

Análise Comparativa de Estratégias de Controle Populacional em Algoritmos Genéticos Aplicados ao Problema das N-Rainhas

Vinícius Rubin¹, Igor Yepes¹

¹Instituto Federal de Educação, Ciência e Tecnologia Farroupilha –
Campus Frederico Westphalen (IFFAR-FW)

Abstract. *This work compares four population control strategies in genetic algorithms applied to the n -queens problem. Using a unified stopping criterion based solely on the maximum number of fitness evaluations, the study isolates the effect of each substitution mechanism on convergence, solution quality, and success rate. Results show that population control strongly influences evolutionary behavior: age-based control offers the best balance between exploration and efficiency, elitism excels in larger instances, and full or partial replacement degrades as diversity decreases. **Keywords:** Genetic Algorithms; Population Control; Evolutionary Computation; N-Queens.*

Resumo. *Este trabalho compara quatro estratégias de controle populacional em algoritmos genéticos aplicados ao problema das n -rainhas. Com um critério de parada unificado baseado apenas no número máximo de avaliações, o estudo isola o impacto das políticas de substituição sobre a convergência, a qualidade das soluções e a taxa de sucesso. Os resultados mostram que o controle populacional influencia fortemente a dinâmica evolutiva: o controle por idade apresenta melhor equilíbrio geral, o elitismo se destaca em instâncias maiores, e as estratégias de substituição total e parcial perdem desempenho com a redução da diversidade. **Palavras-chave:** Algoritmos Genéticos; Controle Populacional; Computação Evolutiva; N-Rainhas.*

1. Introdução

Algoritmos Genéticos (AGs) são métodos de busca inspirados nos mecanismos de evolução natural, amplamente empregados na resolução de problemas complexos devido à sua capacidade de equilibrar exploração estocástica e intensificação por meio de operadores simples de reprodução [Eiben and Smith 2015, Holland 1975]. Esses algoritmos têm sido aplicados em diversas áreas — de problemas combinatórios clássicos a sistemas de engenharia — e sua eficiência depende de um conjunto de decisões de projeto que afetam diretamente o processo evolutivo.

Entre esses componentes, o controle populacional destaca-se como um dos fatores menos padronizados e, ao mesmo tempo, mais influentes no comportamento dos AGs. Esse mecanismo determina como indivíduos são preservados, substituídos ou eliminados ao longo das gerações, afetando diversidade, pressão seletiva, estabilidade evolutiva e risco de convergência prematura. Embora estratégias como substituição total, substituição parcial, elitismo e técnicas baseadas em ciclos de vida sejam amplamente relatadas na literatura, comparações diretas entre elas são raras. Os estudos existentes geralmente

variam simultaneamente representação, operadores ou configurações experimentais, o que dificulta isolar o impacto específico da política de substituição.

Essa ausência de padronização configura uma lacuna importante na área: ainda não está claro, sob condições controladas, como diferentes estratégias de controle populacional influenciam taxa de sucesso, custo computacional e profundidade de busca em AGs. Assim, surge a seguinte pergunta de pesquisa:

Como diferentes estratégias de controle populacional afetam o desempenho de Algoritmos Genéticos quando todos os demais componentes do algoritmo permanecem constantes?

Para investigar essa questão, este trabalho realiza uma comparação sistemática entre quatro mecanismos de controle populacional — substituição total, substituição parcial, elitismo e uma abordagem baseada em idade inspirada no modelo GAVaPS — mantendo fixos representação, função de aptidão, seleção, cruzamento e mutação. O problema das n -rainhas é adotado como cenário experimental por constituir um benchmark clássico, escalável e amplamente utilizado na avaliação de heurísticas evolutivas.

A contribuição principal deste estudo consiste em fornecer uma análise empírica rigorosa e reprodutível sobre o efeito de diferentes políticas de substituição na dinâmica evolutiva dos AGs. Especificamente, o trabalho:

1. compara, sob condições uniformes, o desempenho de quatro estratégias de controle populacional;
2. avalia como cada mecanismo influencia taxa de sucesso, consumo de avaliações, qualidade das soluções e comportamento evolutivo;
3. discute vantagens, limitações e cenários de aplicação de cada abordagem.

Os experimentos incluem múltiplas execuções independentes por configuração e utilizam um critério de parada unificado, baseado exclusivamente em um limite máximo de avaliações da função de aptidão. Esse formato garante igualdade de orçamento computacional entre as estratégias testadas e possibilita uma comparação justa, permitindo medir médias, variâncias e padrões de convergência sob condições estritamente controladas.

2. Fundamentação Teórica

A fundamentação teórica deste trabalho apresenta os conceitos necessários para compreender o funcionamento de Algoritmos Genéticos (AGs), sua aplicação ao problema das n -rainhas e o papel das estratégias de controle populacional na dinâmica evolutiva. Inicialmente, são discutidos os princípios básicos dos AGs, com ênfase em seus componentes estruturais. Em seguida, descreve-se o fluxo geral de um AG clássico, por meio de um pseudocódigo que servirá de referência conceitual para todas as variantes analisadas. Depois, revisita-se o problema das n -rainhas como cenário experimental amplamente utilizado na literatura. São também apresentados os principais elementos do ambiente computacional adotado, com foco em bibliotecas numéricas para experimentação reprodutível. Por fim, são detalhadas as estratégias de controle populacional e os trabalhos relacionados que motivam o recorte específico deste estudo.

2.1. Algoritmos Genéticos

Algoritmos Genéticos (AGs) são meta-heurísticas inspiradas em princípios de evolução biológica, nas quais uma população de soluções candidatas é iterativamente modificada por operadores de seleção, cruzamento e mutação [Eiben and Smith 2015, Holland 1975]. Cada indivíduo é avaliado por uma função de aptidão (*fitness*), e o processo evolutivo tende a favorecer configurações de melhor desempenho ao longo das gerações, compondo um ciclo de variação e seleção que equilibra exploração e intensificação no espaço de busca.

O comportamento de um AG é fortemente influenciado por três grupos de decisões de projeto [de Lacerda and de Leon Ferreira de Carvalho 1999]: (i) a forma de representação dos indivíduos (ou codificação das soluções), (ii) a função de avaliação (função de aptidão) e (iii) os operadores evolutivos e a política de substituição populacional.

A literatura tradicional em AGs dedica grande atenção à escolha da representação (binária, inteira, permutacional, real, entre outras) e ao desenho de operadores de cruzamento e mutação adequados ao problema [Goldberg 1989, Eiben and Smith 2015]. Em contraste, o *controle populacional* — isto é, a forma como indivíduos são mantidos, substituídos, removidos ou preservados ao longo das gerações — costuma ser tratado como um detalhe de implementação, apesar de influenciar diretamente aspectos como diversidade, pressão seletiva, convergência e risco de aprisionamento em ótimos locais [Whitley 1994, Mitchell 1996].

