutações: Perturbação Alfa Aleatório).

Execução	Melhor <i>fitness</i>
1	124.15
2	124.94
3	123.93
4	124.34
5	125.24
6	124.39
7	125.52
8	123.65
9	123.36
10	124.54
Desvio Padrão: 1.86	
Média de <i>fitness</i> : 123.70	
Melhor DNA (7): [4.94, 4.73, 1.52, 0.36, 4.94]	

Tabela 26 – Resultados das execuções e síntese dos dados (Mutações: Sigma baseado em Fase).

Execução	Melhor <i>fitness</i>
1	127.80
2	130.23
3	126.78
4	127.31
5	126.84
6	127.77
7	126.61
8	126.34
9	128.45
10	127.59
Desvio Padrão: 1.13	
Média de <i>fitness</i> : 127.57	
Melhor DNA (2): [5.00, 5.00, 1.90, 0.69, 5.00]	

4.6.1 Análise dos resultados

Diversos métodos de mutação foram avaliados com o objetivo de equilibrar exploração e exploração ao longo do processo evolutivo. Métodos com intensidade fixa apresentaram limitações claras: mutações muito agressivas tendem a destruir boas soluções, enquanto mutações excessivamente suaves podem levar à estagnação precoce.

Nesse contexto, foi adotada uma estratégia baseada em sigma variável por fases evolutivas, com valores definidos como 0.2 na fase inicial, 1.0 na fase intermediária e 0.1 na

fase final. Essa escolha refletiu uma compreensão do comportamento estocástico inerente ao AG e do nível de aleatoriedade presente em cada etapa da evolução.

Na fase inicial, sabe-se que o AG apresenta um alto grau de diversidade devido à inicialização aleatória da população. Portanto, o uso de um valor de sigma reduzido (0.2) mostrou-se mais adequado, evitando que o excesso de mutação amplificasse ainda mais o caos inicial e dificultasse a identificação de padrões estratégicos promissores.

Na fase intermediária, o aumento do sigma para 1.0 permitiu uma exploração mais ampla do espaço de soluções. Nesse estágio, a população já possuía estruturas estratégicas básicas, e mutações mais intensas favorecendo à descoberta de combinações de pesos alternativas, possibilitando a superação de ótimos locais.

Por fim, na fase final, com a redução do sigma para 0.1 foi possível o refinamento fino dos pesos heurísticos. Nesse momento, o objetivo não era mais explorar novas regiões do espaço de soluções, mas ajustar com precisão estratégias já consolidadas, reduzindo variações desnecessárias e promovendo convergência estável.

Por outro lado, a mutação com alfa aleatório probabilístico não obteve um desempenho tão robusto quanto a mutação por fases devido à sua natureza puramente estocástica na definição da intensidade da perturbação (α). Ao sortear a força da mutação de maneira totalmente aleatória para cada gene em cada geração, o algoritmo perdeu o controle tático sobre a evolução; enquanto a mutação por fases adaptou o rigor da busca ao “tempo de vida” da população, o método probabilístico manteve um nível de ruído instável que dificultou o refinamento dos genes mais sensíveis.

4.6.1.1 Interpretação do DNA campeão

A estratégia de mutação adotada contribuiu diretamente para a formação do DNA campeão [5.00, 5.00, 1.90, 0.69, 5.00], especialmente no que diz respeito ao equilíbrio entre agressividade e controle de risco. A moderação da mutação na fase inicial permitiu que pesos elevados para ataque e progressão emergissem de forma consistente, sem serem constantemente perturbados por variações excessivas.

Durante a fase intermediária, mutações mais intensas favoreceram ajustes significativos nos pesos associados à segurança e ao risco, permitindo ao agente experimentar estratégias menos conservadoras sem comprometer totalmente sua eficiência. Esse processo foi essencial para a redução gradual do peso do risco, refletindo a aprendizagem de que jogadas excessivamente perigosas não maximizam o desempenho global.

Na fase final, a baixa intensidade de mutação foi determinante para estabilizar os valores quase máximos observados nos genes de ataque e finalização, ao mesmo tempo em que refinou o peso intermediário da segurança. Esse comportamento evidencia que o agente aprendeu a priorizar ações ofensivas e orientadas ao objetivo, mantendo cautela

suficiente para evitar perdas desnecessárias.

Dessa forma, a mutação por fases não apenas melhorou o desempenho médio do algoritmo, como também desempenhou papel fundamental na consolidação de um DNA campeão estável, interpretável e estrategicamente consistente.

4.7 Análise da configuração final

A escolha dos métodos que compõem a arquitetura deste AG, foi baseada, portanto, na análise detalhada realizada durante todas as etapas de experimentação relatadas anteriormente. Para fins comparativos, foi considerado o trabalho de [Schmiele \(2023\)](#) que focou na gestão de recursos em um ambiente de acúmulo financeiro (Monopoly), enquanto o presente trabalho priorizou a sobrevivência e o posicionamento tático. O comparativo entre ambas as abordagens está sintetizado na Tabela 27.

