



**INSTITUTO
FEDERAL**
Paraíba

Instituto Federal de Educação, Ciência e Tecnologia da Paraíba

Campus João Pessoa

Programa de Pós-Graduação em Tecnologia da Informação

Nível Mestrado Profissional

RICHARD SIDNEY DE SOUZA E SILVA

**OTIMIZAÇÃO DO MODELO DE ESTEGANOGRAFIA LSB
EM IMAGENS UTILIZANDO ALGORITMOS
BIOINSPIRADOS**

DISSERTAÇÃO DE MESTRADO

JOÃO PESSOA

2026

Richard Sidney de Souza e Silva

**Otimização do Modelo de Esteganografia LSB em imagens
utilizando Algoritmos Bioinspirados**

Dissertação apresentada como requisito parcial
para obtenção do título de Mestre em Tecno-
logia da Informação, pelo Programa de Pós-
Graduação em Tecnologia da Informação do
Instituto Federal de Educação, Ciência e Tec-
nologia da Paraíba – IFPB.

Orientador: Prof. Dr. Thiago Gouveia

João Pessoa

2026

Dados Internacionais de Catalogação na Publicação (CIP)
Biblioteca Nilo Peçanha - *Campus* João Pessoa, PB.

S586o Silva, Richard Sidney de Souza e.

Otimização do modelo de esteganografia LSB em imagens
utilizando algoritmos bioinspirados / Richard Sidney de Souza e
Silva. – 2026.

82 f. : il.

Dissertação (Mestrado em Tecnologia da Informação) – Institu-
to Federal de Educação da Paraíba / Programa de Pós-Graduação
em Tecnologia da Informação (PPGTI), 2026.

Orientação: Profº Dr. Thiago Gouveia.

1. Esteganografia LSB. 2. Otimização de colônia de formigas.
3. Algoritmo genético. 4. Meta-heurística. 5. Segurança digital. I.
Título.

CDU 004.056 (043)

RICHARD SIDNEY DE SOUZA E SILVA

**OTIMIZAÇÃO DO MODELO DE ESTEGANOGRAFIA LSB EM IMAGENS UTILIZANDO ALGORITMOS
BIOINSPIRADOS**

DISSERTAÇÃO apresentada como requisito para obtenção do título de Mestre em Tecnologia da Informação, pelo Programa de Pós-Graduação em Tecnologia da Informação do Instituto Federal de Educação, Ciência e Tecnologia da Paraíba - IFPB - Campus João Pessoa.

Aprovado em 06 de março de 2026.

Membros da Banca Examinadora:

Prof. Dr. Thiago Gouveia da Silva
Instituto Federal da Paraíba (IFPB)
Orientador

Prof. Dr. Diego Ernesto Rosa Pessoa
Instituto Federal da Paraíba (IFPB)
Examinador

Prof. Dr. Gilberto Farias de Sousa Filho
Instituto Federal da Paraíba (UFPB)
Examinador

Documento assinado eletronicamente por:

- **Thiago Gouveia da Silva**, PROFESSOR ENS BASICO TECN TECNOLOGICO, em 07/03/2026 08:51:06.
- **Gilberto Farias de Sousa Filho**, PROFESSOR DE ENSINO SUPERIOR NA ÁREA DE ORIENTAÇÃO EDUCACIONAL, em 07/03/2026 10:44:21.
- **Diego Ernesto Rosa Pessoa**, PROFESSOR ENS BASICO TECN TECNOLOGICO, em 07/03/2026 15:39:18.

Este documento foi emitido pelo SUAP em 20/02/2026. Para comprovar sua autenticidade, faça a leitura do QRCode ao lado ou acesse <https://suap.ifpb.edu.br/autenticar-documento/> e forneça os dados abaixo:

Código 837796
Verificador: cbc4300e9e
Código de Autenticação:



*Este trabalho é dedicado aos pesquisadores do Brasil,
que mesmo diante de um cenário desafiador,
mantém firme a dedicação à pesquisa científica.*

AGRADECIMENTOS

Agradeço primeiramente ao meu orientador, Prof. Dr. Thiago Gouveia, pela condução segura e perspicaz deste trabalho. Sua paciência, disponibilidade e profundo conhecimento técnico foram fundamentais para que eu pudesse superar os desafios desta pesquisa. Obrigado pela confiança depositada em mim e por me guiar nos momentos obscuros com todo o rigor científico necessário.

À Fundação de Apoio à Pesquisa do Estado da Paraíba (FAPESQ), expresso minha gratidão pelo fomento concedido. O apoio financeiro foi indispensável para o desenvolvimento dos recursos necessários à conclusão deste estudo.

Aos meus familiares, que sempre foram meu alicerce. Agradeço pelo incentivo constante, pelo carinho incondicional e por acreditarem no meu potencial, mesmo nos momentos mais difíceis.

Aos meus amigos, pela compreensão durante as minhas ausências, pelas palavras de ânimo e pelos momentos de descontração que tornaram essa jornada mais leve. O apoio de vocês foi essencial para manter o equilíbrio e a motivação. Obrigado!

RESUMO

A cibersegurança ocupa um papel central na tecnologia atual, com destaque para o tráfego seguro de arquivos digitais, contexto no qual a esteganografia atua como ferramenta fundamental para o compartilhamento oculto de informações em imagens. Este trabalho propõe uma nova abordagem para a substituição LSB (*Least Significant Bit*) baseada na inserção não sequencial de bits, modelando o processo como um problema de otimização para identificar a combinação ideal entre imagem de cobertura e estratégia de inserção. Para tal, foram aplicados os algoritmos DFS (*Depth-First Search*), Guloso, ACO (*Ant Colony Optimization*) e GA (*Genetic Algorithm*). Os resultados demonstraram que o modelo proposto, aliado às meta-heurísticas (ACO e GA), apresentou desempenho promissor frente à esteganálise, superando o LSB convencional nas métricas de diferença de intensidade, PSNR e SSIM. A utilização de algoritmos bioinspirados otimizou significativamente o tempo de busca pela melhor estratégia. Por fim, a estrutura de árvore de *bitsets* desenvolvida estabelece uma base para avanços na esteganografia LSB, permitindo que trabalhos futuros explorem novos métodos de busca e aprimorem a geração de soluções.

Palavras-chaves: Esteganografia LSB, Otimização de Colônia de Formigas, Algoritmos Genéticos, Meta-heurísticas.

ABSTRACT

Cybersecurity is central to modern technology, especially in secure digital file transfer, where steganography serves as a key tool for covert communication in images. This study proposes a novel Least Significant Bit (LSB) substitution approach utilizing non-sequential bit insertion. The process is modeled as an optimization problem to determine the optimal pair of cover image and insertion strategy. We applied Depth-First Search (DFS), Greedy, Ant Colony Optimization (ACO), and Genetic Algorithm (GA) to address this challenge. Results indicate that the proposed model, particularly when paired with meta-heuristics (ACO and GA), exhibits promising resistance against steganalysis and surpasses conventional LSB methods in intensity difference, PSNR, and SSIM metrics. Furthermore, bio-inspired algorithms significantly reduced the search time for the optimal strategy. The developed bitset tree structure provides a robust baseline for future LSB steganography advancements, facilitating the exploration of new search methods and solution generation techniques.

Key-words: LSB Steganography, Ant Colony Optimization, Genetic Algorithm, Metaheuristics.

LISTA DE FIGURAS

Figura 1 – Exemplo de Esteganografia em arquivos de Texto	19
Figura 2 – Representação digital de uma imagem	20
Figura 3 – Decomposição de uma imagem utilizando o sistema RGB	20
Figura 4 – Método de substituição LSB	21
Figura 5 – Exemplo de uma abordagem de ataque visual	23
Figura 6 – Exemplo de algoritmo DFS na busca de um ponto em um grafo	25
Figura 7 – Comportamento de uma Colônia de Formigas reais	26
Figura 8 – Histórico de produção acadêmica	29
Figura 9 – Publicações filtradas por base	30
Figura 10 – Resultado quantitativo da busca por base	30
Figura 11 – Tendência de produção acadêmica ao longo dos anos	31
Figura 12 – Tendência de produção acadêmica envolvendo Otimização e Esteganografia	32
Figura 13 – Método proposto	39
Figura 14 – Matrizes LSB.	40
Figura 15 – Operação de criação do <i>bitset</i>	42
Figura 16 – Montagem do <i>array</i> de <i>bitsets</i>	43
Figura 17 – Representação como árvore de busca	43
Figura 18 – Implementação da GA proposta	49
Figura 19 – Imagem esteganográfica utilizando LSB comum	53
Figura 20 – Imagem esteganográfica utilizando o modelo LSB proposto	53
Figura 21 – Recortes das imagens submetidas a esteganálise	54
Figura 22 – Gráficos da intensidade da diferença de píxeis	54
Figura 23 – Gráficos da intensidade da diferença de píxeis do segundo experimento . . .	54
Figura 24 – Gráfico da intensidade da diferença de píxeis na imagem gerada pelo LSB proposto	55
Figura 25 – Exposição da informação inserida nas imagens LSB geradas a partir da Mensagem de Teste 1	57
Figura 26 – Exposição da informação inserida nas imagens LSB geradas a partir da Mensagem de Teste 2	57
Figura 27 – Exposição da informação inserida nas imagens LSB geradas a partir da Mensagem de teste 3	58
Figura 28 – Imagem LSB Proposta após esteganálise aural	58
Figura 29 – Imagem original após esteganálise aural	59
Figura 30 – Imagem LSB Proposta após esteganálise aural	59
Figura 31 – Imagem original após esteganálise aural	59
Figura 32 – Imagem LSB Proposta após esteganálise aural	60

Figura 33 – Imagem original após esteganálise aural	60
Figura 34 – Gráfico da intensidade da diferença de pixels na imagem do modelo LSB comum (Teste com a mensagem 1)	60
Figura 35 – Gráfico da intensidade da diferença de pixels na imagem do modelo LSB proposto (Teste com a mensagem 1)	61
Figura 36 – Gráfico da intensidade da diferença de pixels na imagem do modelo LSB comum (Teste com a mensagem 2)	61
Figura 37 – Gráfico da intensidade da diferença de pixels na imagem do modelo LSB proposto (Teste com a mensagem 2)	62
Figura 38 – Gráfico da intensidade da diferença de pixels na imagem do modelo LSB comum (Teste com a mensagem 3)	62
Figura 39 – Gráfico da intensidade da diferença de pixels na imagem do modelo LSB proposto (Teste com a mensagem 3)	63
Figura 40 – Curva de Convergência Média da parametrização do ACO sem evaporação utilizando 100 Formigas a cada iteração	66
Figura 41 – Curva de Convergência Média da parametrização do ACO sem evaporação utilizando 1000 Formigas a cada iteração	67
Figura 42 – Curva de Convergência Média da parametrização do ACO com evaporação utilizando 100 Formigas a cada iteração	70
Figura 43 – Curva de Convergência Média da parametrização do ACO com evaporação utilizando 1000 Formigas a cada iteração	71
Figura 44 – Curva de Convergência Média do Grupo de Configurações do GA	73
Figura 45 – Curva de Convergência Média do GA com a melhor parametrização no experimento com 25 milhões de possibilidades ao longo de 5 mil gerações .	74
Figura 46 – Comparação de eficiência (%)	76

LISTA DE TABELAS

Tabela 1 – Artigos selecionados da base IEEE Xplore	33
Tabela 2 – Artigos selecionados da base Web of Science (WoS)	34
Tabela 3 – Artigos selecionados da base Scopus	35
Tabela 4 – Parâmetros de configuração para o Experimento Inicial.	41
Tabela 5 – Parâmetros de configuração para o Modelo Final.	42
Tabela 6 – Valores de PSNR comparados entre o LSB Proposto e o LSB padrão.	64
Tabela 7 – Valores de SSIM comparados entre o LSB Proposto e o LSB padrão.	64
Tabela 8 – Resultados dos Algoritmos de Busca.	64
Tabela 9 – Resultados da parametrização do ACO sem Evaporação (destaque para a melhor configuração).	66
Tabela 10 – Resultados da parametrização do ACO com Evaporação, comparando o desempenho entre iterações utilizando 100 e 1000 formigas.	69
Tabela 11 – Detalhamento da evolução do Algoritmo Genético, comparando Custo e Tempo ao longo das 5000 gerações,	72
Tabela 12 – Resultados final dos algoritmos inspirados na ACO, comparando o desempenho no experimento com 25 milhões de possibilidades utilizando 100 ciclos.	75
Tabela 13 – Resultado final da melhor hiperparametrização do algoritmo inspirado no GA ao longo de 5 mil gerações.	75
Tabela 14 – Comparativo de perda de qualidade e ganho de velocidade em relação ao método de Força Bruta	75

SUMÁRIO

1	INTRODUÇÃO	14
1.1	Motivação e Definição do Problema	15
1.2	Objetivos	16
1.3	Estrutura do Documento	16
2	FUNDAMENTAÇÃO TEÓRICA	18
2.1	Esteganografia	18
2.1.1	Esteganografia em Imagens	19
2.1.2	Esteganografia LSB	21
2.1.3	Esteganálise	22
2.1.4	Ataque Aural	22
2.2	Algoritmos de Busca e Otimização	23
2.2.1	Busca em Profundidade	24
2.2.2	Algoritmos Gulosos	24
2.2.3	Otimização de Colônia de Formigas (ACO)	25
2.2.4	Algoritmo Genético	27
2.3	Trabalhos Relacionados	28
2.3.1	Análise Bibliométrica	28
2.3.2	Base IEEE Xplore	32
2.3.3	Base Web of Science (WoS)	33
2.3.4	Base Scopus	34
2.3.5	Outros Trabalhos Seleccionados	37
3	MÉTODO PROPOSTO	39
3.1	Método	39
3.2	Modelo de Esteganografia LSB	40
3.2.1	Árvore de busca	43
3.3	Busca e Otimização	44
3.3.1	Algoritmos DFS e Busca Gulosa	44
3.3.2	Meta-heurística ACO	45
3.3.3	Meta-heurística ACO com Evaporação	46
3.3.4	Meta-heurística GA	48
4	RESULTADOS	51
4.1	Modelo LSB Inicial	51
4.2	Modelo LSB Final	55

4.3	Medição Objetiva	63
4.4	Algoritmos de Busca e Otimização	64
5	CONSIDERAÇÕES FINAIS	77
	REFERÊNCIAS BIBLIOGRÁFICAS	79

1 INTRODUÇÃO

O uso massivo de arquivos digitais para compartilhamento de informações na rotina da sociedade atual traz à tona a grande necessidade de evolução nos métodos de segurança digital para garantir a proteção e a confidencialidade dos dados. Esses arquivos podem representar informações compartilhadas em redes sociais, dados bancários, informações pessoais ou sigilosas de grandes empresas e governos. Conforme destacam Li et al. (2017), o rápido desenvolvimento da ciência da computação e do *Big Data* exercem profunda influência na comunicação, tornando a transmissão segura um foco de atenção extensiva para evitar roubos de identidade ou perdas irreparáveis. Dessa forma, pode-se apontar a Segurança da Informação como um dos principais fatores que pautam o uso atual da tecnologia (G et al., 2021).