Neste trabalho, representação, função de aptidão, seleção, cruzamento e mutação são mantidos fixos, e o foco recai exclusivamente sobre o componente de controle populacional. Essa escolha permite isolar e mensurar o impacto de diferentes políticas de substituição no desempenho global do algoritmo, mantendo constantes os demais fatores que tradicionalmente afetam a qualidade da busca.

2.2. Fluxo Geral de um Algoritmo Genético

Embora existam inúmeras variantes, o fluxo de execução de um AG tradicional segue uma estrutura geral bem estabelecida, composta por inicialização da população, avaliação, reprodução e atualização da população ao longo de múltiplas gerações [Eiben and Smith 2015, Mitchell 1996]. Critérios de parada usuais incluem: (i) número máximo de gerações, (ii) orçamento máximo de avaliações da função de aptidão e (iii) limite de tempo de execução.

O pseudocódigo apresentado a seguir resume o processo e serve como modelo conceitual sobre o qual as diferentes estratégias de controle populacional serão discutidas. No contexto deste trabalho, o critério de parada é posteriormente especializado para um limite máximo de avaliações (ver Seção 3).

Algoritmo 1 Pseudocódigo de um Algoritmo Genético Clássico

Entrada: tamanho populacional N , orçamento máximo de avaliações $A_{\max}$, operadores genéticos

Saída: melhor solução encontrada

```
1: Inicializar a população  $P$  com  $N$  indivíduos
2: Avaliar  $P$  e inicializar o contador de avaliações  $A \leftarrow N$ 
3: while critério de parada não satisfeito e  $A < A_{\max}$  do
  // Seleção e Reprodução
4:   Selecionar pais utilizando torneio (ou outro método de seleção)
5:   Aplicar operador de cruzamento para gerar descendentes
6:   Aplicar operador de mutação aos descendentes
7:   Avaliar os descendentes e atualizar  $A$ 
  // Controle Populacional
8:   Combinar populações conforme a estratégia escolhida:
    – substituição total, ou
    – substituição parcial, ou
    – elitismo, ou
    – controle por idade (GAVaPS/Michalewicz)
9:   Atualizar a população  $P$ 
10: end while
11: Retornar o melhor indivíduo de  $P$ 
```

Esse pseudocódigo é utilizado como referência conceitual para as variações testadas neste estudo, alterando-se apenas a estratégia de controle populacional aplicada em cada experimento, enquanto os demais componentes permanecem inalterados. A especialização para um critério de parada baseado exclusivamente em avaliações é detalhada na Seção 3.

2.3. O Problema das n -Rainhas

O problema das n -rainhas consiste em posicionar n rainhas em um tabuleiro de xadrez de dimensões $n \times n$ de forma que nenhuma delas ataque outra, o que implica ausência de conflitos em linhas, colunas e diagonais [Bell and Stevens 2009]. Embora a formulação seja simples, o número de configurações possíveis cresce rapidamente com n , tornando abordagens exaustivas inviáveis para instâncias maiores e caracterizando um problema combinatório de difícil exploração.

Estudos como o de Ali et al. [Ali et al. 2023] mostram que buscas clássicas (BFS/DFS) são eficientes apenas para instâncias pequenas, ao passo que, para valores maiores de n , meta-heurísticas, como AGs, tornam-se alternativas mais competitivas. Essa combinação de estrutura simples, dificuldade ajustável e ampla presença na literatura torna o problema das n -rainhas um *benchmark* natural para avaliar variações de AGs [Sarkar and Nag 2017, Farhan et al. 2015].

Neste trabalho, o problema das n -rainhas é adotado como cenário experimental para comparar estratégias de controle populacional, beneficiando-se de uma representação permutacional que elimina conflitos de linha e coluna e concentra a função de aptidão na

contagem de conflitos diagonais. Essa característica contribui para reduzir o impacto de soluções inviáveis e realçar o efeito das diferentes políticas de substituição populacional.

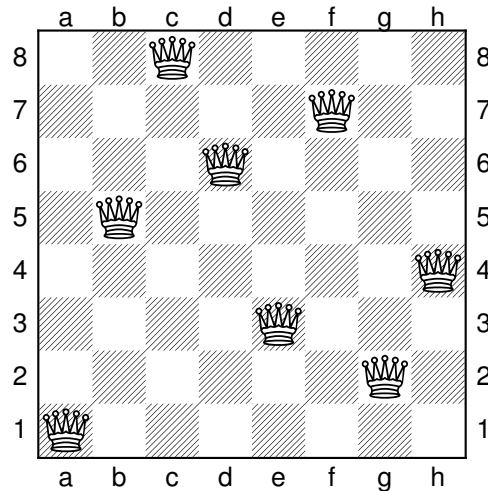


Figura 1. Exemplo de solução válida para o problema das 8-rainhas.

2.4. Uso da Linguagem Python e Bibliotecas Relacionadas

A implementação experimental foi desenvolvida em `Python`, linguagem amplamente adotada em computação científica pela clareza sintática e pelo ecossistema consolidado de bibliotecas numéricas [Python Software Foundation 2025a]. Essa escolha facilita a construção de protótipos, a automatização de experimentos e a reprodutibilidade dos resultados.

A biblioteca `NumPy` foi utilizada para operações vetorizadas e para a representação eficiente da população [Harris et al. 2020]. A geração de números aleatórios baseou-se em `numpy.random` [Python Software Foundation 2025b], garantindo controle explícito de sementes e replicação de cenários. A visualização dos resultados foi realizada com `matplotlib` [Hunter 2007], permitindo a construção de gráficos para análise comparativa das estratégias de controle populacional.

Esse conjunto de ferramentas permite concentrar o esforço no desenho dos experimentos e na análise do comportamento dos AGs, reduzindo interferências de infraestrutura e favorecendo a reprodutibilidade.

2.5. Técnicas de Controle Populacional

2.5.1. Substituição Total

Na substituição total, a cada geração todos os indivíduos da população corrente são substituídos pelos descendentes recém-gerados. Em termos operacionais, o algoritmo:

1. seleciona pares de pais na população atual;
2. aplica cruzamento e mutação para produzir uma nova população de tamanho N ;
3. descarta integralmente a população anterior.

Trata-se do esquema mais simples e amplamente utilizado em AGs, pois facilita a implementação e o controle de parâmetros [Goldberg 1989, Whitley 1994]. No entanto, por descartar toda a geração anterior, tende a reduzir a diversidade rapidamente, tornando o algoritmo mais suscetível à convergência prematura, especialmente quando a pressão seletiva é alta ou a mutação é conservadora [De Jong 1975, Spears 1993]. A substituição total é frequentemente utilizada como referência de comparação para outras políticas de controle populacional.