Tabela 27 – Tabela comparativa deste AG com o do trabalho de referência.

Critério	(SCHMIELE, 2023)	Este trabalho
Geração inicial	Vetores de parâmetros estratégicos com inicialização aleatória.	Vetores contínuos de pesos heurísticos com preenchimento aleatório no intervalo definido.
Método de seleção	Seleção por roleta.	Seleção por torneio.
Método de recombinação	<i>Crossover Radcliff</i> .	<i>Crossover</i> uniforme.
Método de mutação	Mutação gaussiana com parâmetros fixos.	Mutação gaussiana por gene com sigma variável por fases evolutivas.
Função de <i>fitness</i>	Avaliação baseada no desempenho econômico do agente ao longo das partidas.	Avaliação estatística baseada em ranking final, capturas, perdas, finalizações e exposição ao risco.

A seguir, detalham-se as justificativas táticas para a escolha dos métodos implementados nesta pesquisa:

- **Seleção por torneio:** Diferente da seleção por roleta utilizada por [Schmiele \(2023\)](#), que pode sofrer com a dominação prematura de indivíduos “super-aptos”, a seleção por torneio foi escolhida para o Ludo por manter uma pressão seletiva constante. Os experimentos demonstraram que, como o *fitness* no Ludo pode apresentar valores próximos ou negativos (devido às penalidades de morte), a roleta gerava instabilidade. O torneio provou ser mais robusto ao selecionar indivíduos baseando-se na superioridade relativa, o que é corroborado por [Goldberg e Deb \(1991\)](#) e [Miller, Goldberg et al. \(1995\)](#), que indicam que este método oferece uma pressão seletiva

mais controlável e eficiente em comparação à roleta e ao ranking, garantindo diversidade mesmo em gerações com alta competitividade.

- **Crossover uniforme:** Optou-se pelo *crossover* uniforme em detrimento de métodos aritméticos como o de Radcliff. No Ludo, os genes representam pesos heurísticos independentes (ex: ataque vs. segurança). Por meio dos experimentos, observou-se que a recombinação aritmética tendia a neutralizar os comportamentos, gerando agentes com pesos médios e sem personalidade definida. O *crossover* uniforme, amplamente discutido por Syswerda (1989), permitiu que características táticas puras (como uma agressividade alta) fossem herdadas integralmente. Como destacado por Eiben e Smith (2015), este método é particularmente eficaz quando não há uma ordem linear de importância entre os genes, possibilitando combinações mais eficazes de estratégias distintas.
- **Mutação com sigma variável por fases:** Enquanto o modelo de referência utiliza a substituição total de genes, este trabalho implementou a mutação gaussiana adaptativa. A justificativa para o uso de três fases (σ alto para exploração e σ baixo para refinamento) reside na necessidade de ajuste fino dos pesos, uma técnica de controle de parâmetros discutida por Eiben, Hinterding e Michalewicz (1999) e baseada nos princípios de Estratégias Evolutivas de Rechenberg (1973). Testes indicaram que mutações fixas ou substituições aleatórias foram agressivas demais nas gerações finais, destruindo agentes que já estavam próximos do comportamento ideal. O escalonamento do sigma permitiu que o algoritmo explorasse o mapa de *fitness* no início e estabilizasse as vitórias na fase de conclusão, técnica essencial para o balanço entre exploração e refinamento (HINTERDING; MICHALEWICZ; EIBEN, 1997).

A combinação das configurações selecionadas mostrou-se crucial para o melhor desempenho do AG em relação aos valores de *fitness*. Em conjunto, esses parâmetros permitiram ao algoritmo explorar o espaço de soluções de forma eficiente e, ao mesmo tempo, consolidar estratégias eficazes, resultando no DNA campeão final igual a: [5.00, 5.00, 1.90, 0.69, 5.00]

A análise desse vetor revela um comportamento estratégico claro e coerente:

- Progresso (5.00): o agente atribui máxima importância ao avanço constante das peças, evitando estagnações e priorizando movimentações que contribuem diretamente para a vitória.
- Ataque (5.00): capturar peças adversárias é percebido como extremamente vantajoso, tanto por reduzir o progresso dos oponentes quanto por gerar ganhos acumulativos ao longo da partida.

- Segurança (1.90): o agente considera a proteção das próprias peças, mas não de forma excessiva. Isso indica uma postura equilibrada, na qual a segurança é relevante, porém não impede ações ofensivas.
- Risco (0.69): o baixo peso associado ao risco demonstra que o agente aprendeu a evitar jogadas desnecessariamente perigosas, optando por ataques e avanços com maior probabilidade de sucesso.
- Finalização (5.00): há uma valorização quase máxima da conclusão das peças, evidenciando uma estratégia orientada ao objetivo final do jogo, e não apenas a ganhos intermediários.