Neste contexto, destaca-se a criptografia, técnica fundamental para assegurar o sigilo da comunicação, lançando mão de manipulações matemáticas para impedir que invasores compreendam o conteúdo (LI et al., 2017). Entretanto, em determinadas circunstâncias, apenas cifrar os dados pode não ser suficiente, pois a existência de uma mensagem criptografada pode despertar a suspeita de atacantes (G et al., 2021). É necessário, portanto, não apenas ocultar o significado da mensagem, mas também a sua existência. Dessa forma, pode-se elencar a esteganografia como uma ferramenta essencial para suprir essa necessidade, oferecendo uma camada adicional de segurança onde os métodos tradicionais isolados podem falhar (IBANEZ; DJAMAL; NAJMURROKHMAM, 2018).

A esteganografia pode ser definida como a arte e a ciência da comunicação segura, cujo objetivo é enviar informações ocultas de forma confiável e secreta, utilizando mídias como imagens, texto, áudio e vídeo como cobertura. Em um âmbito etimológico, o termo deriva da junção das palavras gregas *stegos* ("cobertura") e *grafia* ("escrita"), traduzindo-se como "escrita coberta" (SINGHAL; BEDI, 2015). É considerada uma subdisciplina vital no domínio da segurança da comunicação de dados.

Embora seu conceito seja antigo, a esteganografia como ferramenta digital tem sido amplamente explorada e diversas técnicas vêm sendo desenvolvidas com o objetivo de ocultar informações em arquivos digitais (SINGHAL; BEDI, 2015). A inovação em técnicas de esteganografia em imagens digitais, especificamente a otimização de métodos como a substituição LSB (*Least Significant Bit*) para dificultar a esteganálise, possui um grande enfoque na literatura recente e este será o tema abordado neste trabalho (IBANEZ; DJAMAL; NAJMURROKHMAM, 2018).

1.1 Motivação e Definição do Problema

A esteganografia é fundamental para estratégias robustas de segurança digital e oferece uma transmissão "oculta" de informações entre dois nós em uma rede, cuja existência é desconhecida para o atacante ou qualquer interceptador (WALIA et al., 2018). Esse tópico desperta o interesse não apenas de instituições de segurança e inteligência, mas também de civis e criminosos que desejam burlar controles ou regulamentos. Técnicas foram utilizadas, por exemplo, em crimes cibernéticos, como o compartilhamento de pornografia infantil e fraudes financeiras, assim como pelos terroristas responsáveis pelo ataque às Torres Gêmeas, nos Estados Unidos (GARCIA, 2018).

Enquanto tema de estudo, a esteganografia figura em pesquisas voltadas à garantia de segurança em aplicações de mensagens instantâneas, amplamente utilizadas para comunicação via texto ou chamadas (IBANEZ; DJAMAL; NAJMURROKHMAN, 2018). O tema também é central no desenvolvimento de métodos eficazes de ocultação de dados em imagens médicas, visando assegurar a confidencialidade de diagnósticos e registros (HUSSAIN; KHODHER, 2024), bem como em estudos que buscam ampliar o uso seguro de materiais digitais no contexto educacional (HUSSEIN; ALHIJAJ, 2023).

Tal cenário demonstra como as pesquisas atuais têm concentrado esforços no desenvolvimento de novas técnicas de esteganografia capazes de mascarar informações em arquivos de imagem, vídeo e áudio (KHALIL; SARHAN; ALSHEWIMY, 2024). Frequentemente, o tema associa-se a conceitos de outras áreas, como aprendizado profundo (*Deep Learning*) e otimização de algoritmos. O objetivo é desenvolver modelos híbridos que combinem essas inovações com as técnicas clássicas, buscando maior robustez e melhor desempenho diante das principais técnicas de esteganálise.

Dentre as abordagens mais difundidas na esteganografia em imagens, destaca-se o método LSB (*Least Significant Bit*) ou Substituição do Bit Menos Significativo. Esta estratégia, quando aplicada a imagens digitais, consiste na alteração do último bit de cada um dos bytes que compõem um pixel (geralmente três, no padrão RGB), o que não afeta a percepção visual da imagem a olho nu. Embora sua implementação seja simples e eficaz, a técnica apresenta limitações e é vulnerável a sistemas de detecção baseados em esteganálise moderna.

As principais limitações da substituição LSB residem na restrição do volume de dados inseridos (carga útil) e na suscetibilidade à detecção por meio de modelos de esteganálise, que exploram alterações estatísticas em atributos como cor e saturação. Além disso, a técnica apresenta fragilidade quando aplicada a imagens de cobertura com predominância de áreas claras ou uniformes (BOUGHACI; KEMOUCHE; LACHIBI, 2016).

Diante desses desafios, autores como Walia et al. (2018) e Banharnsakun (2018) propõem alternativas para mitigar as falhas da substituição LSB, desenvolvendo métodos que otimizam a inserção de dados para alterar o mínimo possível a imagem original. Tais abordagens podem ser

formuladas como problemas de otimização.

Nesse contexto, este trabalho propõe a implementação de um sistema de esteganografia baseado na técnica de substituição LSB. A abordagem desenvolvida concentra-se na busca simultânea pela imagem de cobertura ideal e pela melhor estratégia de inserção dos dados. O objetivo é selecionar, dentre um conjunto de candidatas, a combinação de imagem e estratégia capaz de gerar o resultado mais fiel à original (menor distorção). Para viabilizar essa seleção, o conjunto de soluções possíveis é estruturado como uma Árvore de Busca, representação proposta nesta pesquisa. A utilização de algoritmos de otimização torna-se, portanto, primordial para a exploração eficiente dessa estrutura, tendo sido desenvolvidas e analisadas diferentes implementações ao longo deste estudo.

1.2 Objetivos

Desenvolver e implementar abordagens otimizadas de esteganografia baseadas na técnica de substituição LSB em imagens digitais. Por sua vez, para atingir esta finalidade, seguem os objetivos específicos.

- Investigar a fundamentação teórica e as limitações da técnica de substituição LSB tradicional em imagens digitais.
- Formular o processo de esteganografia LSB como um problema de otimização combinatória, estruturando o espaço de soluções (Árvore de Busca).
- Avaliar algoritmos de otimização e meta-heurísticas adequados para a exploração eficiente do modelo proposto.
- Implementar algoritmos de otimização para selecionar a imagem de cobertura e a estratégia de inserção ideais.
- Validar a eficácia do modelo proposto por meio de métricas de desempenho e esteganálise, comparando os resultados obtidos com a técnica LSB padrão.

1.3 Estrutura do Documento

Este documento está dividido em cinco capítulos, incluindo esta introdução. O restante do trabalho está organizado da seguinte forma:

- O Capítulo 2 apresenta a fundamentação teórica necessária à compreensão do tema, abordando conceitos de esteganografia, processamento de imagens digitais e algoritmos de otimização, além de realizar uma revisão dos trabalhos relacionados encontrados na literatura.

- O Capítulo 3 descreve a metodologia adotada e detalha o modelo proposto. Apresenta, ainda, o delineamento dos experimentos computacionais configurados para validar a abordagem.
- O Capítulo 4 compila e analisa os dados obtidos a partir dos experimentos realizados, discutindo o desempenho do modelo frente às métricas consagradas na literatura.
- O Capítulo 5 sintetiza as principais descobertas da pesquisa e apresenta as considerações finais, destacando as contribuições do trabalho, suas limitações e perspectivas para estudos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo apresenta a fundamentação teórica que embasa o desenvolvimento deste trabalho. A Seção 2.1 discute os conceitos e técnicas da esteganografia, com ênfase no método de substituição LSB. A Seção 2.2 aborda os fundamentos dos algoritmos de busca e otimização utilizados no modelo proposto, destacando duas meta-heurísticas bioinspiradas. Por fim, a Seção 2.3 detalha a revisão da literatura, apresentando a análise bibliográfica realizada e os principais estudos relacionados a esta pesquisa.

2.1 Esteganografia

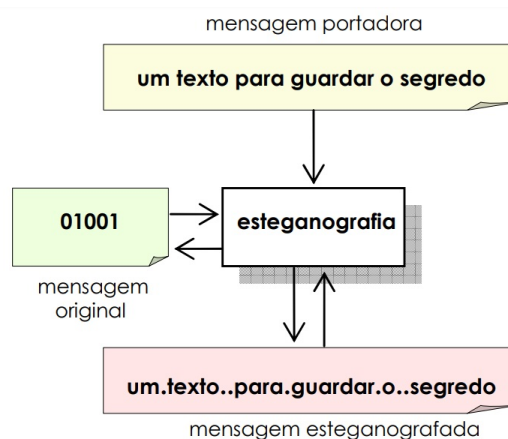
O conceito de camuflar uma mensagem em outra a fim de mascarar seu verdadeiro sentido é antigo. Seus primeiros registros remontam à Grécia Antiga e ao Império Persa, onde a esteganografia era utilizada para comunicação oculta durante conflitos. Contudo, os primeiros livros a tratar formalmente do tema foram escritos séculos mais tarde. O pioneiro foi a obra *Steganographia*, escrita por Johannes Trithemius em 1499. Posteriormente, o filósofo Francis Bacon abordou o tema no século XVII, descrevendo seu método esteganográfico que viria a ser conhecido como Código Bacon.

A utilização da esteganografia como ferramenta de comunicação militar foi identificada nos conflitos do século XX, com a técnica de micropontos, utilizada pelos alemães. Essa aplicação consistia na redução de textos ou imagens a dimensões minúsculas (cerca de 1 mm de diâmetro) para evitar a detecção por destinatários indesejados (HASSAN; HIJAZI, 2017). Na atualidade, atribui-se também o uso dessa ferramenta a atividades ilícitas, como no vazamento de informações sigilosas de empresas e na comunicação entre grupos criminosos, incluindo o compartilhamento de pornografia infantil (PELOSI et al., 2018).

Apesar da presença em atividades maliciosas, a esteganografia é primordialmente uma ferramenta de segurança cibernética. Sua relevância justifica-se pela capacidade de os algoritmos esteganográficos proverem camadas adicionais de confidencialidade, integridade e autenticação aos dados. Nesse sentido, Kumbhakar, Sanyal e Karforma (2023) destaca aplicações legítimas ligadas à segurança digital, como distribuição segura de conteúdo em transações eletrônicas, troca de chaves de acesso e proteção contra interceptação de dados entre origem e destino.

São numerosas as aplicações de segurança da informação que empregam a esteganografia, e estudos voltados à avaliação da robustez dessas técnicas têm mostrado resultados promissores. Diversos tipos de arquivos digitais podem ser utilizados como portadores de informações ocultas, incluindo áudio, texto e imagens. A Figura 1, por exemplo, apresenta o fluxograma de um algoritmo de esteganografia aplicado a textos.

Figura 1 – Exemplo de Esteganografia em arquivos de Texto



Fonte: Jasper e Pereira (2009).

No entanto, embora qualquer mídia digital seja adequada para a esteganografia, as mais amplamente utilizadas são vídeos e imagens, pois contêm uma alta taxa de redundância (repetição), o que faz com que os dados inseridos causem menos distorções perceptíveis (HUSSEIN; ALHIJAJ, 2023). Devido a essa prevalência na transmissão de dados, a inovação em técnicas esteganográficas voltadas para imagens é um tema amplamente explorado na literatura especializada.

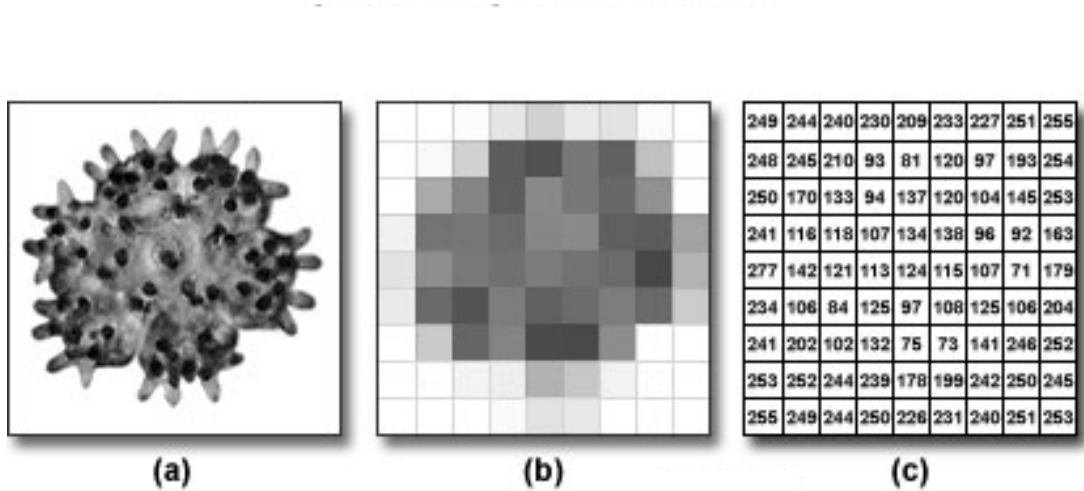
2.1.1 Esteganografia em Imagens

A representação digital de uma imagem é uma coleção de números que constituem diferentes intensidades de luz em diferentes áreas da imagem. Em essência, a representação digital de uma imagem é comumente formada pela sobreposição de matrizes que definem a intensidade de diferentes luzes em cada área de uma grade. A unidade básica desse mapa é denominada pixel. Na Figura 2, encontra-se um esquema que representa a decomposição de uma imagem digital em pixels.