2.5.2. Substituição Parcial

Na substituição parcial, apenas uma fração da população é renovada a cada geração, caracterizando um modelo do tipo *steady-state* [Eiben and Smith 2015, Whitley 1994]. De forma geral, o processo segue os passos:

1. a partir da população atual, selecionam-se pais e produzem-se alguns descendentes;
2. esses descendentes são inseridos na população;
3. um subconjunto de indivíduos é removido, preservando o tamanho populacional.

A fração de indivíduos substituída em cada geração controla o ritmo de renovação: valores menores favorecem uma evolução mais gradual, enquanto valores maiores aproximam o comportamento do modelo geracional [Goldberg 1989, Mitchell 1996].

Na literatura, diferentes critérios podem ser utilizados para decidir quais indivíduos permanecem ou são removidos (por aptidão, aleatoriedade, idade, entre outros). Neste trabalho, a estratégia de substituição parcial preserva uma fração fixa de indivíduos escolhidos **aleatoriamente**, sem preferência explícita pelos mais aptos. O elitismo, por sua vez, é tratado como uma estratégia distinta, responsável por garantir a preservação sistemática dos melhores indivíduos.

2.5.3. Elitismo

O elitismo introduz um mecanismo explícito de preservação dos melhores indivíduos [Goldberg 1989, Whitley 1994]. Em cada geração:

1. uma pequena fração dos indivíduos com melhor aptidão é copiada diretamente para a próxima geração;
2. o restante da população é preenchido com descendentes produzidos pelos operadores de cruzamento e mutação;
3. indivíduos menos aptos são descartados para manter o tamanho populacional.

Esse mecanismo garante que as melhores soluções encontradas não sejam perdidas por efeito do acaso, o que tende a acelerar a convergência [De Jong 1975, Goldberg 1989]. Por outro lado, a preservação sistemática de indivíduos muito aptos pode reduzir a diversidade e acentuar o risco de estagnação em ótimos locais, sobretudo quando a população é pequena [Eiben and Smith 2015, de Lacerda and de Leon Ferreira de Carvalho 1999]. Assim, o elitismo representa um reforço da pressão seletiva que pode ser benéfico ou prejudicial, dependendo do equilíbrio com outros componentes do algoritmo.

2.5.4. Controle Populacional por Idade (GAVaPS / Michalewicz)

A estratégia de controle por idade baseia-se no modelo *Genetic Algorithms with Varying Population Size* (GAVaPS) proposto por Michalewicz [Michalewicz 1994]. Nesse esquema, cada indivíduo passa a carregar, além do cromossomo e da aptidão, duas variáveis adicionais:

- sua **idade** (número de gerações sobrevividas);
- um **tempo de vida** máximo (*lifetime*), definido no momento do nascimento.

A premissa fundamental é que indivíduos mais aptos recebam um *lifetime* maior, permanecendo por mais gerações na população, enquanto indivíduos menos aptos são naturalmente descartados mais cedo [Michalewicz 1994]. O cálculo do *lifetime* é tipicamente obtido a partir da aptidão relativa do indivíduo, considerando estatísticas globais da população (mínimo, máximo e média). Essa aptidão é então mapeada por uma função — frequentemente bi-linear — para o intervalo discreto $[L_{\min}, L_{\max}]$. Assim, indivíduos de melhor desempenho recebem tempos de vida próximos a $L_{\max}$, ao passo que indivíduos fracos aproximam-se de $L_{\min}$.

O ciclo evolutivo do GAVaPS segue, de forma simplificada, três etapas principais [Michalewicz 1994, Eiben and Smith 2015]:

1. incrementar a idade de todos os indivíduos;
2. remover aqueles cuja idade ultrapassou seu *lifetime*;
3. gerar novos descendentes — via seleção, cruzamento e mutação — em quantidade proporcional ao tamanho da população atual, permitindo que o número total de indivíduos varie dinamicamente.

Diferentemente das estratégias tradicionais de substituição, em que a remoção é guiada pela aptidão instantânea, aqui a eliminação ocorre por envelhecimento. Mesmo indivíduos muito aptos eventualmente saem da população ao atingirem o limite de vida. Esse mecanismo introduz uma forma de *diversidade temporal*, impedindo que poucas soluções dominem indefinidamente a população e promovendo um equilíbrio mais natural entre intensificação (preservar bons indivíduos por mais tempo) e exploração (renovar continuamente a população) [Michalewicz 1994].

No contexto deste trabalho, o controle por idade é comparado diretamente com as políticas de substituição total, substituição parcial e elitismo, de modo a avaliar como a introdução do *lifetime* e da variação dinâmica do tamanho populacional impacta a taxa de sucesso, o esforço computacional (em número de avaliações) e o comportamento evolutivo de AGs aplicados ao problema das n -rainhas [Richter and Berlezi 2012, Sarkar and Nag 2017, Ali et al. 2023].

2.6. Trabalhos Relacionados

Diversos estudos utilizam o problema das n -rainhas como *benchmark* para investigar variações em AGs, especialmente no que diz respeito à representação e ao controle de diversidade. Esta subseção destaca trabalhos que fundamentam o desenho experimental adotado e situam a contribuição deste estudo, focado no impacto de diferentes estratégias de controle populacional.

2.6.1. Representações Binária e Inteira para o Problema das n -Rainhas [Richter and Berlezi 2012]

Richter et al. comparam uma codificação binária e uma codificação inteira do tipo *InOrder*. A primeira exige conversão e permite conflitos em linhas; a segunda mapeia diretamente colunas para linhas, evitando soluções inviáveis. Os autores mostram que a representação *InOrder* reduz conflitos e acelera a convergência, reforçando a importância de uma representação adequada antes de comparar variações de um AG. O presente trabalho adota uma representação por permutação semelhante justamente para isolar o efeito das estratégias de substituição populacional.

2.6.2. Comparação entre Algoritmo Genético e Buscas Clássicas (BFS/DFS) [Ali et al. 2023]

Ali et al. comparam AG, BFS e DFS no problema das n -rainhas. Para instâncias pequenas, as buscas clássicas são mais rápidas, mas o AG se torna mais competitivo à medida que o tamanho do tabuleiro cresce. Embora o foco seja a comparação entre paradigmas, o estudo reforça a adequação dos AGs para instâncias maiores. Diferentemente dos autores, este trabalho concentra-se exclusivamente em comparar estratégias dentro do próprio AG.

2.6.3. Algoritmo Genético Adaptativo com Controle de Diversidade [Sarkar and Nag 2017]

Sarkar e Nag apresentam um AG adaptativo com mecanismos explícitos de preservação de diversidade, incluindo mutações mais frequentes e estruturas auxiliares para evitar estagnação. Os resultados mostram boa escalabilidade para N elevados, sugerindo que o modo de renovação populacional influencia diretamente a convergência. Contudo, por combinar várias modificações simultâneas, o efeito de cada componente é difícil de isolar. Aqui, apenas a política de substituição é variada, mantendo-se constantes representação e operadores.