Esse conjunto de pesos caracterizou um agente agressivo, objetivo e estrategicamente cauteloso, capaz de pressionar os adversários sem se expor excessivamente.

Análise de robustez

Com o objetivo de verificar se o DNA campeão final foi resultado de uma convergência consistente, e não de uma execução isolada favorecida pela aleatoriedade, foi realizada uma análise do desvio padrão dos DNAs campeões obtidos a partir das diferentes configurações selecionadas, na tabela 28.

Tabela 28 – Análise do agente campeão das melhores configurações nos experimentos.

Parâmetros Testados	Progresso	Ataque	Segurança	Risco	Finalização
População	5.00	5.00	1.62	0.41	4.88
Geração	4.97	5.00	0.69	1.39	5.00
Elitismo	5.00	4.95	2.08	1.25	5.00
Torneio	5.00	5.00	1.03	0.80	5.00
Cruzamento	5.00	5.00	0.96	0.25	5.00
Mutação	5.00	5.00	1.90	0.69	5.00
Média	4.995	4.991	1.38	0.78	4.98
Desvio	0.01	0.02	0.56	0.45	0.04

Os resultados indicam desvios padrão baixos para todos os genes, especialmente para o progresso, o ataque e a finalização, demonstrou que o algoritmo genético convergiu repetidamente para soluções que buscavam um agente mais agressivo e finalizador. Esse comportamento sugeriu que:

- O espaço de soluções apresentou uma região claramente ótima ou quase ótima;
- A função de *fitness* direcionou o processo evolutivo de forma consistente;

- As configurações escolhidas promoveram estabilidade e reprodutibilidade.

Em termos práticos, esse teste foi fundamental para validar que o comportamento estratégico aprendido pelo agente não dependeu de inicializações específicas nem de flutuações estocásticas, reforçando a confiabilidade e a generalização da solução encontrada. A baixa variabilidade observada entre os DNAs campeões indicou que o algoritmo genético convergiu repetidamente para uma mesma região do espaço de soluções, caracterizada por um equilíbrio consistente entre agressividade, segurança e orientação à finalização.

Esse resultado foi particularmente relevante no contexto de jogos estocásticos, nos quais pequenas variações iniciais podem levar a trajetórias evolutivas distintas. A convergência estrutural observada demonstrou que a função de *fitness*, aliada às configurações escolhidas, conduziu o processo evolutivo de forma estável, favorecendo estratégias efetivamente superiores.

4.8 Teste de validação do agente campeão

Finalmente, foi conduzido um experimento de validação do agente campeão, no qual este foi submetido a 100 partidas completas contra três agentes aleatórios, a fim de verificar se esse desempenho se traduziu em vantagem real no ambiente de jogo, conforme figura 5.

Os agentes aleatórios utilizados nesse experimento não possuíam qualquer estratégia fixa ou heurística de decisão. Em cada jogada, as ações possíveis foram selecionadas de forma puramente aleatória, sem considerar estado do jogo, riscos, oportunidades de ataque ou progressão estratégica. Deste modo, essa configuração estabeleceu um cenário de referência neutro, permitindo avaliar se o agente evoluído foi capaz de explorar de forma consistente a ausência de estratégia dos oponentes.

Todas as partidas foram realizadas sob as mesmas condições iniciais, garantindo igualdade de oportunidades e comparabilidade entre os resultados. As métricas analisadas incluíram taxa de vitória, número médio de capturas realizadas, número de vezes em que o agente foi capturado, quantidade de peças finalizadas, frequência de escolha por casas seguras e número de jogadas classificadas como arriscadas.