Os pixels constituem a unidade fundamental de uma imagem digital, encarregados de codificar informações cromáticas e de luminosidade. A nitidez da imagem está diretamente relacionada à sua resolução espacial, frequentemente medida em pixels por polegada (PPI). Paralelamente, a fidelidade das cores depende da profundidade de bits, que dita a quantidade de tons distintos que podem ser representados.

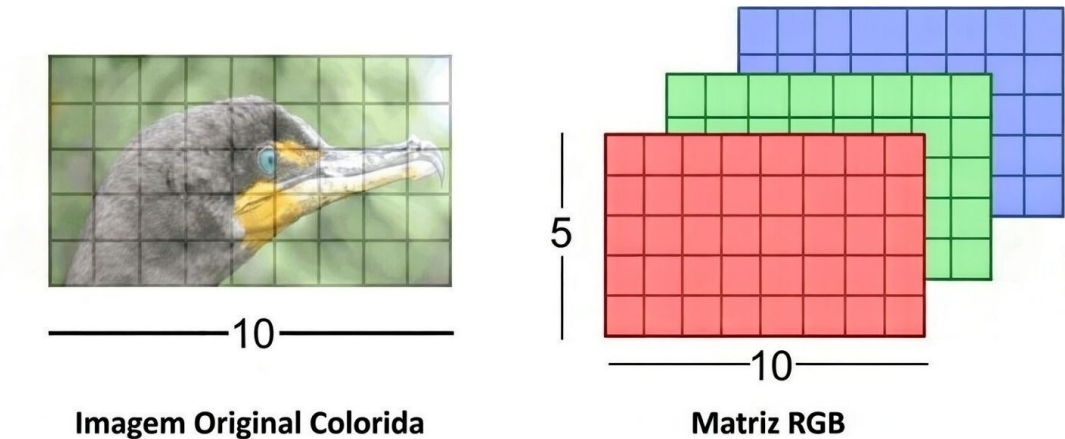
As imagens digitais coloridas são construídas utilizando sistemas de cores básicas. Comumente, o sistema utilizado para tal representação é o sistema aditivo *Red Green Blue* (RGB). Este é formado por valores numéricos, os quais representam a quantidade de luz vermelha, verde e azul que será sobreposta em um único pixel. Na Figura 3, está representada a decomposição de uma imagem nas três matrizes de intensidade de cor, matriz vermelha, verde e azul, como definido no sistema RGB.

Figura 2 – Representação digital de uma imagem



Fonte: Adaptado de Florida State University (s.d.).

Figura 3 – Decomposição de uma imagem utilizando o sistema RGB



Fonte: Adaptado de Singh (2015).

Tendo em vista que normalmente são dispostos 8 *bits* para a representação de cada cor primária, normalmente cada pixel é composto por três valores no intervalo entre 0 e 255. Neste contexto, existem mais de 16 milhões de cores que podem ser reproduzidas.

Além da representação binária, outro ponto importante quando se trata de imagens digitais são os métodos de compressão utilizados. Essas técnicas recorrem à fórmulas matemáticas para analisar e condensar dados de imagem, resultando em tamanhos de arquivo menores, o que facilita a transmissão dessas imagens.

Ambos os aspectos, a representação binária e a técnica de compressão, são fatores determinantes na escolha do método esteganográfico. A partir dessas características, há uma variedade de técnicas disponíveis na literatura, as quais podem ser classificadas, primordialmente, em abordagens que manipulam o domínio espacial e o domínio da frequência (OLEWI et al., 2024).

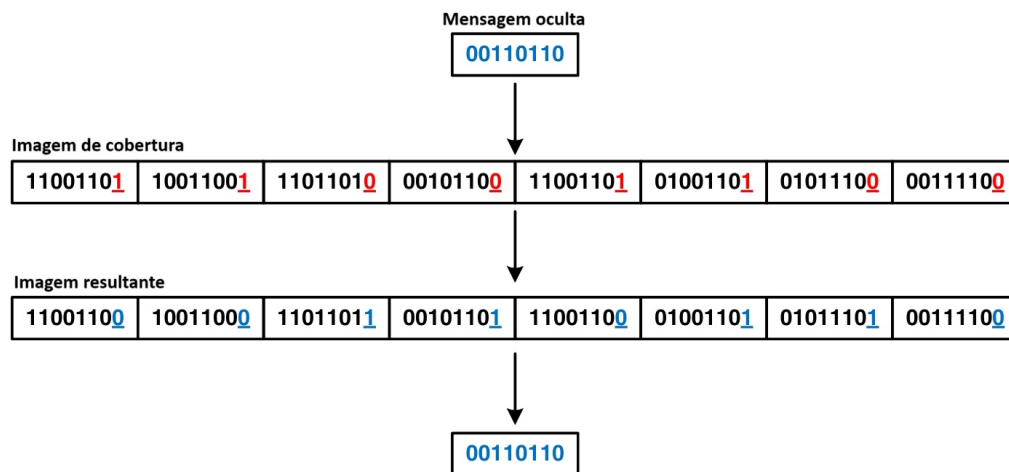
Ao projetar uma técnica esteganográfica para imagens digitais, deve-se considerar o equilíbrio entre três aspectos fundamentais: a capacidade de carga (*payload*), referente à quantidade de bits disponíveis para ocultação; a imperceptibilidade, que assegura que a imagem resultante seja visual e estatisticamente indistinguível da original; e a robustez, que define a capacidade da mensagem oculta resistir a ataques, compressões ou processamentos sofridos pela imagem (KHALIL; SARHAN; ALSHEWIMY, 2024).

2.1.2 Esteganografia LSB

As técnicas de esteganografia em imagens digitais no domínio espacial fundamentam-se na manipulação direta das intensidades dos pixels da imagem hospedeira para ocultar a mensagem secreta. Dentre os métodos dessa categoria, destaca-se a técnica de substituição do bit menos significativo, conhecida como LSB (*Least Significant Bit*), sendo uma das mais difundidas na literatura.

Em uma implementação típica desse método, o algoritmo substitui os bits menos significativos dos pixels selecionados pelos bits que compõem a mensagem secreta. Esse processo visa garantir a inserção dos dados preservando a imperceptibilidade visual a olho nu (WALIA et al., 2018), conforme ilustrado na Figura 4.

Figura 4 – Método de substituição LSB



Fonte: Banharnsakun (2018).

Abordagens mais sofisticadas, que recorrem à técnicas de otimização ou ao aprendizado profundo (*Deep Learning*), permitem a inserção da informação em regiões adaptativas. Nesses casos, a seleção dos pixels considera as características da visão humana, explorando áreas onde pequenas distorções são estatisticamente ou visualmente toleráveis.

2.1.3 Esteganálise

A esteganálise é a ferramenta que surgiu em contraposição à esteganografia, sendo amplamente utilizada na Computação Forense para detectar a existência de conteúdo esteganográfico em dispositivos digitais e, eventualmente, para a interpretação do conteúdo oculto. Essa ferramenta pode ser definida como a ciência de detectar e interromper comunicações secretas. Seu requisito fundamental consiste em determinar, com precisão, se uma mensagem secreta reside no objeto de análise (imagem, vídeo ou mídia suspeita), utilizando técnicas estatísticas avançadas como a RS *Steganalysis* para identificar alterações nos padrões da mídia de cobertura (IBANEZ; DJAMAL; NAJMURROKHMAN, 2018).

Em geral, a esteganálise concentra-se na verificação da existência de informações ocultas, sem necessariamente ter o foco imediato na recuperação do conteúdo da mensagem; sua intenção é primordialmente a detecção. Contudo, modelos avançados podem ser desenvolvidos com o intuito de identificar o algoritmo esteganográfico utilizado, estimar o tamanho da mensagem oculta (*payload*) ou até mesmo realizar a sua remoção. A extração para leitura efetiva da mensagem é uma tarefa de maior complexidade e demanda, frequentemente, o conhecimento prévio da técnica de esteganografia empregada.

Com a evolução da área de processamento digital de sinais, novas técnicas esteganográficas emergiram, impulsionando concomitantemente o desenvolvimento de novas abordagens esteganalíticas. Dependendo do método de ataque empregado pelo examinador forense, as técnicas podem ser classificadas em diferentes categorias, como a esteganálise estatística, esteganálise específica (direcionada a um método conhecido), esteganálise cega (universal), estrutural e visual (Aural).

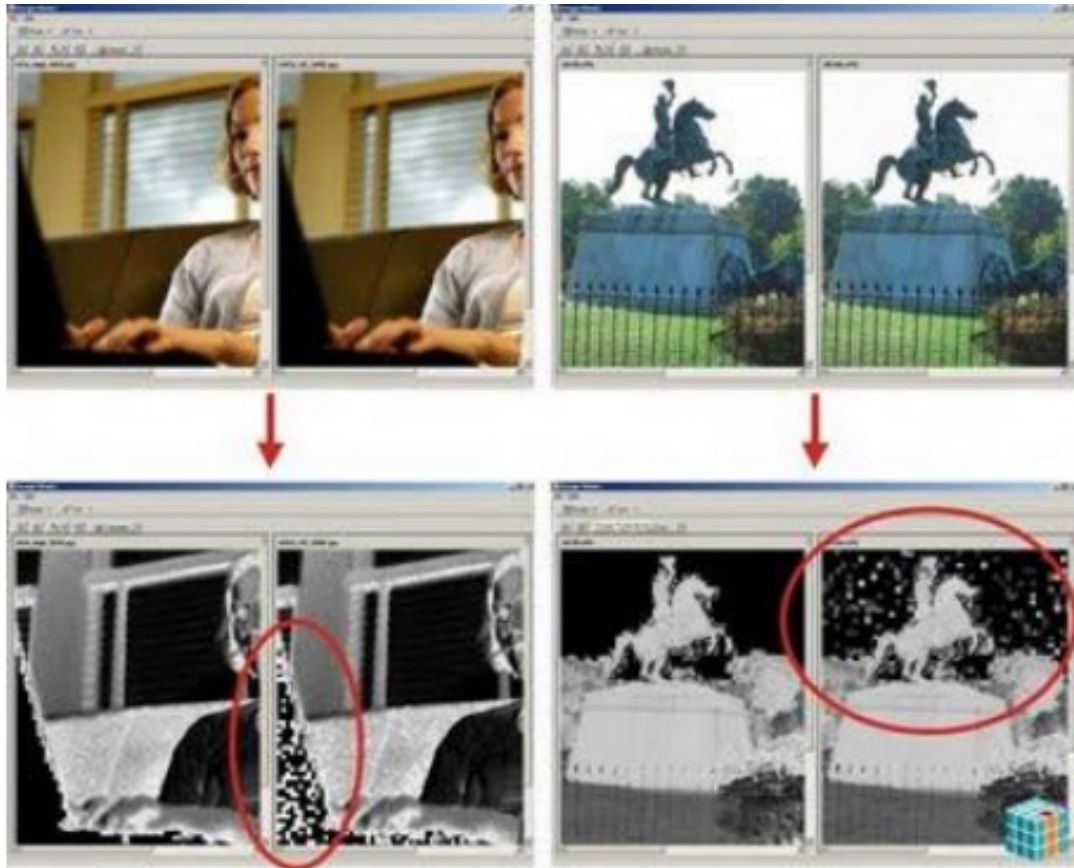
2.1.4 Ataque Aural

A esteganálise visual apresenta o conceito mais intuitivo dentre os métodos de esteganálise, consistindo na inspeção da imagem em busca de anomalias perceptíveis ou artefatos que indiquem a presença de dados ocultos. No entanto, atualmente, mesmo algoritmos simples são capazes de realizar a ocultação sem deixar vestígios que possam ser observados a olho nu.

No entanto, a aplicação de processamento digital ou filtros de realce pode evidenciar anomalias previamente imperceptíveis. Em métodos que manipulam diretamente os pixels de cor, alterações na paleta de cores podem revelar padrões artificiais, conforme observado na Figura 5. A técnica LSB é um exemplo clássico de método suscetível à detecção por meio de esteganálise visual aprimorada.

Normalmente, em imagens naturais que representam cenas complexas, a distribuição dos valores de bit 0 e 1 no plano LSB tende ao equilíbrio (aleatoriedade). Entretanto, quando um texto é convertido para o formato binário, observa-se frequentemente uma predominância de bits 0 em relação aos bits 1.

Figura 5 – Exemplo de uma abordagem de ataque visual



Fonte: SGURSKY (2015)

Essa discrepância gera uma inconsistência estatística, causada pela quebra da aleatoriedade na disposição dos bits, criando um padrão de alteração na imagem, o que auxilia o perito forense a classificar o arquivo como uma imagem alterada por esteganografia (KARAMPIDIS; KAVALLIERATOUA; PAPADOURAKIS, 2018).

2.2 Algoritmos de Busca e Otimização

No âmbito da Ciência da Computação, especialmente em Pesquisa Operacional e Inteligência Artificial, muitos desafios são formulados como problemas de otimização combinatória. A resolução desses problemas consiste, fundamentalmente, em navegar por um espaço de estados finito em busca da melhor solução possível. Uma das abordagens clássicas para realizar essa varredura de forma sistemática e exaustiva é o algoritmo de Busca em Profundidade, ou DFS (*Depth First Search*).

Contudo, é comum que problemas dessa classe apresentem um espaço de busca vasto. Nesses casos, a quantidade de soluções possíveis cresce exponencialmente em relação ao tamanho da entrada. Sob tal condição, abordagens de força bruta ou varreduras exaustivas (como o DFS puro) tornam-se computacionalmente impraticáveis, demandando tempo de execução proibitivo.

Aprofundando a discussão, é possível classificar os métodos de otimização em duas grandes categorias, as abordagens para problemas de soluções exatas e as abordagens para problemas de soluções aproximadas. As soluções aproximadas são indicadas para casos em que uma resposta próxima da ideal (subótima) é suficiente para a resolução do problema, dada a inviabilidade de se encontrar o ótimo global em tempo hábil. Nesses cenários, o algoritmo eficiente é aquele que estabelece um equilíbrio entre uma solução de boa qualidade e um baixo custo de processamento.