2.6.4. Busca de Todas as Soluções Válidas para as 8-Rainhas [Farhan et al. 2015]

Farhan et al. utilizam um AG para encontrar todas as 92 soluções da instância de 8 rainhas, empregando representação vetorial sem conflitos em linhas/colunas e avaliando apenas ataques diagonais. Embora o objetivo seja diferente — cobertura completa e não comparação entre estratégias — o trabalho ilustra a capacidade dos AGs de explorar múltiplas soluções válidas. Neste estudo, essa característica é explorada para observar como diferentes políticas de substituição e controle por idade afetam a taxa de sucesso ao encontrar ao menos uma solução válida.

3. Metodologia

Esta seção descreve o desenho experimental empregado para comparar quatro estratégias de controle populacional em Algoritmos Genéticos (AGs) aplicados ao problema das n -rainhas. O estudo segue diretrizes de experimentação reprodutível em computação

evolutiva [Eiben and Smith 2015], mantendo constantes todos os componentes do algoritmo que não estão diretamente relacionados ao mecanismo de substituição populacional. Dessa forma, o impacto de cada estratégia pode ser analisado isoladamente, sem interferências provenientes de diferenças estruturais no operador evolutivo.

3.1. Algoritmo Genético Base

A representação utilizada é uma permutação de tamanho n , onde a posição i do vetor indica a linha ocupada pela rainha na coluna i , garantindo a ausência de conflitos em linhas e colunas. Assim, a função de aptidão consiste exclusivamente no número de pares de rainhas em conflito diagonal, métrica de uso amplamente consolidado na literatura e de complexidade $O(n^2)$.

A população inicial é gerada embaralhando sequências de 0 a $n - 1$. O algoritmo utiliza:

- **Seleção:** torneio binário;
- **Cruzamento:** *Order Crossover* (OX), aplicado em lote;
- **Mutação:** troca de duas posições (*swap*);
- **Taxa de mutação:** fixa em 10% para todas as configurações.

A taxa de mutação fixa elimina interações entre parâmetros que poderiam mascarar o efeito das estratégias de controle populacional, permitindo uma comparação mais justa.

3.2. Estratégias de Controle Populacional

Quatro estratégias foram avaliadas:

- **Substituição total:** todos os indivíduos são substituídos a cada geração.
- **Substituição parcial:** uma fração fixa da população é substituída por novos descendentes.
- **Elitismo:** um subconjunto dos melhores indivíduos é preservado integralmente.
- **Controle por idade (GAVaPS):** cada indivíduo recebe um *lifetime* proporcional à sua aptidão e é removido quando o ultrapassa, produzindo uma população de tamanho dinâmico.

Nas estratégias de população fixa, utiliza-se uma população de 100 indivíduos. Para o controle por idade, o mesmo valor é utilizado como população inicial, permitindo variação ao longo da execução.

3.3. Controle por Idade no Contexto Experimental

O controle por idade utilizado segue o modelo descrito na Seção 2. A cada iteração:

1. as idades dos indivíduos são incrementadas;
2. indivíduos cuja idade excede seu *lifetime* são eliminados;
3. seja S o número de sobreviventes, gera-se

$$N_{\text{novos}} = \lfloor \tau \cdot S \rfloor ,$$

com $\tau = 0,5$, produzindo descendentes por seleção, cruzamento e mutação.

Assim, o tamanho da população torna-se dinâmico:

$$\text{Pop}(t + 1) = S + N_{\text{novos}} .$$

Caso $S = 0$, ocorre reinicialização completa da população.

Tabela 1. Configurações avaliadas por estratégia de controle populacional

Estratégia	Parâmetros	Valores
Substituição total	Tamanho da população	100
Substituição parcial	Fração substituída	25%
	Tamanho da população	100
Elitismo	Fração de elite	10%
	Tamanho da população	100
Controle por idade	Intervalo de lifetime	$L_{\min} = 1, L_{\max} = 5$
	Taxa de reprodução	0,5
	Tamanho inicial da população	100

3.4. Desenho Experimental

Para cada estratégia e cada valor de n , realizam-se **100 execuções independentes**, utilizando sementes distintas na inicialização dos geradores de números aleatórios. A repetição independente é fundamental para reduzir variância e permitir inferências confiáveis sobre o desempenho estocástico de cada abordagem.

Critério de parada

Uma mudança central deste trabalho é o uso de um critério de parada exclusivamente baseado no **número total de avaliações da função de aptidão**. Esse critério é amplamente adotado em experimentos científicos com algoritmos evolutivos por refletir diretamente o esforço computacional empregado na busca [Eiben and Smith 2015].

O limite máximo de avaliações é calculado a partir do tamanho da população inicial (denotado por P) e do tamanho do problema n , segundo:

$$\text{limite_avaliacoes} = 2 \cdot P \cdot n.$$

Esse cálculo define um orçamento fixo e igualitário de esforço computacional para todas as estratégias, independentemente de diferenças no custo por geração ou da variação do tamanho populacional no controle por idade. A execução é encerrada somente quando:

- uma solução válida (aptidão zero) é encontrada; ou
- o limite de avaliações é atingido.

O número de gerações executadas passa a ser uma **métrica observada**, e não um critério de parada.

Métricas coletadas

Para cada execução, registram-se:

- **Indicador de sucesso:** se a solução foi encontrada antes do limite;

- **Total de avaliações realizadas;**
- **Número de gerações efetivamente executadas;**
- **Tempo total de execução;**
- **Diversidade populacional média e entropia média,** quando aplicável.

Essas métricas permitem avaliar simultaneamente a eficácia (taxa de sucesso), a eficiência (avaliações até a solução) e o comportamento evolutivo (dinâmica populacional e diversidade) das diferentes estratégias de controle populacional.

4. Desenvolvimento

Esta seção descreve a implementação prática do estudo, desde a organização modular da ferramenta experimental até a condução automatizada das execuções e o processamento dos resultados utilizados na análise apresentada na Seção 5.

4.1. Arquitetura da Ferramenta Experimental

A implementação foi construída como uma *ferramenta de experimentação* em Python, estruturada em módulos independentes e de responsabilidade bem definida. Essa modularização facilita a reprodutibilidade, a substituição de componentes e a adição de novas estratégias ou cenários experimentais.

A arquitetura geral organiza-se nos seguintes blocos:

- **Núcleo do algoritmo genético:** implementa o laço evolutivo, mantém o estado da população, controla o contador de avaliações e aplica os operadores genéticos em cada iteração.
- **Módulo de representação e aptidão:** implementa a codificação em permutação e a função de aptidão para o problema das n -rainhas, incluindo uma versão altamente otimizada com Numba para acelerar a avaliação de conflitos diagonais.
- **Módulo de operadores genéticos:** reúne a lógica de seleção, cruzamento (*OX*) e mutação (*swap*), todos implementados de forma vetorizada para minimizar custos de memória e maximizar paralelismo.
- **Módulo de controle populacional:** implementa as quatro estratégias comparadas — substituição total, substituição parcial, elitismo e controle por idade — de forma isolada, garantindo que o restante do algoritmo permaneça invariável entre cenários.
- **Módulo de experimentos:** responsável por orquestrar execuções repetidas, aplicar o critério de parada baseado no número de avaliações, registrar métricas e armazenar os resultados em formato `.csv` e `.json`.
- **Módulo de registro e persistência:** implementa rotinas de logging e serialização, garantindo rastreabilidade completa das execuções (semente utilizada, parâmetros, duração, avaliações realizadas).