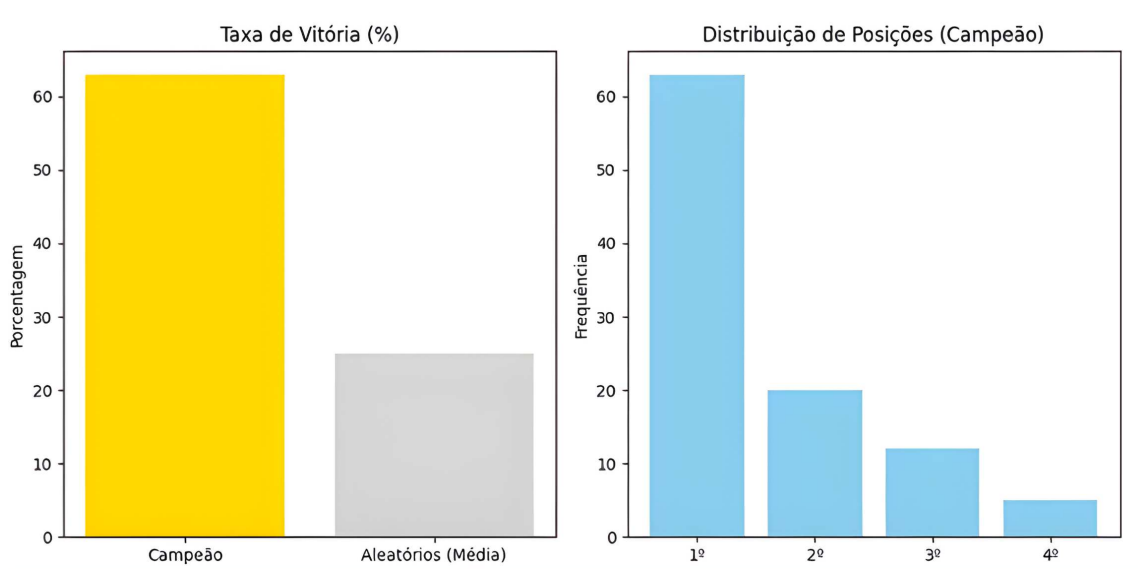


Figura 5 – Distribuição de vitórias (agente campeão vs aleatório).

Tabela 29 – Comparação de desempenho do agente campeão com um aleatório.

Métricas	Agente campeão	Média jogadores aleatórios
Taxa de vitória (%)	63.00%	12.33%
Capturas	9.41	5.31
Ser capturado	5.64	6.71
Finalização da peça	3.93	3.22
Optar por casas seguras	20.29	19.73
Jogadas arriscadas	12.52	22.03

A análise comparativa detalhada na Figura 5 evidenciou a superioridade tática do indivíduo evoluído. Enquanto os jogadores aleatórios apresentaram uma distribuição de métricas dispersa e ineficiente, o agente campeão demonstrou uma especialização clara: houve um aumento expressivo no volume de capturas ofensivas e uma redução drástica na exposição ao perigo. Este comportamento indica que o algoritmo genético conseguiu converter os pesos heurísticos do DNA em uma vantagem posicional real, onde a agressividade não foi aleatória, mas direcionada para momentos em que o risco de contra-ataque foi minimizado.

Os resultados demonstraram que o agente campeão apresentou taxa de vitória significativamente superior em relação aos agentes aleatórios (63.00% contra uma média de 12.33%), evidenciando que a estratégia aprendida foi efetivamente mais eficiente do que decisões tomadas ao acaso. Além disso, observou-se que o agente realizou um número médio maior de capturas (9.41), ao mesmo tempo em que foi capturado com menor frequência (5.64), indicando um equilíbrio eficaz entre comportamento ofensivo e preservação das próprias peças.

Outro aspecto relevante foi a menor incidência de jogadas arriscadas por parte do agente campeão (12.52 contra 22.03 dos aleatórios). Esse comportamento confirmou a interpretação do DNA final, especialmente o baixo peso associado ao risco, refletindo uma estratégia que priorizou ações com maior expectativa de retorno e menor probabilidade de perda. Ainda assim, o agente não se tornou excessivamente conservador, como evidenciado pela elevada taxa de finalização de peças, que se manteve próxima ao limite máximo de 4.0 por partida.

Esse conjunto de resultados reforçou que o alto valor de *fitness* correspondeu à aprendizagem de um comportamento estratégico consistente e eficaz no ambiente de jogo. Dessa forma, a validação confirmou que o agente evoluído apresenta desempenho superior em cenários não diretamente explorados durante o processo de evolução.

4.9 Considerações finais

A análise dos resultados apresentados neste capítulo permitiu afirmar que o AG foi capaz de evoluir agentes com comportamento estratégico coerente e robusto. A convergência consistente para DNAs estruturalmente semelhantes, aliada ao baixo desvio padrão observado entre execuções independentes, evidenciou a estabilidade do processo evolutivo e a adequação das configurações selecionadas.

Adicionalmente, a validação por meio de partidas contra agentes aleatórios demonstrou que a estratégia aprendida se traduz em vantagem prática no jogo de tabuleiro Ludo. O agente campeão apresentou um equilíbrio claro entre agressividade, controle de risco e foco na finalização, características diretamente refletidas nos pesos do DNA final.

Dessa forma, os resultados experimentais confirmaram que a configuração final do Algoritmo Genético, bem como a estratégia emergente do agente evoluído, foram apropriadas para o problema proposto.

5 Conclusão

Este Trabalho de Conclusão de Curso teve como objetivo investigar a aplicação de Algoritmos Genéticos na otimização dos parâmetros de decisão de um agente inteligente no jogo Ludo, um ambiente caracterizado por forte componente estocástico e interação entre múltiplos jogadores. Diferentemente de abordagens baseadas em regras fixas, buscou-se avaliar se técnicas evolutivas seriam capazes de adaptar automaticamente estratégias de jogo, resultando em comportamentos mais consistentes e eficazes ao longo de múltiplas partidas.