No âmbito das soluções aproximadas, distinguem-se duas subcategorias principais, os métodos heurísticos e os meta-heurísticos (DESALE et al., 2015). Como exemplo de abordagem heurística, cita-se a Busca Gulosa (*Greedy Search*). Já entre as meta-heurísticas, o presente trabalho enfoca dois algoritmos bioinspirados, a Otimização por Colônia de Formigas (*Ant Colony Optimization* - ACO) e os Algoritmos Genéticos (*Genetic Algorithms* - GA).

2.2.1 Busca em Profundidade

A importância da busca em profundidade (DFS) para algoritmos de grafos foi estabelecida na década de 70, em que os primeiros trabalhos na área mostraram como esse tipo de busca poderia ser usada para construir uma variedade de algoritmos eficientes para grafos, como descrito por (KING; LAUNCHBURY, 1995). Nesse tipo de busca, expande-se um estado inicial de um grafo gerando seus sucessores imediatos. Em cada passo subsequente, um dos sucessores gerados mais recentes é selecionado e expandido.

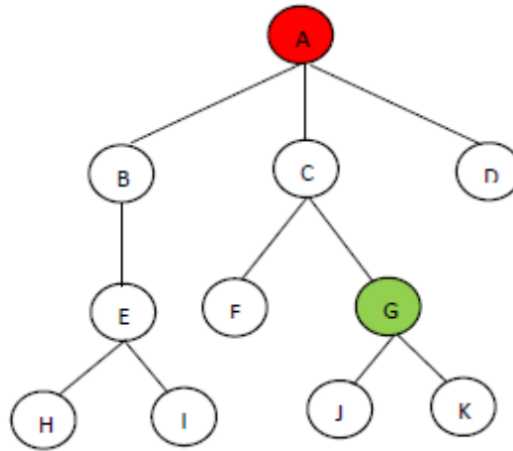
De forma simplificada, o algoritmo DFS inicia definindo todos os vértices do grafo como "não visitados". Em seguida, seleciona-se um vértice de partida e percorre-se uma aresta que leva a um vértice adjacente ainda não visitado. O processo repete-se recursivamente a partir deste novo ponto. A exploração continua em profundidade até que se atinja um vértice que não possui arestas para vizinhos não visitados. Nesse momento, realiza-se o retrocesso (*backtracking*) até o vértice anterior que ainda permita o acesso a novos caminhos inexplorados.

Utilizando como exemplo o grafo da Figura 6, partindo do vértice A e seguindo o procedimento descrito, chega-se, eventualmente, a um ponto em que todos os vértices alcançáveis a partir da origem foram visitados. Caso ainda restem vértices não visitados, seleciona-se um novo ponto de partida arbitrário e a busca é reiniciada. O processo repete-se até que, finalmente, a totalidade dos vértices tenha sido visitada e todas as arestas examinadas.

2.2.2 Algoritmos Gulosos

Em muitos cenários de aplicações de busca otimizada, encontrar a solução ótima global torna-se computacionalmente proibitiva ou, mesmo quando viável, o custo de processamento não justifica o retorno obtido. Nesses casos, a exigência de uma solução ótima estrita é relaxada em favor de um resultado satisfatório, priorizando-se a redução significativa do tempo de execução e do consumo de recursos.

Figura 6 – Exemplo de algoritmo DFS na busca de um ponto em um grafo



Fonte: Rachmawati, Efendi e Situmorang (2020).

Portanto, em cenários como este, busca-se encontrar soluções de alta qualidade com a maior eficiência possível, equilibrando a excelência do resultado com a otimização do tempo de resposta. Para atender a essa necessidade, são frequentemente utilizadas abordagens heurísticas, dentre as quais se destacam os algoritmos de busca gulosa (*Greedy Search*).

Essa abordagem constrói a solução do problema etapa por etapa, fundamentando-se na premissa de fazer a melhor escolha local a cada iteração para obter o benefício imediato, sem considerar as implicações futuras ou revisitar decisões passadas. Segundo Júnior (2017), os algoritmos gulosos são amplamente aplicados em problemas de otimização com o objetivo de identificar um conjunto de candidatos que maximize (ou minimize) o valor de uma função objetivo.

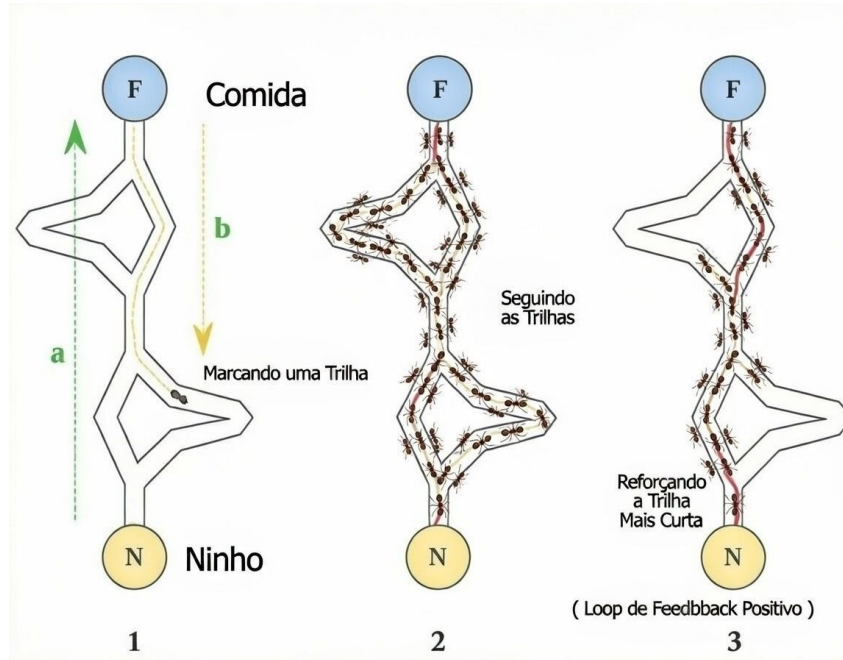
2.2.3 Otimização de Colônia de Formigas (ACO)

A Otimização por Colônia de Formigas (ACO - *Ant Colony Optimization*) é uma meta-heurística probabilística proposta originalmente por Marco Dorigo no início da década de 1990, destinada à resolução de problemas de otimização combinatória difíceis (*NP-hard*). A técnica inspira-se no comportamento de forrageamento de colônias de formigas reais, especificamente na capacidade desses insetos de encontrar o caminho mais curto entre o ninho e uma fonte de alimento sem a necessidade de controle centralizado ou visão global do ambiente (DORIGO; STÜTZLE, 2019).

O mecanismo central que rege esse comportamento é denominado **estigmergia**, mecanismo de auto-organização por meio da comunicação indireta mediada pelo ambiente. Ao se deslocarem, as formigas depositam no solo uma substância química volátil chamada feromônio. Inicialmente, a exploração ocorre de forma dispersa, mas à medida que determinados caminhos são percorridos com maior frequência, eles acumulam maior intensidade química.

Consequentemente, outras formigas são atraídas para essas rotas de maior concentração, reforçando-as recursivamente. Esse processo de *feedback* positivo permite que a colônia convirja rapidamente para as rotas otimizadas (AKHTAR, 2019), conforme ilustrado na Figura 7.

Figura 7 – Comportamento de uma Colônia de Formigas reais



Fonte: Adaptado de Choudhary (2024).

Computacionalmente, o ACO modela o problema de otimização como um grafo de construção $G = (V, E)$, onde formigas artificiais constroem soluções passo a passo. Diferentemente de algoritmos gulosos determinísticos, a escolha do próximo componente da solução é estocástica. De acordo com Babu e Chiranjeevi (2016), a probabilidade P_{ij}^k de uma formiga k , localizada no nó i , mover-se para o nó j é dada pela seguinte regra de transição aleatória:

$$P_{ij}^k(t) = \frac{[\tau_{ij}(t)]^\alpha \cdot [\eta_{ij}]^\beta}{\sum_{l \in N_i^k} [\tau_{il}(t)]^\alpha \cdot [\eta_{il}]^\beta}, \quad \text{se } j \in N_i^k \quad (1)$$

onde:

- $\tau_{ij}(t)$ é a intensidade da trilha de feromônio na aresta (i, j) no instante t ;
- η_{ij} é a informação heurística (visibilidade), geralmente calculada como o inverso do custo da aresta $(1/d_{ij})$;
- α e β são parâmetros que ponderam a importância relativa do feromônio (memória histórica) e da heurística (conhecimento guloso), respectivamente;
- N_i^k é o conjunto de nós vizinhos viáveis para a formiga k (para evitar visitas repetidas em problemas como o TSP).

O equilíbrio entre α e β é crucial: se $\alpha = 0$, o algoritmo torna-se puramente guloso; se $\beta = 0$, a busca depende exclusivamente do feromônio, o que pode levar à estagnação em ótimos locais (BABU; CHIRANJEEVI, 2016).

Além da construção da solução, a dinâmica do algoritmo depende da atualização do feromônio. Este processo envolve dois mecanismos antagônicos: o reforço das trilhas pertencentes às melhores soluções encontradas e a evaporação. A evaporação é modelada matematicamente pela redução da intensidade do feromônio em todas as arestas ao final de cada ciclo, conforme a equação:

$$\tau_{ij}(t+1) = (1 - \rho) \cdot \tau_{ij}(t) + \Delta\tau_{ij} \quad (2)$$

onde $\rho \in (0, 1]$ é a taxa de evaporação. Segundo Dorigo e Stützle (2019), a evaporação é um mecanismo de equilíbrio vital para evitar a convergência prematura, impedindo que a busca fique restrita rapidamente às primeiras soluções subótimas encontradas.

2.2.4 Algoritmo Genético

Os Algoritmos Genéticos (GAs) constituem uma das vertentes da Computação Evolutiva, inspirados nos mecanismos da evolução biológica e na teoria da seleção natural. Segundo D. e Chakraborty (2019), os GAs operam sob a premissa da onde a pressão seletiva favorece a preservação das melhores soluções, onde a busca pela solução ótima não ocorre através de um caminho determinístico único, mas sim pela evolução iterativa de uma população de soluções candidatas. Embora tenham suas raízes nos estudos de John Holland na década de 1960, firmaram-se como uma ferramenta robusta para resolver problemas de otimização complexos, tanto restritos quanto irrestritos, onde métodos analíticos tradicionais falham.

Diferentemente de métodos baseados em gradientes, que dependem de derivadas e buscam a otimização a partir de um único ponto inicial, correndo o risco de estagnar em mínimos locais, os Algoritmos Genéticos são estocásticos e globais. Li et al. (2014) explicam que o poder do GA reside na manipulação simultânea de uma população de cromossomos (indivíduos). Tal processo é concretizado por meio de operadores específicos que, quando adequadamente calibrados, são capazes de aprimorar substancialmente o desempenho final do algoritmo.

Na implementação de um GA, o processo evolutivo é orquestrado por três operadores genéticos fundamentais, que são seleção, cruzamento (*Crossover*) e mutação, aplicados ciclicamente até que um critério de parada seja atingido. Contudo, algumas implementações adicionam mais um operador, o elitismo. Os operadores citados encontram-se descritos a seguir:

1. **Seleção:** Conforme detalhado por D. e Chakraborty (2019), esta etapa seleciona os indivíduos de melhor desempenho (maior aptidão ou *fitness*) da população atual para agirem como pais da próxima geração, garantindo que as melhores características sejam preservadas.
2. **Cruzamento (*Crossover*):** Considerado o operador principal por Li et al. (2014), o cruzamento combina a informação genética de dois pais para criar novos descendentes. Isso permite a recombinação de boas subsoluções, facilitando a exploração de novas áreas promissoras no espaço de busca.
3. **Mutação:** Este operador introduz alterações aleatórias em um ou mais genes do indivíduo. Segundo Li et al. (2014), a mutação é vital para manter a diversidade genética da população, prevenindo a convergência prematura para ótimos locais e garantindo que o algoritmo não fique confinado a um único pico no espaço de soluções.
4. **Elitismo:** Uma estratégia complementar citada por D. e Chakraborty (2019) que consiste em copiar, inalterados, os melhores indivíduos da geração atual para a próxima. Isso garante que a qualidade da melhor solução encontrada nunca diminua (monotonicidade) ao longo do processo evolutivo.

Essa abordagem híbrida permite que o algoritmo explore diferentes regiões do espaço de busca em paralelo, oferecendo um equilíbrio eficaz entre a exploração de novos caminhos e o refinamento das melhores soluções encontradas, o que aumenta significativamente a probabilidade de encontrar o ótimo global (D.; CHAKRABORTY, 2019).

2.3 Trabalhos Relacionados

Esta seção dedica-se à apresentação de trabalhos relacionados, compreendendo a análise bibliométrica focada em técnicas de esteganografia LSB abordadas como problemas de otimização. O levantamento objetiva mapear o estado da arte, destacando a relevância e as contribuições da abordagem proposta. A discussão organiza-se conforme as bases de dados consultadas, incorporando, adicionalmente, obras importantes encontradas em outras fontes que fundamentaram esta pesquisa.

2.3.1 Análise Bibliométrica

O recurso da análise bibliométrica foi utilizado neste trabalho para lidar com o grande volume de dados científicos disponíveis sobre o tema. Técnicas bibliométricas foram utilizadas essencialmente para a análise quantitativa da produção acadêmica gerada ao longo dos últimos vinte e cinco anos (de 2000 a 2024), permitindo evidenciar a relevância do tema, identificar tendências e os pontos comuns aos artigos encontrados.