Essa separação possibilita que o mesmo núcleo do AG seja testado com diferentes políticas de substituição, alterando-se apenas o módulo de controle populacional. Como consequência, qualquer diferença observada nos resultados é atribuível à estratégia de controle — e não ao restante do algoritmo.

4.2. Implementação dos Operadores Genéticos

Os operadores genéticos seguem rigorosamente as definições apresentadas na Seção 3. Cada indivíduo é um vetor permutacional de tamanho n , no qual o valor na coluna i indica a linha ocupada pela rainha.

A função de aptidão avalia exclusivamente conflitos diagonais. Sua implementação inclui:

- versões sequencial e paralela otimizadas com Numba;
- buffers pré-alocados para evitar realocação de memória;
- vetorização completa das operações de contagem.

A seleção é realizada por torneio binário vetorizado. O cruzamento *OX* é aplicado em lote, reduzindo o overhead de chamadas repetidas. A mutação por troca também é vetorizada e utiliza um kernel compilado pelo Numba para troca in-place de múltiplos indivíduos em paralelo.

Essas otimizações garantem que o esforço computacional entre as estratégias seja comparável, minimizando vieses relacionados ao custo interno dos operadores.

4.3. Implementação das Estratégias de Controle Populacional

As quatro estratégias avaliadas compartilham operadores e função de aptidão, diferindo apenas na forma como atualizam a população:

- **Substituição total:** toda a população é substituída a cada geração.
- **Substituição parcial:** uma fração fixa da população é substituída; os indivíduos preservados são selecionados *aleatoriamente*, conforme regra estabelecida neste trabalho.
- **Elitismo:** preserva-se uma fração fixa, composta exclusivamente pelos melhores indivíduos.
- **Controle por idade (GAVaPS):** cada indivíduo recebe um *lifetime* calculado via função bilinear dependente da aptidão; indivíduos cuja idade ultrapassa o limite são removidos; o tamanho da população varia dinamicamente a cada iteração.

No controle por idade, a população inicial é de 100 indivíduos, porém o tamanho pode encolher ou crescer a cada iteração, seguindo:

$$\text{Pop}(t + 1) = S + \lfloor 0,5 S \rfloor .$$

Esse comportamento é implementado sem comprometer a comparabilidade entre estratégias, pois o critério de parada depende apenas do número de avaliações da aptidão, não do número de gerações.

4.4. Execução dos Experimentos

A execução de cada cenário segue cinco etapas principais:

1. **Inicialização:** a população é gerada aleatoriamente e os contadores internos (idade, lifetime e avaliações) são zerados.
2. **Laço evolutivo:** seleção, cruzamento, mutação e avaliação dos novos indivíduos são aplicados repetidamente.

3. **Aplicação da política de controle populacional:** a população é atualizada de acordo com a estratégia em estudo.
4. **Critério de parada:** a execução é encerrada quando:

$$\text{quant_avaliacoes} \geq 2 \cdot P \cdot n,$$

onde P é o tamanho da população inicial (100 neste estudo). Se uma solução ótima (aptidão zero) surgir antes desse limite, a execução também é encerrada.

5. **Registro das métricas:** ao final, são salvos em arquivo o número total de avaliações, o tempo total, a quantidade de gerações executadas, a semente, e um indicador de sucesso.

A escolha de utilizar apenas avaliações como critério de parada garante orçamento computacional idêntico entre as estratégias, mesmo quando o número de indivíduos avaliados por geração é variável, como ocorre no controle por idade.

4.5. Organização dos Dados e Pré-processamento

Cada execução gera um registro individual no formato `.csv`, contendo:

- tamanho do problema n ;
- estratégia de controle populacional;
- semente usada;
- avaliações totais realizadas;
- número de gerações executadas;
- tempo total;
- indicador de sucesso.

Antes da análise estatística, realiza-se uma etapa de pré-processamento que inclui:

- **validação de registros**, removendo execuções incompletas;
- **cálculo de métricas derivadas**, como taxa de sucesso, avaliações médias até a solução e desempenho relativo;
- **agrupamento por estratégia e tamanho n** , com geração automática de tabelas e gráficos.

Esse fluxo garante independência total entre a implementação do AG e a análise dos dados, permitindo a replicação do estudo apenas pela adição de novos arquivos de saída e reexecução dos scripts de pré-processamento.

5. Resultados e Discussão

Esta seção apresenta e discute os resultados obtidos nos experimentos descritos na Seção 3. Todas as estratégias foram avaliadas sob as mesmas condições: mesmo AG base, mesmos operadores e, principalmente, o mesmo critério de parada, definido exclusivamente pelo limite total de avaliações da função de aptidão:

$$\text{limite_avaliacoes} = \lfloor 2 \cdot N \cdot n \rfloor,$$

onde N é o tamanho da população e n é o tamanho do tabuleiro. Esse limite estabelece um orçamento computacional igual para todas as estratégias, garantindo que diferenças de desempenho reflitam apenas o comportamento evolutivo de cada mecanismo de controle populacional.

Os resultados exibem valores médios obtidos a partir de múltiplas execuções independentes, permitindo analisar padrões gerais de convergência, qualidade, estabilidade e eficiência. Para cada estratégia de controle populacional, a análise considerou como parâmetros de desempenho o número médio de avaliações da função de aptidão consumidas, a geração na qual a melhor solução foi encontrada, o número médio de gerações necessárias até a convergência, o valor médio da melhor aptidão obtida, a taxa de sucesso na identificação de soluções válidas e o tempo médio total de execução.

5.1. Avaliações médias por n

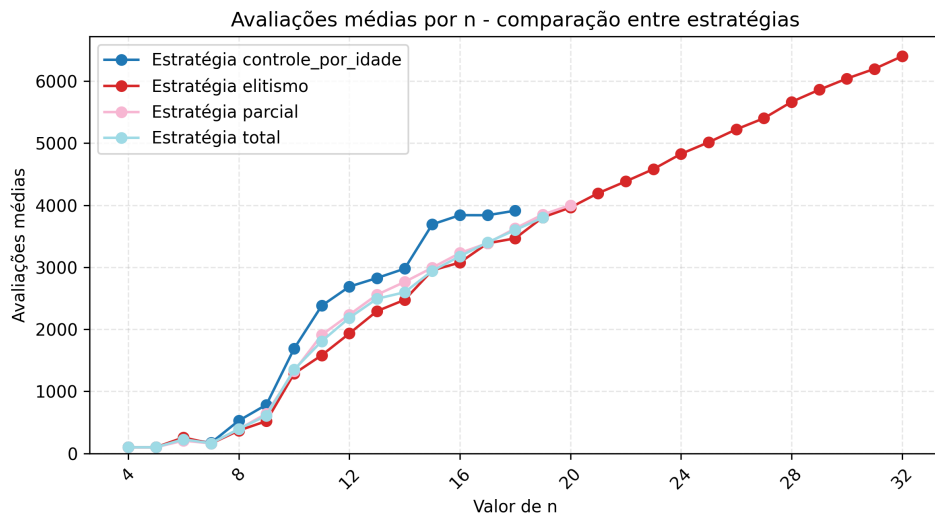


Figura 2. Avaliações médias consumidas por n .