Para atingir esse objetivo, foi desenvolvido ambiente de simulação lógica do jogo Ludo e implementado um Algoritmo Genético no qual cada indivíduo representa uma estratégia de decisão codificada por um conjunto de pesos associados a heurísticas de jogo. A metodologia adotada permitiu analisar de forma sistemática o impacto de diferentes configurações evolutivas, incluindo tamanho da população, número de gerações, métodos de seleção, taxas de cruzamento, elitismo e estratégias de mutação, sempre considerando a variabilidade inerente ao ambiente de jogo.

Esta pesquisa foi motivada por estudos anteriores que aplicaram Algoritmos Genéticos em jogos de tabuleiro, em especial o trabalho desenvolvido no contexto do jogo Monopoly ([SCHMIELE, 2023](#)), que serviu como inspiração metodológica para a investigação apresentada. Embora os jogos possuam dinâmicas distintas, a adaptação da abordagem evolutiva para o Ludo demonstrou que a metodologia é flexível e capaz de lidar com diferentes níveis de aleatoriedade e interação estratégica, reforçando o potencial dos Algoritmos Genéticos como ferramenta de otimização em ambientes de decisão complexos.

As escolhas metodológicas, definidas empiricamente ao longo desta pesquisa, ratificaram princípios fundamentais da Computação Evolutiva, evidenciando a consistência entre os resultados obtidos e os pressupostos teóricos da área. A superioridade da seleção por torneio em relação à seleção por roleta corrobora a literatura, que aponta sua maior robustez frente a ruídos estatísticos e pequenas diferenças de aptidão, além de proporcionar controle mais estável da pressão seletiva ([BÄCK; FOGEL; MICHALEWICZ, 2002](#)). O cruzamento uniforme demonstrou superioridade na preservação e recombinação eficaz de componentes estratégicos relevantes, mantendo a integridade de características táticas específicas. Em contraste, operadores aritméticos evidenciaram tendência à suavização comportamental decorrente da combinação linear dos genitores, resultando na neutralização de estratégias mais marcantes e na redução da diversidade expressiva das soluções ([SYSWERDA, 1989](#); [MICHALEWICZ, 1996](#)). Por fim, a adoção de mutação com sigma variável ao longo das gerações evidenciou comportamento consistente com os princípios

de controle adaptativo descritos na literatura de *Evolution Strategies* (BEYER; SCHWEFEL, 2002), nos quais a redução progressiva favorece a transição da exploração global do espaço de busca para o refinamento local próximo à convergência. Essa dinâmica contribuiu para preservar indivíduos de alta aptidão nas fases finais, minimizando perturbações excessivas e reduzindo o risco de degradação de soluções já consolidadas.

Além disso, os resultados experimentais apresentados no Capítulo 4 demonstraram que o AG foi capaz de evoluir agentes com desempenho significativamente superior a agentes aleatórios, além de apresentar comportamento estratégico consistente e estável entre execuções independentes. A convergência recorrente para DNAs estruturalmente semelhantes, aliada aos baixos desvios padrão observados nas melhores configurações, indica que o processo evolutivo conseguiu explorar de forma eficiente o espaço de estratégias, sem perda excessiva de diversidade. A análise dos DNAs campeões revelou padrões estratégicos claros, como a valorização de ações ofensivas e de finalização, combinadas com controle moderado de risco. Esses padrões se refletiram diretamente no comportamento observado em jogo, evidenciando que os parâmetros evoluídos não apenas maximizam a função de *fitness*, mas também resultam em estratégias coerentes do ponto de vista do domínio do problema.

Como direções para trabalhos futuros, recomenda-se investigar operadores de cruzamento alternativos, explorar estratégias de mutação adaptativas mais sofisticadas e avaliar a integração de Algoritmos Genéticos com técnicas de Aprendizado por Reforço. Adicionalmente, a aplicação da metodologia a variantes mais complexas do jogo Ludo ou a outros jogos estocásticos de tabuleiro poderia ampliar a compreensão sobre a escalabilidade e a capacidade de generalização da abordagem proposta.

Em conclusão, a utilização de Algoritmos Genéticos para otimização de agentes inteligentes em ambientes estocásticos mostrou-se eficaz, produzindo resultados consistentes à literatura e estratégias emergentes interpretáveis. Este trabalho contribui para o entendimento prático da aplicação de técnicas evolutivas em jogos e reforça a importância de análises experimentais criteriosas no desenvolvimento de agentes inteligentes adaptativos.