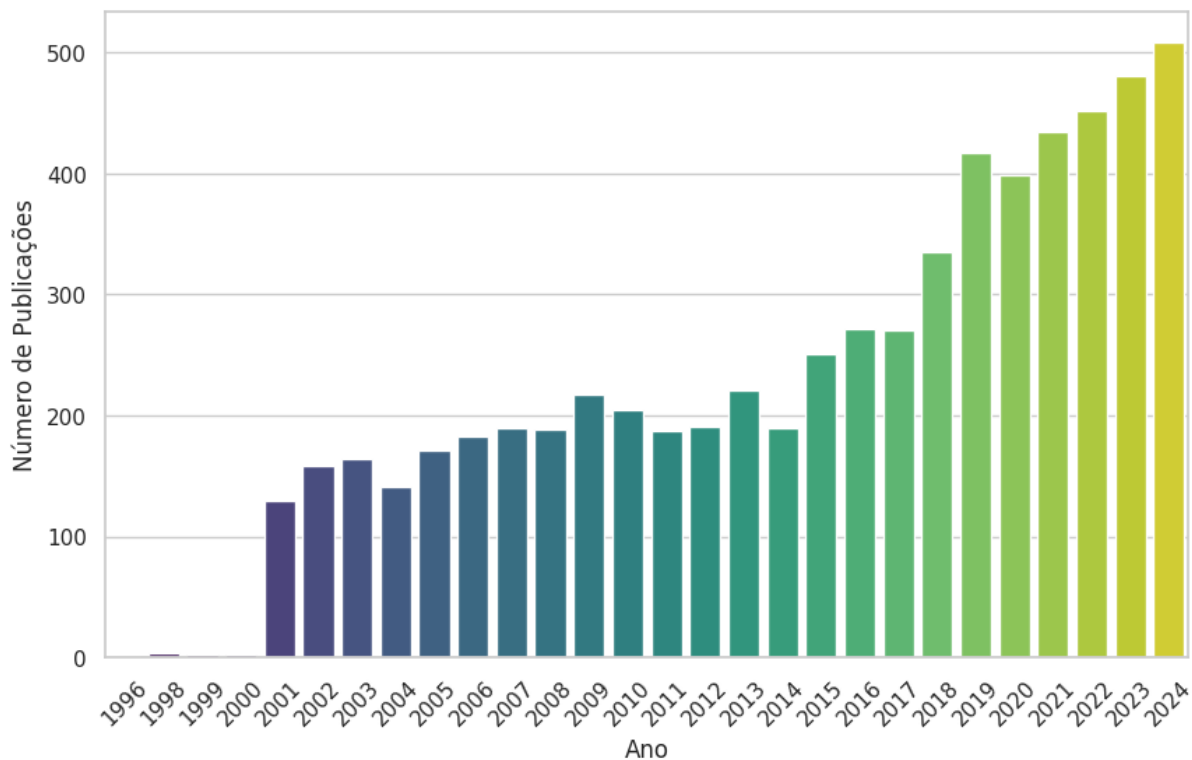
A análise mapeou o *corpus* bibliográfico relacionado ao tema desta pesquisa, utilizando as bases de dados *Scopus*, *Web of Science* e *IEEE Xplore*. A Figura 8 apresenta a distribuição temporal do número de publicações, consolidando todos os artigos recuperados. Em todas as três bases, foi empregado o mesmo termo de busca¹:

```
"lsb steganogra*
OR least significant bit steganogra*
OR image steganogra*"
```

Os resultados das buscas em cada uma das bases foram exportados, convertidos para o formato CSV e submetidos à análise. Inicialmente, foram aplicados critérios de exclusão aos registros do arquivo gerado, verificando-se as colunas referentes ao DOI, Título e Autores. Qualquer artigo com dados ausentes nesses campos foi excluído.

Em seguida, a análise foi delimitada aos trabalhos publicados nos últimos vinte e cinco anos (de 2000 a 2024). Foram identificados artigos duplicados, ou seja, artigos indexados em mais de uma base apresentando o mesmo DOI e Título. Assim, os artigos considerados duplicados foram removidos, mantendo-se apenas uma ocorrência única de cada trabalho.

Figura 8 – Histórico de produção acadêmica

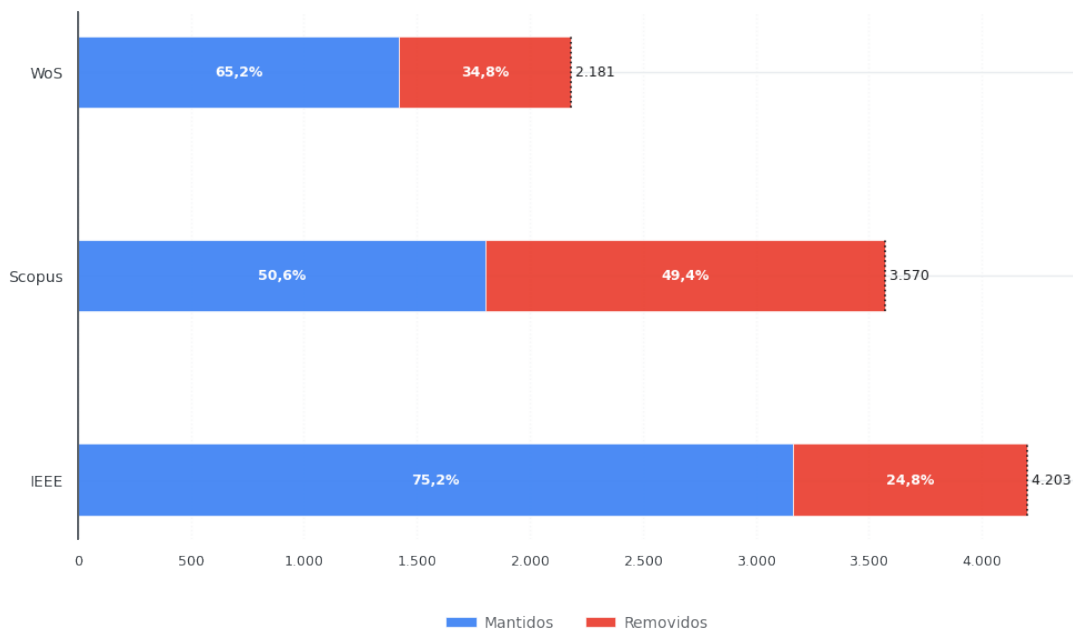


Fonte: Autor.

¹ O operador “*” representa o truncamento para capturar variações terminológicas (como singular e plural) nas bases consultadas.

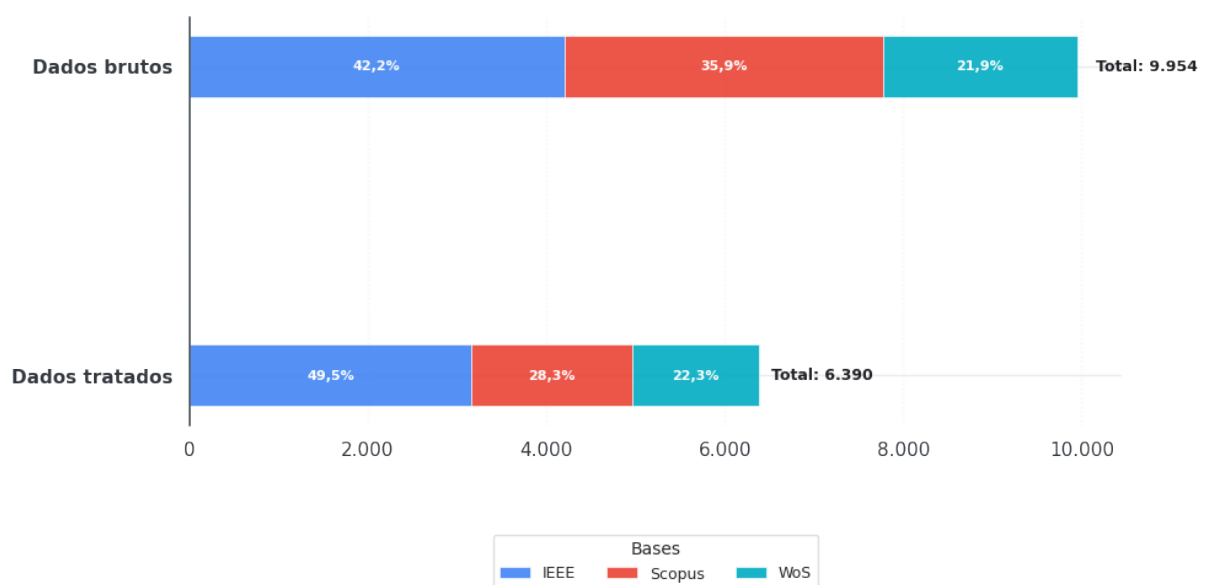
As Figuras 9 e 10 apresentam os dados resultantes após a aplicação dos critérios de exclusão. A Figura 9 detalha o quantitativo inicial de artigos recuperados em cada base, bem como a taxa de retenção após a filtragem. Já a Figura 10 exibe os dados brutos e tratados, segmentados por base de dados, evidenciando a participação percentual de cada fonte na composição do corpus final utilizado nas análises subsequentes.

Figura 9 – Publicações filtradas por base



Fonte: Autor.

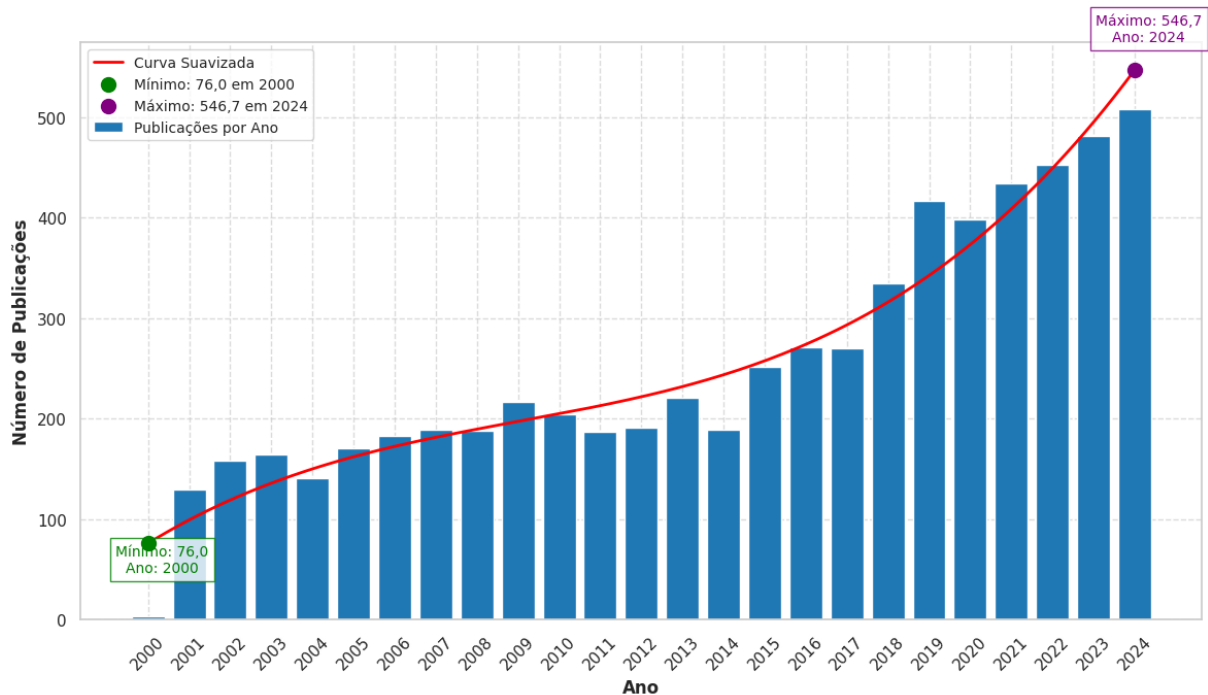
Figura 10 – Resultado quantitativo da busca por base



Fonte: Autor.

O conjunto de artigos selecionados totalizou 6.390 registros. A análise preliminar examinou a evolução da produção acadêmica sobre o tema ao longo do período estipulado, possibilitando a geração de uma curva de tendência. A Figura 11 apresenta esses dados, destacando os valores máximos e mínimos observados ao longo da série histórica.

Figura 11 – Tendência de produção acadêmica ao longo dos anos



Fonte: Autor.

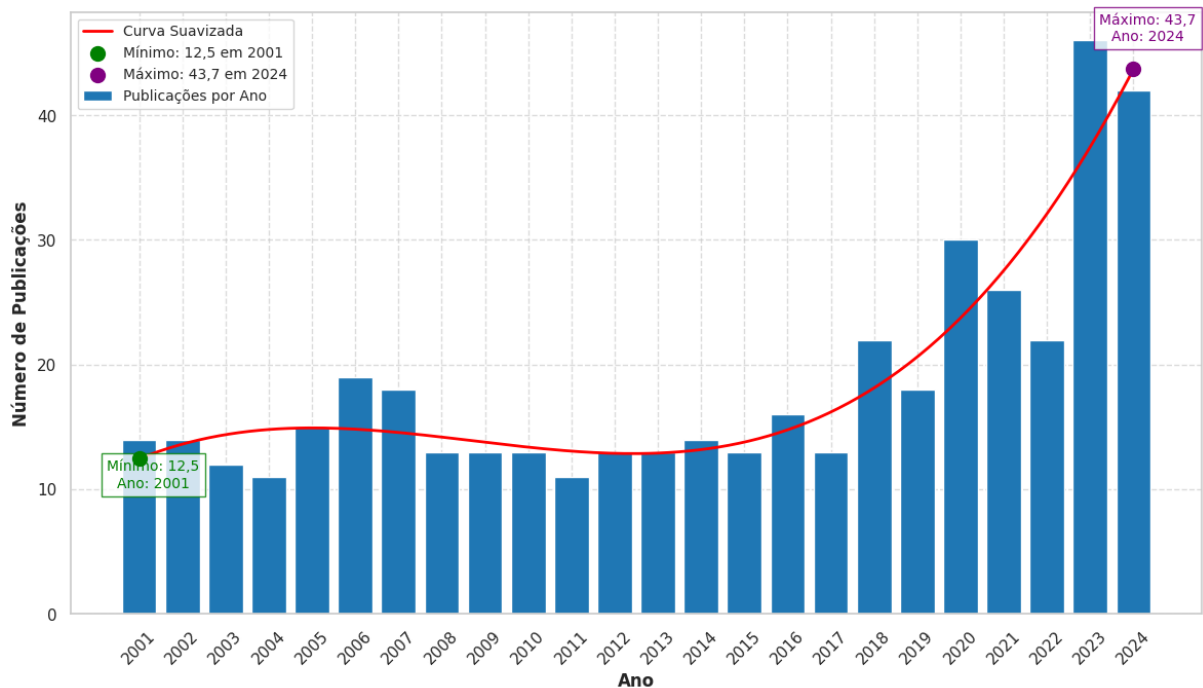
Posteriormente, novos critérios de refinamento foram aplicados ao dataframe tratado. Utilizou-se um filtro para selecionar especificamente os artigos que mencionassem termos relacionados a "Algoritmos de Otimização" em seus Títulos, Resumos ou Palavras-chave. Essa estratégia resultou em um subconjunto reduzido, permitindo uma nova análise temporal focada. A evolução deste subconjunto, bem como sua respectiva curva de tendência, encontra-se na Figura 12. A expressão regular utilizada para gerar este subconjunto é apresentada a seguir:

```
optim.* algorithm|optim.* technique|heuristic|metaheuristic
```

Por fim, os últimos critérios de exclusão foram aplicados, mantendo-se apenas artigos que mencionassem, em seu título, resumo ou palavras-chave, termos relacionados à Esteganografia LSB. A expressão regular utilizada no último filtro aplicado ao dataframe está apresentada a seguir:

```
\blsb steganogra.\b|\bleast significant bit steganogra.\b|
\blsb technique\b|\blsb embed\b
```

Figura 12 – Tendência de produção acadêmica envolvendo Otimização e Esteganografia



Fonte: Autor.