As avaliações médias correspondem diretamente ao **critério de parada adotado neste estudo**: cada execução é interrompida assim que encontra uma solução ou quando atinge o limite máximo de avaliações permitido para aquele valor de n . Como essa métrica é, ao mesmo tempo, o principal limitador computacional e o próprio ponto de convergência do algoritmo, é esperado que diferentes estratégias apresentem valores muito semelhantes em instâncias simples.

Os resultados observados seguem esse comportamento:

- Para instâncias pequenas ($n \leq 12$), todas as estratégias consomem praticamente o mesmo número de avaliações, justamente porque o critério de parada — baseado no limite máximo de avaliações — domina o comportamento e tende a ser atingido de forma quase uniforme entre as abordagens.
- Entre $n = 12$ e $n = 18$, estratégias como **controle por idade**, **substituição parcial** e **substituição total** passam a utilizar mais avaliações que o elitismo, evidenciando maior esforço de busca antes de convergir.
- O **elitismo** apresenta crescimento quase linear no uso de avaliações e, para valores maiores de n , aproxima-se sistematicamente do limite máximo, indicando um processo de busca mais intenso e persistente.

5.2. Geração da melhor solução

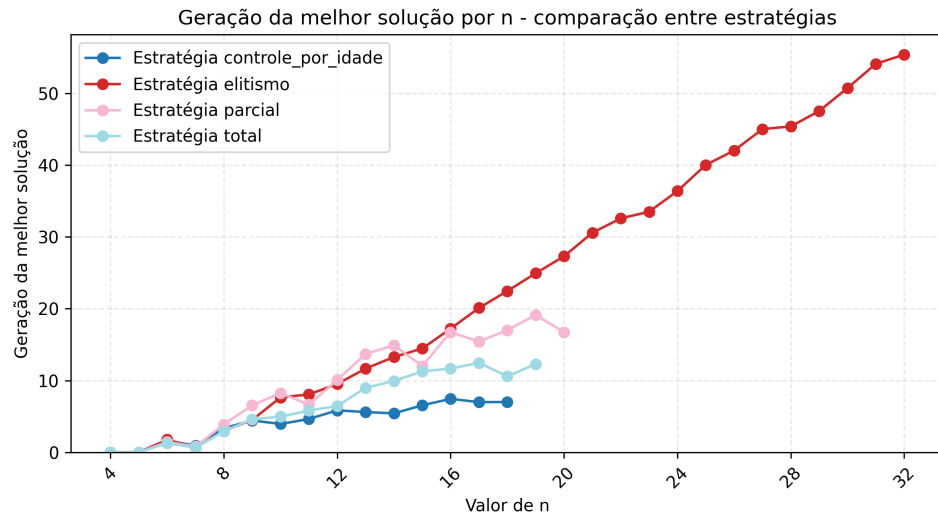


Figura 3. Geração média da melhor solução.

A geração na qual a melhor solução é encontrada reflete a **velocidade inicial de convergência**. Como o critério de parada não depende mais de gerações, esse indicador não controla o término, mas continua útil para caracterizar o comportamento dinâmico das estratégias.

- O **controle por idade** encontra boas soluções cedo e com estabilidade até cerca de $n = 18$.
- O **elitismo** encontra suas melhores soluções de forma mais tardia conforme n aumenta, indicando exploração inicial limitada.
- As estratégias de **substituição total** e **parcial** exibem maior variabilidade e sinais de perda precoce de diversidade.

5.3. Gerações médias

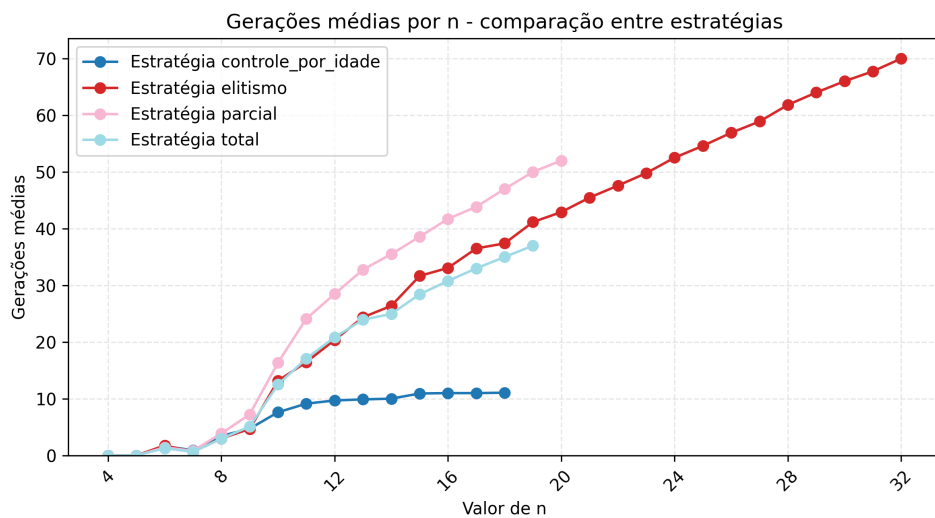


Figura 4. Gerações médias executadas por n .

Com o critério baseado exclusivamente em avaliações, o número de gerações torna-se uma **medida observada**, refletindo quantos ciclos evolutivos cada estratégia consegue executar dentro do mesmo orçamento.

- O **controle por idade** mantém o menor número de gerações até $n \approx 20$, o que sugere ciclos evolutivos mais produtivos.
- O **elitismo** executa muitas gerações, chegando a mais de 70 para $n = 32$, devido à necessidade de refinamento detalhado.
- As estratégias **total** e **parcial** apresentam crescimento rápido no número de gerações, mas sem ganho proporcional de qualidade.