Referências

- ANTONIO, N. d. S. **Otimização da função de avaliação de Dominó de 4 pontas utilizando Algoritmo genético**. Dissertação (Mestrado) — Universidade Federal do Amazonas, 2011. Citado 6 vezes nas páginas 14, 16, 17, 20, 21 e 22.
- BÄCK, T.; FOGEL, D. B.; MICHALEWICZ, Z. **Introduction to Evolutionary Computing**. Berlin, Heidelberg: Springer, 2002. (Natural Computing Series). ISBN 978-3-540-40184-1. Citado na página 65.
- BÄCK, T.; SCHWEFEL, H.-P. An overview of evolutionary algorithms for parameter optimization. **Evolutionary Computation**, MIT Press, v. 1, n. 1, p. 1–23, 1993. Citado na página 36.
- BAI, H.; CHENG, R.; JIN, Y. Evolutionary reinforcement learning: A survey. **Intelligent Computing**, Science Partner Journal (SPJ), v. 2, n. 1, p. 659–697, 2023. Disponível em: <<https://spj.science.org/doi/full/10.34133/icomputing.0025>>. Citado na página 11.
- BEYER, H.-G.; SCHWEFEL, H.-P. **Evolution Strategies – A Comprehensive Introduction**. [S.l.]: Springer, 2002. Citado na página 66.
- BLOEMBERGEN, D.; TUYLS, K.; HENNES, D.; KAISERS, M. Evolutionary dynamics of multi-agent learning: A survey. **Journal of Artificial Intelligence Research**, AI Access Foundation, v. 53, p. 659–697, 2015. Disponível em: <<https://www.jair.org/index.php/jair/article/view/10952>>. Citado na página 11.
- CAMPOS, L. M. L. d.; PEREIRA, J. H. A. Algoritmo neuroevolucionário que implementa comportamentos inatos por meio de agentes autônomos. **Brazilian Journal of Development**, v. 5, n. 12, p. 30402–30418, 2019. Citado na página 18.
- DARWIN, C. **On the Origin of Species by Means of Natural Selection, or the Preservation of Favoured Races in the Struggle for Life**. 1st. ed. London: John Murray, 1859. Citado na página 14.
- DYULICHEVA, Y. Y.; GLAZIEVA, A. O. Game based learning with artificial intelligence and immersive technologies: An overview. **CS&SE@ SW**, p. 146–159, 2021. Citado na página 18.
- EIBEN, Á. E.; HINTERDING, R.; MICHALEWICZ, Z. Parameter control in evolutionary algorithms. **IEEE Transactions on evolutionary computation**, IEEE, v. 3, n. 2, p. 124–141, 1999. Citado na página 60.
- EIBEN, A. E.; SMITH, J. E. **Introduction to Evolutionary Computing**. Berlin, Heidelberg: Springer-Verlag, 2003. (Natural Computing Series). ISBN 978-3-540-40187-2. Citado na página 36.
- _____. **Introduction to Evolutionary Computing**. 2nd. ed. [S.l.]: Springer-Verlag Berlin Heidelberg, 2015. (Natural Computing Series). ISBN 978-3-662-44873-1. Citado na página 60.

EL-OMARI, N. K. T. Sea lion optimization algorithm for solving the maximum flow problem. **International Journal of Computer Science and Network Security (IJCSNS)**, v. 20, n. 8, p. 30–38, 8 2020. Citado na página 14.

FOGEL, D. B. **Blondie24: Playing at the Edge of AI**. [S.l.]: Morgan Kaufmann, 2002. Citado na página 18.

_____. **Evolutionary Computation: Toward a New Philosophy of Machine Intelligence**. 3rd. ed. Piscataway, NJ: John Wiley & Sons, 2006. (IEEE Press Series on Computational Intelligence). ISBN 978-0-471-66951-7. Citado na página 35.

GABRIEL, P. H. R.; DELBEM, A. C. B. Fundamentos de algoritmos evolutivos. In: **Anais do Simpósio de Sistemas Computacionais (STB)**. São Carlos, SP: Universidade de São Paulo, 2008. Citado na página 20.

GOEBLER, R. **Agentes inteligentes para pesquisas na WEB**. Monografia (Trabalho de Conclusão de Curso (Graduação em Sistemas de Informação)) — Universidade Federal de Santa Catarina, Florianópolis, 2004. Citado na página 17.

GOLDBERG, D. E. **Genetic Algorithms in Search, Optimization, and Machine Learning**. Reading, MA: Addison-Wesley, 1989. ISBN 0-201-15767-5. Citado 4 vezes nas páginas 14, 21, 22 e 30.

GOLDBERG, D. E.; DEB, K. A Comparative Analysis of Selection Schemes Used in Genetic Algorithms. In: RAWLINS, G. J. (Ed.). **Foundations of Genetic Algorithms**. San Francisco: Morgan Kaufmann, 1991. p. 69–93. Citado na página 59.

HARIK, G. R.; GOLDBERG, D. E. The linkage learning genetic algorithm. **Evolutionary Computation**, MIT Press, v. 7, n. 3, p. 231–253, 1999. Citado na página 32.