Ao final do processo, a aplicação dos critérios resultou em um corpus final de 22 artigos diretamente relacionados ao tema deste trabalho. Essa análise evidenciou a relevância do tema e demonstrou o crescente interesse dos pesquisadores no uso de ferramentas de otimização como meio de aprimorar os resultados da Esteganografia LSB ao longo dos últimos vinte anos. As informações dos trabalhos selecionados, cujo acesso foi possível, constam nas Tabelas 1, 2 e 3.

2.3.2 Base IEEE Xplore

Na base IEEE, os trabalhos selecionados, presentes na Tabela 1, abordam o aprimoramento das técnicas de ocultação e a evolução dos métodos de esteganálise. Ibanez, Djamal e Najmurrokhman (2018) propõem uma abordagem defensiva fundamentada em Algoritmos Genéticos (GA). Diferentemente da substituição simples, o método emprega o GA para permutar e embaralhar a mensagem previamente à sua inserção, atuando como uma camada de criptografia que dificulta a extração não autorizada. Os autores fazem uso do teste PSNR (*Peak Signal-to-Noise Ratio*), utilizado com o intuito de verificar a qualidade da imagem resultante.

Embora guarde semelhanças com a abordagem desenvolvida neste trabalho, o método de Ibanez, Djamal e Najmurrokhman (2018) exige que a mensagem seja reembaralhada a cada nova operação de ocultação. Ressalta-se que, ainda que os autores não reportem o custo computacional temporal, os resultados experimentais indicam eficácia utilizando 1000 gerações do algoritmo genético.

Tabela 1 – Artigos selecionados da base IEEE Xplore

Autores (Ano)	Título
Ibanez, Djamal e Najmurokhman (2018)	Optimization of Least Significant Bit Steganography Using Genetic Algorithm to Improve Data Security
Shaozhang et al. (2009)	Detecting LSB Steganography Based on Noise Function
Luo, Liu e Liu (2005)	Detecting LSB steganography based on dynamic masks

Fonte: Autor.

Em contrapartida, os demais estudos desta base dedicam-se à evolução da esteganálise. Luo, Liu e Liu (2005) desenvolveram um método baseado em Máscaras Dinâmicas, capaz de identificar alterações no plano LSB através da análise de correlações espaciais em vizinhanças de pixels, detectando inserções imperceptíveis à inspeção visual. Complementarmente, Shaozhang et al. (2009) abordam a detecção sob a ótica estatística, introduzindo uma Função de Ruído que distingue efetivamente o ruído natural da imagem daquele induzido artificialmente pela esteganografia LSB.

2.3.3 Base Web of Science (WoS)

Os trabalhos selecionados da base WoS, presentes na Tabela 2, são recentes e revelam três vertentes distintas de otimização aplicadas à esteganografia moderna, confirmando a tendência de hibridização entre segurança e inteligência computacional. Em consonância com a abordagem deste trabalho, Khalil, Sarhan e Alshewimy (2024) concentram-se na otimização espacial, empregando a meta-heurística de Busca Tabu para determinar as posições ideais de inserção dos bits na imagem. Tal técnica é aplicada em conjunto com a criptografia baseada em Mapas Caóticos 2D.

Por outro lado, Kumbhakar, Sanyal e Karforma (2023) exploram a otimização no domínio da frequência e da representação de dados. A abordagem utiliza *Deep Autoencoders* em conjunto com a Transformada Discreta de Cosseno (DCT) e criptografia assimétrica ElGamal. Diferentemente da substituição espacial direta, este método prioriza a robustez contra ataques de processamento de imagem e a capacidade de ocultação, mantendo um alto índice de similaridade estrutural (SSIM), que uma das métricas utilizadas na avaliação de um novo modelo de esteganografia em imagens.

Tabela 2 – Artigos selecionados da base Web of Science (WoS)

Autores (Ano)	Título
Khalil, Sarhan e Alshewimy (2024)	A secure image steganography based on LSB technique and 2D chaotic maps
Olewi et al. (2024)	Enhanced Security Through Integrated Morse Code Encryption and LSB Steganography in Digital Communications
Kumbhakar, Sanyal e Karforma (2023)	An optimal and efficient data security technique through crypto-stegano for E-commerce

Fonte: Autor.

Sob uma perspectiva distinta, focada na eficiência do pré-processamento, Olewi et al. (2024) propõem a otimização da própria carga útil (*payload*). Em vez de buscar "onde" esconder, os autores reduzem "o que" será escondido, utilizando Código Morse integrado a técnicas de compressão e criptografia. Essa estratégia minimiza a quantidade de alterações necessárias nos bits menos significativos, resultando em um Erro Quadrático Médio (MSE) extremamente baixo e preservando a qualidade estatística da imagem de cobertura.

Os estudos encontrados, apesar de aplicarem meta-heurísticas para aprimorar a qualidade da esteganografia, realizam a busca de soluções de forma síncrona à inserção, resultando em um custo computacional elevado. Além disso, o foco dessas pesquisas recai predominantemente sobre o reforço via criptografia. Este trabalho propõe uma abordagem distinta ao desacoplar a busca de estratégias por meio do pré-processamento, mitigando a latência e otimizando o desempenho do sistema.

2.3.4 Base Scopus

A análise do corpus bibliográfico extraído da base Scopus, com os artigos expostos na Tabela 3, revela que a evolução da esteganografia LSB vai além da simples ocultação sequencial de dados, ramificando-se em ecossistemas complexos, explorando diferentes algoritmos meta-heurísticos bioinspirados, arquiteturas de criptografia híbridas e o foco em alterações na própria lógica de funcionamento do algoritmo LSB para torná-lo menos detectável a técnicas de esteganálise especializadas.

O primeiro trabalho analisado dessa base justifica o investimento em métodos mais complexos de inserção pela capacidade de detecção da esteganálise contemporânea. Conforme evidenciado por Ge, Gao e Wang (2007), o uso de estatísticas RS e classificadores de aprendizado de máquina permite identificar substituições LSB convencionais com precisão elevada. Isso impulsiona inovações nessa área, para que novas técnicas sejam capazes de superar os métodos de estagnálises atuais.

Uma vertente dominante na literatura é o emprego de meta-heurísticas para solucionar o problema de seleção dos pixels para inserção de dados ou a busca por coeficientes ideais. Um exemplo identificado na base Scopus é o trabalho de Hussain e Khodher (2024), que utiliza o Algoritmo do Morcego (*Bat Algorithm*) para proteger imagens médicas. Neste trabalho, o algoritmo é explorado para otimizar a seleção das posições de inserção na imagem de cobertura, atuando em conjunto com a criptografia caótica para aprimorar a imperceptibilidade e a segurança da esteganografia LSB. O conjunto de imagens utilizado restringiu-se a exemplos médicos.

Tabela 3 – Artigos selecionados da base Scopus

Autores (Ano)	Título
Hussain e Khodher (2024)	Securing Medical Images Using Chaotic Map Encryption and LSB Steganography
Hussein e Alhijaj (2023)	Protection of images by combination of vernam stream cipher, AES and LSB steganography in a video clip
Sari et al. (2022)	An Enhancement of DES, AES Based on Imperceptibility Along with LSB
G et al. (2021)	Enhanced Data Encryption in IOT using ECC Cryptography and LSB Steganography
Khudher (2021)	LSB Steganography strengthen footprint biometric template
Rafrastara et al. (2019)	Image steganography using inverted LSB based on 2nd, 3rd and 4th LSB pattern
Nasreen (2016)	An efficient SEAC system for different data transmission over internet
Boughaci, Kemouche e Lachibi (2016)	Stochastic local search combined with LSB technique for image steganography
Singhal e Bedi (2015)	Steganography using cuckoo optimized wavelet coefficients
Sajasi e Eftekhari-Moghadam (2013)	A high quality image hiding scheme based upon noise visibility function and an optimal chaotic based encryption method
Mare, Vladutiu e Prodan (2011)	Decreasing change impact using smart LSB pixel mapping and data rearrangement
Zin e Soe (2011)	Implementation and analysis of three steganographic approaches
Ge, Gao e Wang (2007)	Least significant bit steganography detection with machine learning techniques

Fonte: Autor.

Abordando uma perspectiva distinta, o trabalho de Singhal e Bedi (2015) utilizou a esteganografia LSB como base para a construção de um modelo de inserção no domínio da frequência, aplicando a otimização *Cuckoo Search* nos coeficientes da Transformada Wavelet (DWT).

Em uma linha similar, Khudher (2021) empregaram a otimização por Colônia de Abelhas Artificiais (ABC), outro algoritmo bioinspirado, para fortalecer templates biométricos de pegadas (*footprints*). Contudo, nesse estudo, a otimização não visa aprimorar o desempenho da esteganografia LSB, mas sim garantir a segurança e a privacidade dos dados contra acessos não autorizados. A esteganografia é usada apenas como parte do sistema, sem ser o foco do trabalho.

Historicamente, a busca pela inserção dos dados na melhor posição remonta a Mare, Vladutiu e Prodan (2011), que propuseram o uso de Algoritmos Genéticos (GA) para realizar um "mapeamento inteligente de pixels" e o rearranjo de dados, objetivando minimizar o impacto visual antes da inserção. Nesse método, a imagem passa por um pré-processamento para criar um mapa de regiões adequadas para inserção dos dados, a partir disso, a implementação do algoritmo GA seleciona as melhores posições para ocultar as informações na imagem de cobertura. Neste estudo a geração da imagem executa o processo a cada informação oculta, sem manter um banco de dados disponível para inserção, o que torna o processo mais lento.

Complementarmente, Boughaci, Kemouche e Lachibi (2016) demonstraram que a Busca Local Estocástica (SLS) supera a substituição cega ao identificar blocos na imagem estatisticamente similares à mensagem. Nessa abordagem, durante a ocultação, tanto a imagem quanto a mensagem são submetidas ao modelo, onde a informação é segmentada em blocos de 4 bits, sendo a melhor combinação determinada pela Distância de Hamming como função de aptidão (fitness). De forma análoga ao proposto por Mare, Vladutiu e Prodan (2011), cada aplicação do método demanda o pré-processamento da imagem de cobertura e a execução do algoritmo de otimização para viabilizar a inserção dos dados. Ou seja, assim como os demais o processo é lento e exige o reprocessamento a cada nova informação.

G et al. (2021) abordam uma implementação voltada à Internet das Coisas (IoT), propondo uma arquitetura que combina a Criptografia de Curva Elíptica (ECC) com LSB. No campo multimídia, Hussein e Alhijaj (2023) expandem a hibridização da esteganografia LSB ao aplicar o *Advanced Encryption Standard* (AES) em vídeos. Na mesma vertente, Sari et al. (2022) priorizam a imperceptibilidade do LSB ao combiná-lo sequencialmente com os métodos clássicos *Data Encryption Standard* (DES) e AES. Historicamente, essas abordagens dialogam com o estudo de Zin e Soe (2011), que estabeleceu as bases para a integração de cifras de fluxo (RC4) com LSB.

Por fim, observa-se uma vertente de estudos que propõe alterações na lógica de substituição ou a adaptação à textura da imagem. Rafrastara et al. (2019) destacam-se com a técnica de *Modified Inverted LSB*, analisando padrões até o 4º bit menos significativo para realizar inversões baseadas na vizinhança dos pixels. De forma semelhante, Sajasi e Eftekhari-Moghadam (2013)

empregam a Função de Visibilidade de Ruído (NVF) para ajustar a carga útil de acordo com a textura local.

2.3.5 Outros Trabalhos Seleccionados

Adicionalmente, outros trabalhos relevantes para a fundamentação desta pesquisa foram identificados em bases distintas das contempladas na análise bibliométrica. Tais estudos são detalhados nesta subsecção.

Wazirali et al. (2019) abordam o *trade-off* entre capacidade de ocultação e degradação da imagem como um problema de otimização. O estudo propõe uma busca pela solução ótima via Algoritmos Genéticos, empregando operações que maximizam a correspondência entre os bits menos significativos. O esquema superou abordagens LSB tradicionais, apresentando menores distorções e aprimorando a indetectabilidade.

Similarmente, Banharnsakun (2018) aplicam a otimização por Colônia de Abelhas Artificiais (ABC) para refinar a técnica LSB. O trabalho modela a atribuição de blocos para inserção como um problema combinatório, adaptando o algoritmo ABC, originalmente numérico, através de listas de atribuição. Os resultados demonstraram avanços na qualidade visual e na robustez contra distorções, com a segurança reforçada pela exigência da permutação correta para a recuperação dos dados.

Com o objetivo de mitigar a degradação visual, Walia et al. (2018) desenvolveram um esquema de substituição LSB dirigido por chave, cuja configuração ótima é determinada via Busca Cuckoo (CS). Além da alta capacidade de incorporação, o método mostrou-se resistente a distorções. A eficácia foi validada através das métricas PSNR e SSIM (Structural Similarity Matrix). Assim como no trabalho de Dasgupta, Mondal e Dutta (2013), onde os autores utilizam Algoritmos Genéticos para otimizar a técnica LSB padrão, empregando métodos de anti-esteganálise para fins comparativos. Essa abordagem resultou em melhorias significativas no índice PSNR. Apesar de utilizarem meta-heurísticas para melhorar a imperceptibilidade, tais abordagens geralmente realizam a busca no momento da inserção, elevando o tempo de processamento.