5.4. Melhor aptidão média

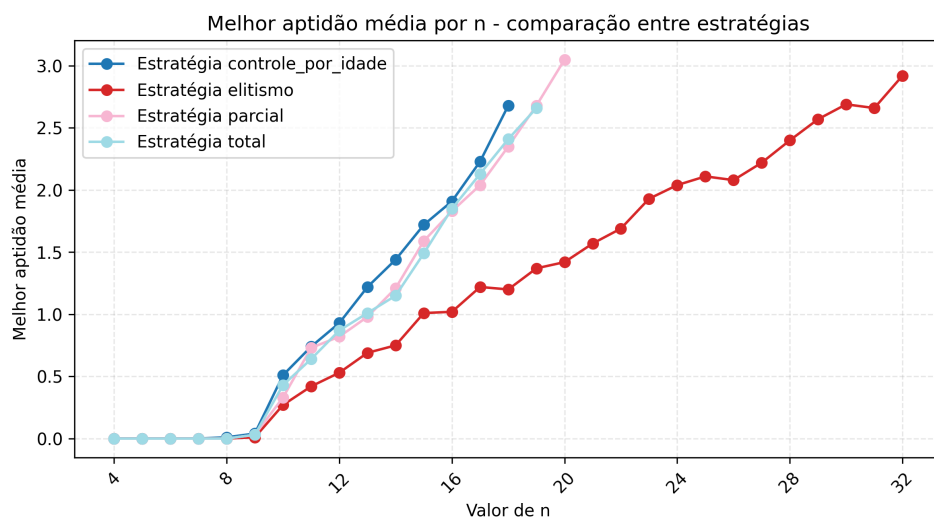


Figura 5. Melhor aptidão média encontrada por n .

A melhor aptidão média representa a **qualidade evolutiva global** das soluções encontradas por cada estratégia.

- Para $n \leq 12$, todas as estratégias encontram soluções ótimas.
- Para $n > 15$, as estratégias **total** e **parcial** deterioram rapidamente, devido à perda de diversidade.
- Para $n > 25$, o **elitismo** alcança as melhores soluções, ainda que com custo temporal maior.

5.5. Taxa de sucesso

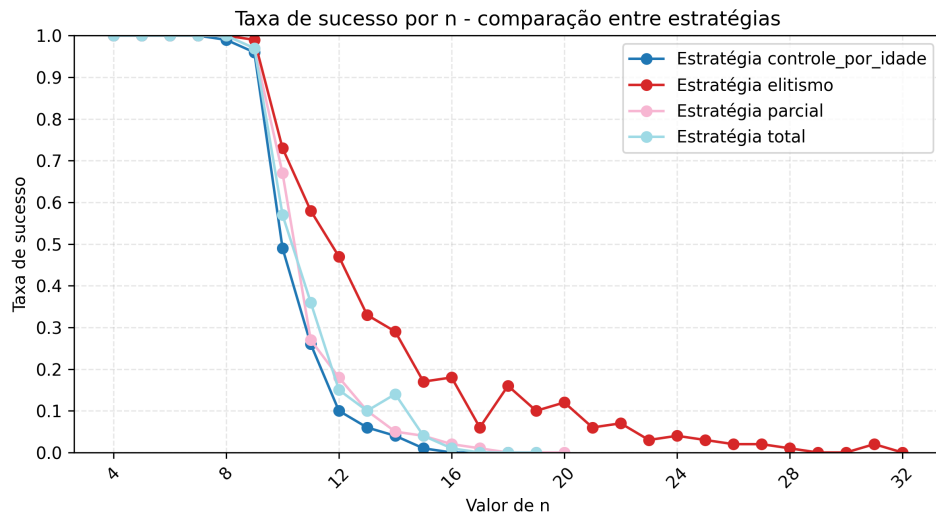


Figura 6. Taxa de sucesso por n .

A taxa de sucesso corresponde à proporção de execuções que encontraram uma solução válida antes de atingir o limite de avaliações.

- Todas as estratégias atingem **100% de sucesso até $n = 10$** .
- As estratégias **substituição total** e **parcial** colapsam a partir de $n \geq 14$.
- O **controle por idade** mantém alto desempenho até $n = 18$.
- O **elitismo** apresenta a melhor taxa de sucesso geral, obtendo soluções até $n \approx 32$.

5.6. Tempo médio de execução

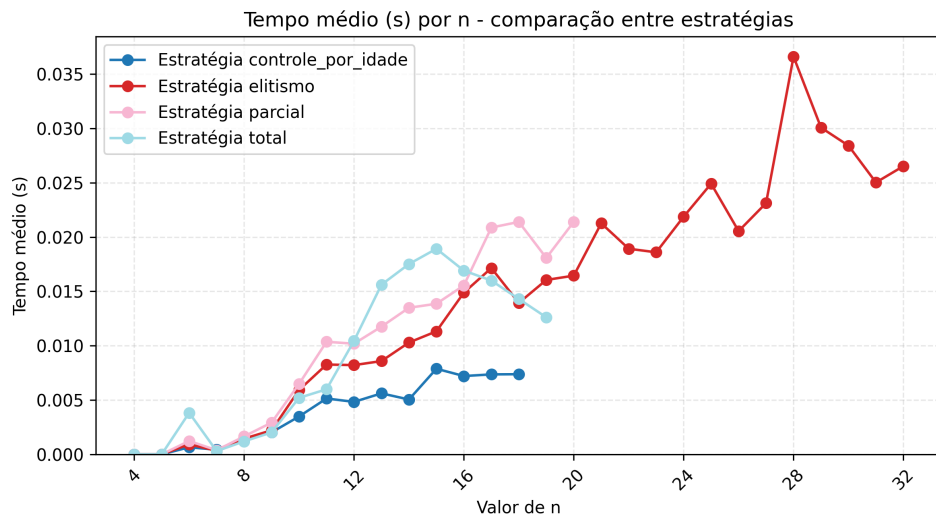


Figura 7. Tempo médio por execução.

O tempo médio de execução está diretamente relacionado ao custo total das avaliações e ao número de gerações possíveis dentro do limite estabelecido.

- As estratégias **total**, **parcial** e **controle por idade** apresentam tempos baixos e estáveis.
- O **elitismo** é mais lento, especialmente para n maiores, devido ao elevado número de gerações executadas.
- O tempo cresce de forma coerente com o número de gerações realizadas em cada estratégia.

5.7. Síntese geral

Os resultados permitem caracterizar claramente o comportamento das estratégias:

- **Elitismo**: Melhor taxa de sucesso e melhores soluções para instâncias grandes, embora demandando mais tempo devido ao grande número de gerações.
- **Controle por idade (GAVaPS)**: Melhor equilíbrio geral entre diversidade, qualidade e eficiência. Apresenta desempenho superior para instâncias de tamanho médio.
- **Substituição parcial e total**: Adequadas apenas para instâncias pequenas; degeneram rapidamente para valores moderados e altos de n por perda extrema de diversidade.

Em síntese, os resultados evidenciam que o **controle populacional** desempenha papel determinante no comportamento de AGs. Mesmo com orçamento idêntico de avaliações, diferentes estratégias produzem padrões de convergência substancialmente distintos, destacando a relevância de políticas de substituição no desenho de heurísticas evolutivas.