HINTERDING, R.; MICHALEWICZ, Z.; EIBEN, A. E. Adaptation in evolutionary computation: A survey. In: IEEE. **Proceedings of 1997 IEEE International Conference on Evolutionary Computation (ICEC'97)**. [S.l.], 1997. p. 65–69. Citado na página 60.

HOLLAND, J. H. Adaptation in natural and artificial systems. **Ann Arbor: The University of Michigan Press**, v. 32, 1975. Citado 4 vezes nas páginas 14, 21, 22 e 30.

IDEAS, R. **Ludo Game**. 2026. Acessado em: 4 fev. 2026. Disponível em: <<https://play.google.com/store/apps/details?id=com.renderedideas.ludogame>>. Citado 2 vezes nas páginas 6 e 19.

JONG, A. K. D. **An analysis of the behavior of a class of genetic adaptive systems**. Tese (Doutorado) — University of Michigan, 1975. Citado na página 30.

JUNIOR, F. C. d. L. **Algoritmo Q-learning como estratégia de exploração e/ou exploração para metaheurísticas GRASP e algoritmo genético**. Tese (Doutorado) — Universidade Federal do Rio Grande do Norte, Natal, RN, 2009. Citado na página 15.

- LIU, S.; AL. et. Rolling horizon evolutionary algorithms for stochastic game environments. **Expert Systems with Applications**, v. 191, p. 116244, 2022. ISSN 0957-4174. Citado na página 11.
- LUCAS, D. C. Algoritmos genéticos: uma introdução. **Apostila referente a disciplina de Inteligencia Computacional, Universidade Federal do Rio Grande do Sul, Brasil**, 2002. Citado na página 16.
- MARTINS, V. S.; FREITAS, J. C. G. de; MENDES, I. N.; TEIXEIRA, O. N. Algoritmos genéticos aplicado ao mundo de wumpus: uma comparação entre agentes baseados em regras e agentes inteligentes. In: SBC. **Escola Regional de Alto Desempenho Norte 2 (ERAD-NO2) e Escola Regional de Aprendizado de Máquina e Inteligência Artificial Norte 2 (ERAMIA-NO2)**. [S.l.], 2021. p. 33–36. Citado na página 18.
- MEDINA, J.; MÜLLER, R. M. A utilização de algoritmos genéticos no desenvolvimento de jogos. **ANAIS DO ENIC**, n. 1, 2009. Citado na página 18.
- MENDES, R.; TOGELIUS, J.; NEALEN, A. Hyper-heuristic evolution of strategy game players. In: **Proceedings of the IEEE Conference on Games (CoG)**. [S.l.: s.n.], 2020. p. 120–127. Citado na página 11.
- MICHALEWICZ, Z. **Genetic Algorithms + Data Structures = Evolution Programs**. 3rd, revised and extended. ed. Berlin, Heidelberg: Springer Science & Business Media, 1996. ISBN 978-3-540-60676-5. Citado 2 vezes nas páginas 34 e 65.
- MILLER, B. L.; GOLDBERG, D. E. et al. Genetic algorithms, tournament selection, and the effects of noise. **Complex systems**, [Champaign, IL, USA: Complex Systems Publications, Inc., c1987-, v. 9, n. 3, p. 193–212, 1995. Citado 2 vezes nas páginas 30 e 59.
- MUNIZ, L. R. R. **Algoritmo genético aplicado ao problema do 8-puzzle**. Dissertação (Trabalho de Conclusão de Curso (Graduação)) — Universidade Federal de Uberlândia, Uberlândia, 2025. Citado na página 21.
- NETO, L. P. F. G. **Development of a bot like entity to emulate an user in a tridimensional virtual environment**. Dissertação (Mestrado) — Universidade do Minho (Portugal), 2023. Citado na página 18.
- ONTANÓN, S.; SYNNAEVE, G.; URIARTE, A.; ESCALANTE, H.; CASTRO, D.; BIDABE, D. A survey of real-time strategy game ai research and competition in starcraft. **IEEE Transactions on Computational Intelligence and AI in Games**, IEEE, v. 5, n. 4, p. 293–311, 2013. Disponível em: <<https://ieeexplore.ieee.org/abstract/document/6637024>>. Citado na página 11.
- PACHECO, M. A. C. et al. Algoritmos genéticos: princípios e aplicações. **ICA: Laboratório de Inteligência Computacional Aplicada. Departamento de Engenharia Elétrica. Pontifícia Universidade Católica do Rio de Janeiro. Fonte desconhecida**, v. 28, p. 22, 1999. Citado na página 11.
- PEGORIN, D. F.; MENECELLI, R. C.; SILVA, P. P. Desenvolvimento de agentes inteligentes em jogos de tabuleiro digitais através de algoritmo genético. **RETEC, Ourinhos**, v. 15, n. 1, p. 15–32, 2022. Citado na página 18.