Em síntese, a revisão da literatura revela que, embora o estado da arte tenha avançado significativamente na robustez do tratamento de mensagens via criptografia híbrida e nas técnicas de inserção adaptativa, persiste uma lacuna quanto à eficiência operacional dos processos de otimização. Enquanto as abordagens contemporâneas realizam a busca heurística de forma síncrona, o que eleva o custo computacional no ato da ocultação, este trabalho propõe um modelo inovador baseado no desacoplamento dessas etapas.

A inovação desta proposta reside na transferência do esforço de processamento para uma etapa prévia de mapeamento, reduzindo drasticamente a latência no momento da inserção ao limitar a tarefa do sistema à seleção de uma solução já indexada na árvore de busca.

Tal estrutura não apenas amplia as possibilidades de aplicação de meta-heurísticas complexas, como também se fundamenta em um cenário onde a otimização é um requisito de segurança essencial frente à precisão da esteganálise moderna. Adicionalmente, o levantamento bibliográfico ratifica o uso das métricas PSNR e SSIM, aliado à análise visual e à submissão dos esteganogramas a testes estatísticos, como os parâmetros consolidados para validar a imperceptibilidade e a fidelidade dos resultados obtidos nesta pesquisa.

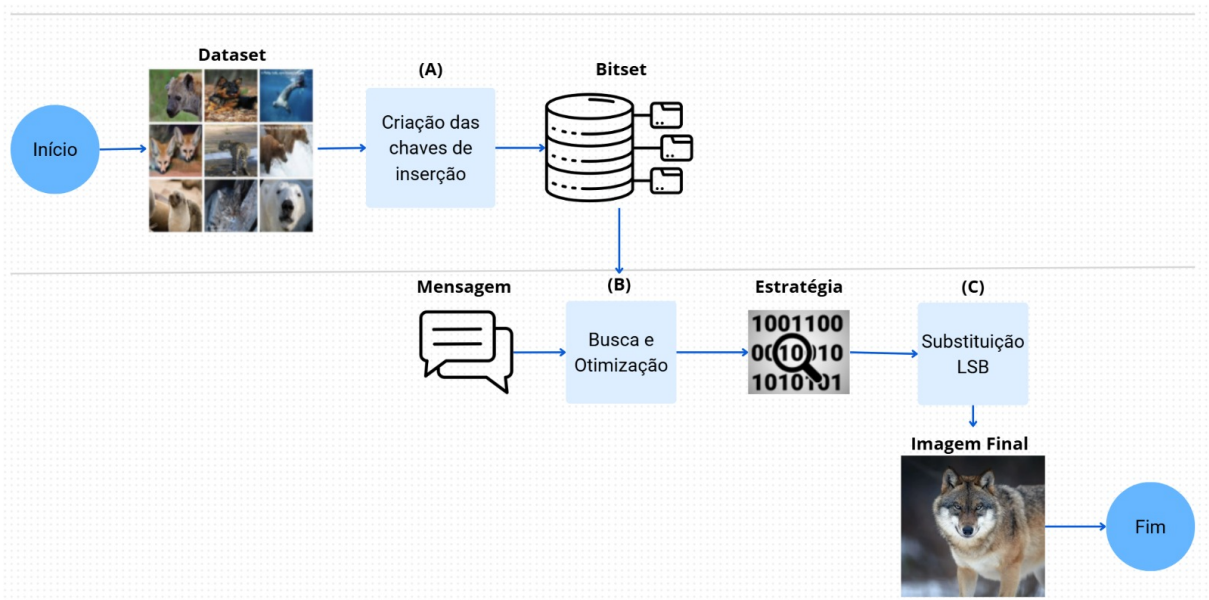
3 MÉTODO PROPOSTO

Esta Seção apresenta o método desenvolvido para uma nova implementação de esteganografia LSB, bem como os algoritmos de otimização empregados. O problema foi modelado com o objetivo de identificar o par ideal composto por imagem de cobertura e chave de inserção. Para tal, adotaram-se abordagens baseadas em algoritmos de busca em duas meta-heurísticas bioinspiradas, a Otimização por Colônia de Formigas (ACO) e o Algoritmo Genético (GA). Inicialmente, descrevem-se as estruturas fundamentais para a construção da árvore de busca; posteriormente, detalham-se os métodos de otimização aplicados.

3.1 Método

A Figura 13 ilustra o fluxo do método proposto. A fase inicial compreende o pré-processamento, no qual um amplo conjunto de imagens é selecionado para servir de base. Em paralelo, estratégias de inserção (chaves de inserção) são criadas. O *dataset* selecionado é utilizado como entrada na Etapa A (geração de chaves). Nesta etapa, cada imagem do banco é associada a uma chave de inserção, resultando em um arquivo de *bitsets* que contém milhões de possibilidades de ocultação de dados.

Figura 13 – Método proposto



Fonte: Autor.

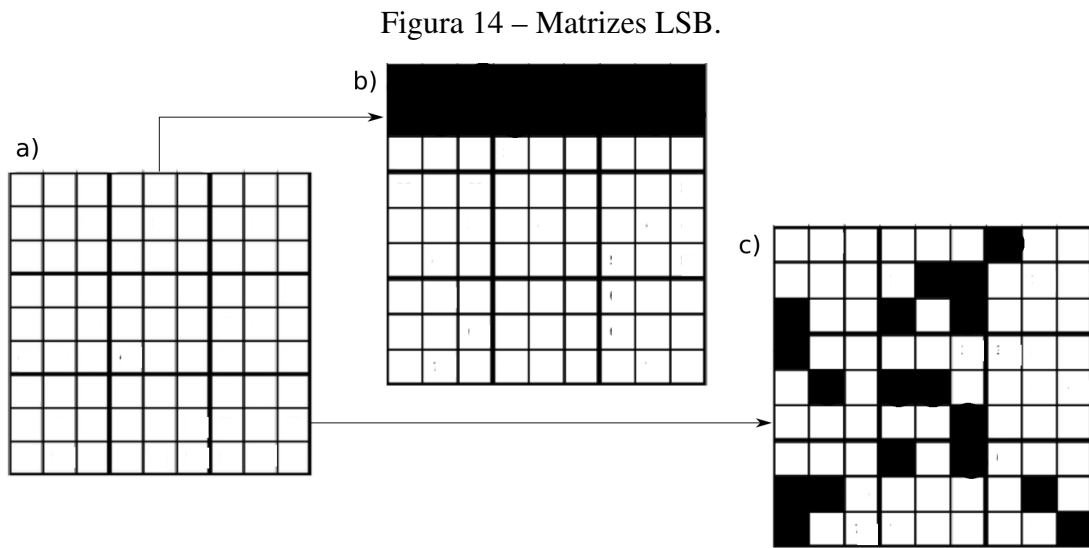
A segunda fase foca no processo de otimização e substituição LSB. A mensagem secreta e o arquivo de *bitsets* servem de entrada para a Etapa B, onde os dados são estruturados em uma árvore de busca. Algoritmos baseados nas meta-heurísticos (ACO ou GA) percorrem essa

estrutura para identificar a chave (par de estratégia e imagem de cobertura) mais eficaz para a inserção dos dados.

Por fim, a Etapa C recebe a melhor chave de inserção selecionada pelo algoritmo e executa a inserção não sequencial na imagem correspondente, gerando imagem final. Os detalhes operacionais de cada etapa são discutidos nas subseções a seguir.

3.2 Modelo de Esteganografia LSB

Esta Seção descreve um novo método de esteganografia LSB para ocultar mensagens em imagens digitais. O modelo proposto efetua a inserção dos bits da mensagem de forma aleatória e não sequencial, diferente do que ocorre na implementação habitual do modelo LSB. Tal comportamento faz com que a mensagem oculta não seja agrupada em um único ponto da imagem de cobertura, como demonstrado na Figura 14. Adicionalmente, é proposto o uso de um banco de imagens com o objetivo de selecionar uma imagem que minimize o número de bits alterados.



Fonte: Autor.

Na Figura 14 encontram-se três matrizes LSB de dimensão 9x9. A matriz (a) é uma representação de uma matriz LSB de uma imagem de cobertura sem nenhuma manipulação. Na matriz (b) encontra-se a matriz LSB da imagem de cobertura após o processo de esteganografia LSB mais comum, onde os bits da mensagem oculta são inseridos sequencialmente em um único ponto da imagem representado pelos espaços em preto. E na matriz (c) tem-se a representação da matriz LSB de uma imagem de cobertura após o processo de esteganografia LSB onde os bits da mensagem foram inseridos de maneira aleatória na imagem de cobertura.

O primeiro passo para a implementação do modelo foi a montagem do *benchmark*. Nos experimentos iniciais, para as imagens foi utilizado um banco de dados disponível no *Kaggle*,

o *Dogs vs Cats*¹, que possui um total de 25 mil imagens, com dimensões distintas. Para as mensagens, foi utilizado um banco de mensagens de texto extraídas da rede social X, antigo Twitter, o *Twitter Threads*².

Posteriormente, após a coleta dos resultados parciais, foram utilizadas outras bases mais complexas de imagens, uma vez que a primeira base utilizada possuía imagens da mesma natureza. A base selecionada foi a *Real and Fake Image Dataset*³, um banco de imagens com mais de 30 GB de dados de variados tipos, como pessoas, paisagens, animais etc. Para o banco textual, no experimento final foram utilizadas mensagens extraídas do Reddit, *Predicting Reddit Community Engagement Dataset*⁴.

A implementação do modelo proposto requer etapas de pré-processamento. A primeira consiste no mapeamento das dimensões de todas as imagens do banco de dados, visando identificar a capacidade máxima de incorporação com base na imagem de menor dimensão. Essa etapa foi necessária apenas no experimento inicial, visto que as imagens apresentavam dimensões variadas. A imagem com as menores dimensões delimitou o espaço operacional do algoritmo LSB em 50x38 pixels, estabelecendo o tamanho máximo das mensagens em 1024 bits (128 caracteres). No modelo final, a utilização da base *Real and Fake Image Dataset*, com imagens padronizadas em 200x200 pixels, ampliou o espaço operacional do algoritmo. Isso permitiu aumentar o tamanho máximo das mensagens utilizadas nos experimentos para 2240 bits (280 caracteres).

Com a quantidade máxima de informação definida, realiza-se o segundo passo do pré-processamento: a geração das estratégias de inserção de bits. Uma estratégia é definida como um vetor de tamanho m , cujos valores consistem em posições aleatórias dentro do intervalo $[0, n]$. Nesse contexto, m representa o tamanho da mensagem em bits e n corresponde ao tamanho total do vetor LSB disponível na imagem de referência (determinada pela menor dimensão do banco de dados). Para a execução dos testes, foi gerado um conjunto fixo de z estratégias distintas, que serviram como população inicial para a busca da solução ótima.

Tabela 4 – Parâmetros de configuração para o Experimento Inicial.

Parâmetro	Valor
Capacidade Máxima de inserção (n)	5.700 (Bits)
Tamanho Máximo da Mensagem (m)	1.024 bits
Estratégias Geradas (z)	1.000
Imagens Utilizadas	25.000
Quantidades de Chaves	25.000.000

Fonte: Autor.

¹ <https://www.kaggle.com/competitions/dogs-vs-cats>

² <https://www.kaggle.com/datasets/danielgrijalvas/twitter-threads>

³ <https://www.kaggle.com/datasets/awsaf49/artifact-dataset?select=afhq>

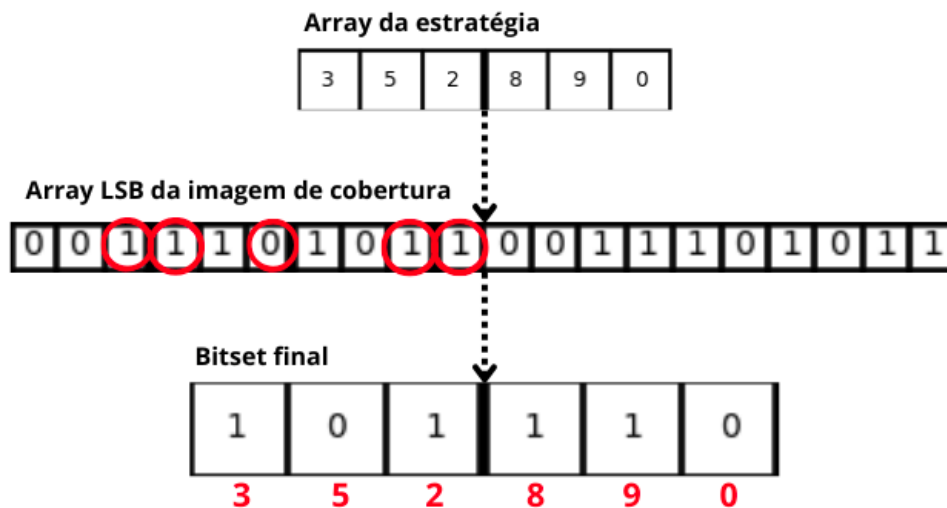
⁴ <https://www.kaggle.com/datasets/datancoffee/predicting-reddit-community-engagement-dataset>

Tabela 5 – Parâmetros de configuração para o Modelo Final.