6. Conclusão e Trabalhos Futuros

6.1. Conclusões Gerais

Este trabalho investigou o impacto de diferentes estratégias de controle populacional em algoritmos genéticos aplicados ao problema das n -rainhas, adotando um critério de parada unificado baseado exclusivamente no número máximo de avaliações da função de aptidão. A implementação modular desenvolvida permitiu conduzir experimentos sistemáticos, comparáveis e reprodutíveis, possibilitando uma avaliação justa das quatro abordagens estudadas: substituição total, substituição parcial, elitismo e controle por idade (inspirado no GAVaPS de Michalewicz).

Os resultados confirmam que o mecanismo de substituição populacional exerce influência direta e profunda sobre o comportamento evolutivo do AG, afetando tanto a taxa de sucesso quanto a estabilidade da busca, a diversidade populacional e a eficiência computacional. Entre as estratégias avaliadas, o **elitismo** apresentou desempenho superior em instâncias maiores, principalmente por sua habilidade de preservar indivíduos de alta aptidão ao longo das avaliações.

O **controle por idade**, por sua vez, demonstrou o melhor equilíbrio global, combinando capacidade exploratória, preservação natural de diversidade e rapidez na convergência para valores intermediários de n . Já as estratégias de **substituição total** e **substituição parcial** apresentaram desempenho satisfatório apenas nas menores instâncias, perdendo competitividade à medida que n aumenta — consequência direta da menor resiliência à perda de diversidade e da maior tendência à estagnação populacional.

De forma geral, os achados reforçam que políticas de substituição não devem ser tratadas como meros detalhes implementacionais, mas sim como componentes estruturais da dinâmica evolutiva. O controle por idade, com seu processo de renovação gradual e gerenciamento adaptativo da longevidade dos indivíduos, destaca-se como uma alternativa promissora e flexível para problemas combinatórios desafiadores.

6.2. Limitações do Estudo

Apesar dos resultados consistentes, algumas limitações merecem destaque:

- **Domínio único de avaliação:** o estudo concentrou-se exclusivamente no problema das n -rainhas. Embora seja um benchmark consolidado, suas particularidades podem favorecer ou desfavorecer determinadas estratégias.
- **Parâmetros fixos:** taxas de mutação, tamanho populacional, pressão seletiva e demais parâmetros foram mantidos constantes. Diferentes configurações podem modificar substancialmente o desempenho relativo das estratégias.
- **Execução sequencial:** todos os experimentos foram executados de forma não paralela. Estratégias que envolvem populações maiores ou mais dinâmicas poderiam se beneficiar de paralelização.
- **Análise estatística limitada:** ainda que médias e tendências tenham sido exploradas, análises estatísticas formais — como testes de hipótese ou estimativas de variabilidade — não foram aprofundadas.

Tais limitações não comprometem as conclusões centrais, mas apontam oportunidades relevantes para estudos complementares.

6.3. Trabalhos Futuros

Com base nas conclusões e limitações, diversas direções promissoras podem ser exploradas:

- **Avaliar a generalidade das estratégias:** aplicar os mesmos métodos a outros problemas combinatórios, como Caixeiro Viajante, Sudoku, timetabling ou funções de otimização contínua.
- **Aprofundar variantes do GAVaPS:** investigar ajustes adaptativos de idade, mecanismos avançados de reprodução e políticas dinâmicas de variação do tamanho populacional.
- **Adotar análises estatísticas mais robustas:** incorporar ANOVA, testes não paramétricos, bootstrap ou intervalos de confiança para validar formalmente as diferenças entre estratégias.
- **Otimizar desempenho:** integrar paralelização via GPU, Numba, multiprocessing ou frameworks distribuídos, permitindo testar populações maiores e limites superiores de avaliação.
- **Explorar estratégias híbridas:** combinar controle por idade com elitismo, mutação adaptativa ou mecanismos explícitos de preservação de diversidade.
- **Criar um painel visual interativo:** desenvolver ferramentas que permitam acompanhar, em tempo real, a evolução das populações, curvas de aptidão e métricas auxiliares.

Essas direções podem ampliar a compreensão sobre a dinâmica evolutiva dos algoritmos genéticos e fortalecer a aplicabilidade prática das estratégias investigadas.

Referências

- Ali, H. et al. (2023). Performance comparison of genetic algorithms with traditional search techniques on the n-queen problem. In *Proceedings of ICIT*.
- Bell, J. and Stevens, B. (2009). A survey of known results and research areas for n-queens. *Discrete Mathematics*, 309(1):1–31.
- De Jong, K. A. (1975). *An Analysis of the Behavior of a Class of Genetic Adaptive Systems*. PhD thesis, University of Michigan. Ph.D. thesis, referência fundacional dos GAs.
- de Lacerda, E. G. M. and de Leon Ferreira de Carvalho, A. C. P. (1999). Introdução aos algoritmos genéticos. In *Sistemas inteligentes: aplicações a recursos hídricos e sistemas ambientais*. Ed. Universidade / UFRGS / ABRH, Porto Alegre.
- Eiben, A. E. and Smith, J. E. (2015). *Introduction to Evolutionary Computing*. Natural Computing Series. Springer, Berlin Heidelberg, 2 edition.
- Farhan, A. S., Tareq, W. Z., and Awad, F. H. (2015). Solving n queen problem using genetic algorithm. *International Journal of Computer Applications*, 122(12).
- Goldberg, D. E. (1989). *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley.
- Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., et al. (2020). Array programming with NumPy. *Nature*, 585(7825):357–362.
- Holland, J. H. (1975). *Adaptation in Natural and Artificial Systems*. University of Michigan Press.
- Hunter, J. D. (2007). Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, 9(3):90–95.
- Michalewicz, Z. (1994). Genetic algorithms with varying population size. In *Proceedings of the First IEEE Conference on Evolutionary Computation*, volume 1, pages 115–119. IEEE.
- Mitchell, M. (1996). *An Introduction to Genetic Algorithms*. MIT Press.
- Python Software Foundation (2025a). *The Python Language Reference, version 3.x*. Python Software Foundation. Acesso em: 28 jun. 2025.
- Python Software Foundation (2025b). *Python Module Reference: random — generate pseudo-random numbers*. Python Software Foundation, versão 3.13.5 edition. Acesso em: 28 jun. 2025.
- Richter, H. and Berlezi, R. (2012). Algoritmo genético aplicado ao problema das n rainhas. *Anais da UNIJUÍ*. Relatório técnico disponível em proceedings.sbmac.
- Sarkar, U. and Nag, S. (2017). An adaptive genetic algorithm for solving n-queens problem. *arXiv preprint*.
- Spears, W. M. (1993). Crossover or mutation? In Whitley, D., editor, *Foundations of Genetic Algorithms 2*, pages 221–237. Morgan Kaufmann.
- Whitley, D. (1994). A genetic algorithm tutorial. *Statistics and Computing*, 4(2):65–85.