- PEREZ-LIEBANA, D.; AL. et. General video game ai: A multi-track framework for evaluating agents, games, and content generation. In: **IEEE Transactions on Games**. [S.l.: s.n.], 2021. v. 11, p. 195–208. Citado na página 11.
- PILA, A. D. História e terminologia a respeito da computação evolutiva. **Revista de Ciências Exatas e Tecnologia**, v. 1, n. 1, p. 40–50, 2006. Citado na página 14.
- POZO, A.; CAVALHEIRO, A. d. F.; ISHIDA, C.; SPINOSA, E.; RODRIGUES, E. M. Computação evolutiva. **Universidade Federal do Paraná, 61p.(Grupo de Pesquisas em Computação Evolutiva, Departamento de Informática-Universidade Federal do Paraná)**, 2005. Citado 4 vezes nas páginas 6, 15, 16 e 17.
- RECHENBERG, I. **Evolutionsstrategie: Optimierung technischer Systeme nach Prinzipien der biologischen Evolution**. Stuttgart: Frommann-Holzboog, 1973. ISBN 3-7728-0373-3. Citado 2 vezes nas páginas 36 e 60.
- RICHARDS, N.; MORIARTY, D. E.; MIIKKULAINEN, R. Evolving neural networks to play go. **Applied intelligence**, Springer, v. 8, n. 1, p. 85–96, 1998. Citado na página 18.
- SALLABERRY, C.; LADEIRA, M. Evolutionary adaptation of evaluation functions in board games. **Journal of Intelligent Robotic Systems**, v. 99, p. 455–470, 2020. Citado na página 11.
- SCHMIELE, D. A. F. **Algoritmos genéticos aplicados à análise estratégica de jogos de tabuleiro**. Dissertação (Mestrado) — Universidade Federal de Uberlândia, 2023. Citado 6 vezes nas páginas 12, 21, 22, 24, 59 e 65.
- SCHWEFEL, H.-P. **Numerical Optimization of Computer Models**. Chichester, UK: John Wiley & Sons, Inc., 1981. Tradução do original em alemão: Numerische Optimierung von Computer-Modellen mittels der Evolutionsstrategie (1977). ISBN 0471099880. Citado na página 35.
- SILVA, P. M. F. d. **An evaluation of deep reinforcement learning & neuroevolution in stealth game problems**. Dissertação (Mestrado) — IADE – Faculdade de Design, Tecnologia e Comunicação, 2024. Citado na página 18.
- SILVA, R. M. **Computação evolutiva aplicada a jogos no estilo presa-predador**. Dissertação (Mestrado) — Universidade Federal de Uberlândia, 2019. Citado na página 18.
- SPEARS, W. M.; JONG, K. A. D. On the virtues of parameterized uniform crossover. In: **Proceedings of the 4th International Conference on Genetic Algorithms**. [S.l.]: Morgan Kaufmann, 1991. p. 230–237. Citado na página 31.
- SYSWERDA, G. Uniform crossover in genetic algorithms. In: MORGAN KAUFMANN PUBLISHERS INC. **Proceedings of the 3rd International Conference on Genetic Algorithms**. San Mateo, CA, USA, 1989. p. 2–9. Citado 2 vezes nas páginas 60 e 65.
- TOGELIUS, J.; YANNAKAKIS, G. N.; BAUMGARTEN, T.; LUCAS, S. M.; PEREZ-LIEBANA, D.; MAHLMANN, T.; NEALEN, A. A survey of procedural content generation methods for games. **IEEE Transactions on Computational**

Intelligence and AI in Games, IEEE, v. 3, n. 2, p. 141–152, 2011. Disponível em: <https://ieeexplore.ieee.org/abstract/document/5756645>. Citado na página 11.

VILAS-BOAS, C. R. **Introdução ao estudo de grafos: origem e aplicações**. Tese (Doutorado) — [sn], 2016. Citado 3 vezes nas páginas 15, 16 e 17.

WANG, Y.; WANG, Y.; TIAN, F.; MA, J.; JIN, Q. Intelligent games meeting with multi-agent deep reinforcement learning: a comprehensive review. **Artificial Intelligence Review**, Springer, v. 58, n. 6, p. 165, 2025. Citado na página 18.

WITTKAMP, M.; BARONE, L. Evolving adaptive play for the game of spoof using genetic programming. In: IEEE. **2006 IEEE Symposium on Computational Intelligence and Games**. [S.l.], 2006. p. 164–172. Citado na página 20.

ZHANG, Y.; AL. et. Strategic evolution in multi-agent systems using genetic algorithms. **IEEE Transactions on Games**, v. 13, n. 2, p. 145–158, 2021. Citado na página 11.