Parâmetro	Valor
Capacidade Máxima de inserção (n)	120.000 (Bits)
Tamanho Máximo da Mensagem (m)	2.240 bits
Estratégias Geradas (z)	1.000
Imagens Utilizadas	25.000
Quantidades de Chaves	25.000.000

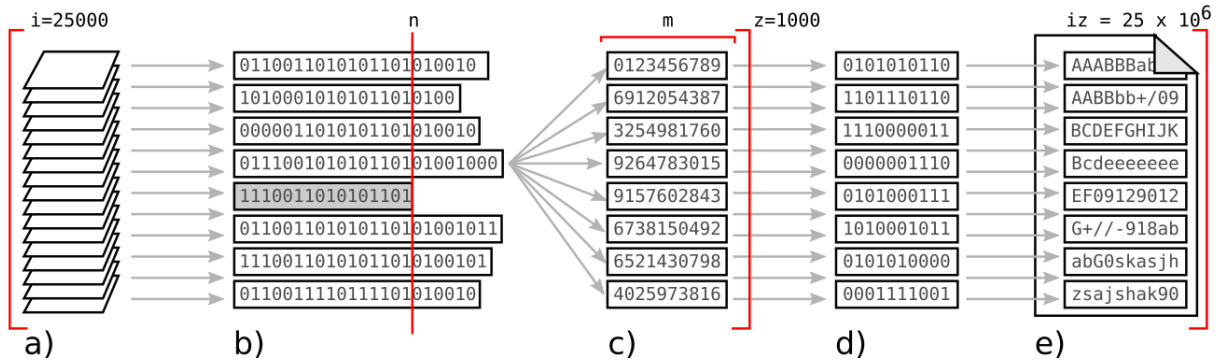
Fonte: Autor.

O último passo do pré-processamento é a geração de um arquivo de *bitsets*. Para cada imagem do banco foram criados z *arrays* binários, onde cada *array* é o reposicionamento de alguns bits do *array* LSB da imagem de cobertura segundo a de estratégia utilizada, como apresentado na Figura 15. Por fim, foi gerada uma lista contendo todos os *arrays* binários resultantes. A lista final possui 25 milhões de *bitsets* neste formato, salvos de forma ordenada em um arquivo de texto, onde cada *array* encontra-se codificado em base64.

Figura 15 – Operação de criação do *bitset*.

Fonte: Autor.

O objetivo do arquivo construído é criar um banco de *arrays* de comparação. Assim, quando uma mensagem for selecionada para ser oculta em uma imagem, será realizada uma busca estruturada nos *bitsets*, cujo resultado ótimo será um *bitset* com a menor distância de Hamming da representação binária da mensagem. Assim, com um banco de 25 milhões de possibilidades, a busca pelo melhor resultado torna-se um problema de otimização. A Figura 16 resume o processo de geração do arquivo de *bitsets*. Em (a) temos um conjunto com $i = 25000$ imagens, das quais são extraídas a mesma quantidade de *arrays* LSB (b). Em (c) temos o conjunto de z *arrays* de estratégia, sendo aplicados a cada imagem, gerando 25 milhões de *bitsets* (d), sendo ordenados e salvos em um arquivo utilizando codificação de base64 (e).

Figura 16 – Montagem do *array* de *bitsets*.

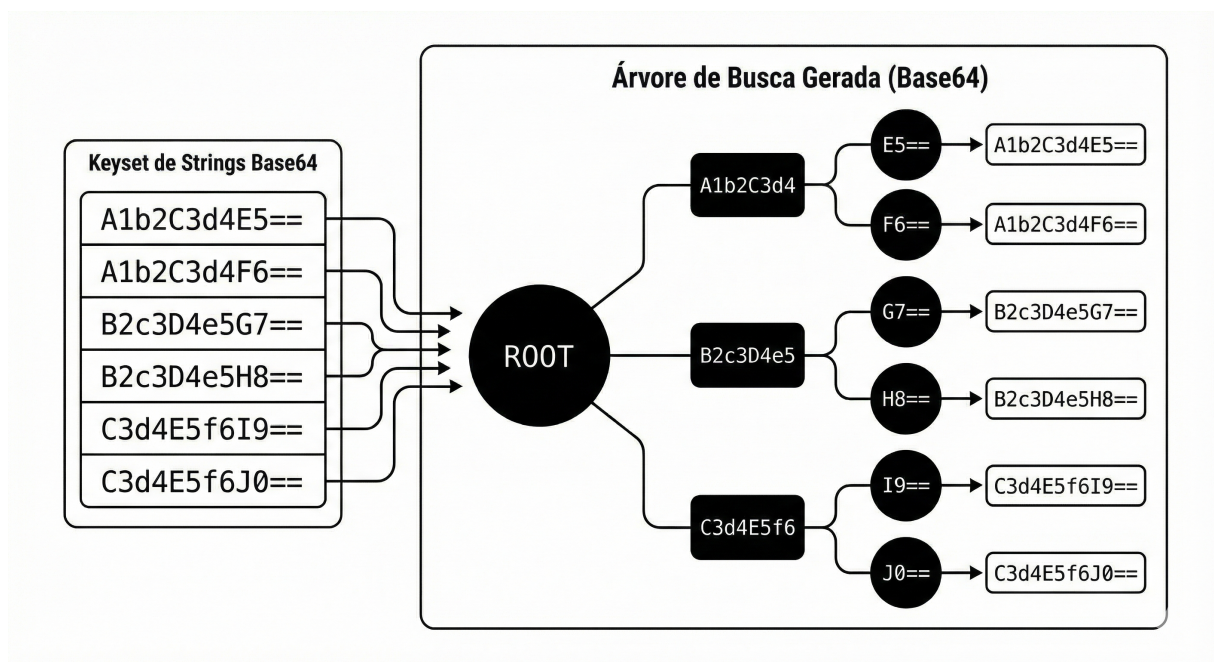
Fonte: Autor.

3.2.1 Árvore de busca

A próxima etapa proposta neste trabalho é realizar a decomposição do arquivo de *bitsets* construído em células, a partir da qual pode-se obter uma estrutura em forma de grafo, que denominamos árvore de busca. A árvore de busca implementada é uma estrutura fundamental para a aplicação dos algoritmos de busca e otimização que propomos DFS, Guloso, ACO e GA, sendo construída utilizando o arquivo de *bitsets* ordenado e codificado em base64.

A Figura 17 denota uma representação simplificada da árvore de busca. Os filhos são o resultado da diferença entre *bitsets*, enquanto os pais representam a parte comum entre eles. Pode-se observar, à esquerda, 6 conjuntos de *bitsets*, um em cada linha, ordenados como no arquivo original. À medida que caminha-se para a direita nos conjuntos, a árvore ramifica-se.

Figura 17 – Representação como árvore de busca



Fonte: Autor.

A estrutura de dados adotada na Árvore de Busca assemelha-se à estrutura *Patricia Trie*, descrita por Jung et al. (2002) como o método de acesso mais rápido entre as árvores binárias (*binary tries*), destacando-se por ter a estrutura de árvore mais rasa. Assim como na *Patricia Trie* clássica, é realizada a compressão de caminhos, onde nós únicos são fundidos para armazenar prefixos longos, reduzindo a altura da árvore. Diferencia-se, contudo, pelo método de construção, em vez de inserções sequenciais dinâmicas, utiliza-se um algoritmo determinístico de divisão e conquista sobre o conjunto de dados pré-ordenado. Isso elimina a sobrecarga de rebalanceamento e busca durante a criação, resultando em uma árvore estática otimizada para as operações de leitura das meta-heurísticas.

Dessa forma, define-se a estrutura da árvore de busca empregada neste trabalho. As estratégias, codificadas em Base64 e ordenadas no arquivo de origem, compartilham nós comuns e ramificam-se à medida que seus caracteres divergem. O resultado ideal da otimização corresponde ao caminho completo (da raiz a uma folha) que minimize a Distância de Hamming em relação à mensagem binarizada alvo.

De forma didática, o cálculo dessa métrica é construído gradativamente durante a navegação do algoritmo. Ao visitar um nó, o agente de busca (seja uma formiga ou um cromossomo) extrai o trecho da *string* ali armazenado, e então compara-o estritamente com o segmento equivalente da mensagem secreta. O número de caracteres divergentes forma o custo daquele passo. Assim, a avaliação de uma solução não exige que o caminho seja finalizado para ser medido, o custo final é simplesmente o somatório contínuo dessas divergências locais ao longo da descida pela topologia da árvore.

3.3 Busca e Otimização

Este trabalho propõe quatro algoritmos de busca e otimização, cujo objetivo é encontrar no arquivo de *bitsets* a combinação mais próxima de um resultado ótimo entre a mensagem secreta, a estratégia de inserção dos bits e a imagem do banco utilizada. Inicialmente, foi utilizado um algoritmo com uma implementação de uma busca simples, para percorrer todo o *bitset*, com 25 milhões de possibilidades, procurando o *array* com a menor distância de Hamming da mensagem utilizada. As seções seguintes descrevem os demais algoritmos propostos.

3.3.1 Algoritmos DFS e Busca Gulosa

Seguem dois algoritmos propostos com base na árvore de busca definida na Subseção 3.2.1. O primeiro é baseado em DFS, cujo objetivo é percorrer toda a árvore, seguindo cada caminho até atingir um nó folha antes de retroceder e explorar outros caminhos. O processo é repetido até que todos os nós sejam visitados. O segundo é uma implementação do *Greedy Search*, ou Busca Gulosa, que pretende percorrer parcialmente a árvore de busca, partindo da raiz e optando pelo filho de menor custo até chegar a um nó folha. Nessa execução, objetiva-se

uma redução significativa no tempo de processamento, embora possa não alcançar as soluções de menor custo identificadas pela DFS.

3.3.2 Meta-heurística ACO

Esta seção apresenta as heurísticas baseadas em ACO desenvolvidas para identificar o melhor par imagem-estratégia, operando sobre a árvore de busca definida na Subseção 3.2.1. Assim, foram propostas duas variações do algoritmo, onde a primeira trata-se de uma versão simplificada, caracterizada pela ausência do fator de evaporação do feromônio. Nesse modelo, cada formiga percorre o trajeto da raiz até uma folha, depositando feromônio nos vértices, processo que influencia probabilisticamente as rotas das gerações subsequentes. O Algoritmo 1 detalha essa implementação.

Algoritmo 1: Método baseado na meta-heurística ACO (Sem Evaporação)

```

1 Function root, Msg, T, K,  $\alpha$ ,  $\beta$ , Q:
2    $C_{best} \leftarrow \infty$ ;
3   for  $t \leftarrow 1$  to  $T$  do
4     for  $k \leftarrow 1$  to  $K$  do
5        $C_{path} \leftarrow \text{AntSearch}(\text{root}, \text{Msg}, 0, 0)$ ;
6        $C_{best} \leftarrow \min(C_{path}, C_{best})$ ;
7   return  $C_{best}$ ;

8 Function AntSearch( $u, \text{Msg}, pos, C_{curr}$ ):
9    $C_{node} \leftarrow \text{Hamming}(u.str, \text{Msg}[pos])$ ;
10   $C_{total} \leftarrow C_{curr} + C_{node}$ ;
11  if  $u$  is Leaf then
12    return  $C_{total}$ ;
13   $S_{probs} \leftarrow \emptyset$ ;
14  foreach  $v \in \text{Children}(u)$  do
15     $h \leftarrow \text{Hamming}(v.str, \text{Msg}[\text{next\_pos}])$ ;
16     $\eta_v \leftarrow 1.0 / (h + \epsilon)$ ;
17     $P_v \leftarrow (\tau_v)^\alpha \cdot (\eta_v)^\beta$ ;
18    Add  $P_v$  to  $S_{probs}$ ;
19   $v_{next} \leftarrow \text{SelectNode}(\text{Children}(u), S_{probs})$ ;
20   $C_{final} \leftarrow \text{AntSearch}(v_{next}, \text{Msg}, pos + \text{len}(u), C_{total})$ ;
21   $\Delta\tau \leftarrow Q / (C_{final} + \epsilon)$ ;
22   $\tau_{v_{next}} \leftarrow \tau_{v_{next}} + \Delta\tau$ ;
23  return  $C_{final}$ ;

```

A inicialização do método ocorre na linha 2, onde o melhor custo global C_{best} é definido como infinito. O núcleo do processo de busca é delimitado pelas linhas 3 a 7, que controlam a evolução da colônia de formigas artificiais. O laço externo (linha 3) avança as iterações t de 1 a T , enquanto o laço interno (linha 4) deposita K formigas para construir soluções independentes. Na linha 5, cada formiga invoca a função *AntSearch* para percorrer a árvore de busca. Imediatamente

após a construção do caminho, a linha 6 executa a atualização da melhor solução global, onde a função $\min(C_{path}, C_{best})$ compara o desempenho da formiga atual com o histórico, garantindo que C_{best} retenha sempre o menor custo encontrado durante a execução.

O procedimento *AntSearch*, que modela a navegação, inicia-se na linha 9. As linhas 10 e 11 computam o custo acumulado da rota parcial, somando o custo atual ao custo do novo nó. O critério de parada é estabelecido nas linhas 12 e 13, onde ao atingir um nó folha, a recursão retorna o custo total da solução.

A construção estocástica da rota ocorre no bloco das linhas 15 a 19. Para cada nó filho v , a linha 16 calcula a Distância de Hamming, e a linha 17 define a heurística local η_v . A probabilidade não normalizada P_v é composta na linha 18, ponderando o feromônio acumulado τ_v e a heurística η_v pelos expoentes α e β , respectivamente.

A escolha estocástica do próximo nó é realizada na linha 19 (*SelectNode*), utilizando a distribuição de probabilidade construída anteriormente. Em seguida, a linha 20 executa a chamada recursiva, aprofundando a busca na árvore e retornando o custo final da solução (C_{final}).

A atualização do feromônio ocorre após o cálculo do reforço (linha 21), incorporando uma constante de estabilidade $\varepsilon = 0,1$ ao denominador para evitar divisões por zero caso a Distância de Hamming seja nula. A atualização é então efetivada na linha 22. Neste ponto, o valor de feromônio do nó escolhido recebe o incremento aditivo ($\tau_{v_{next}} \leftarrow \tau_{v_{next}} + \Delta\tau$), consolidando o aprendizado da formiga. A recursão encerra-se na linha 23, retornando o custo total para o nível anterior.

Dessa forma, adapta-se o modelo clássico para a navegação em árvore. A probabilidade de transição de um nó pai u para um filho v (linha 19) e a regra de atualização de feromônio (linha 22) são formalizadas, respectivamente, por:

$$p_{uv} = \frac{[\tau_v]^\alpha \cdot [\eta_v]^\beta}{\sum_{k \in \text{Children}(u)} [\tau_k]^\alpha \cdot [\eta_k]^\beta} \quad (3)$$

$$\tau_v \leftarrow \tau_v + \frac{Q}{C_{final}} \quad (4)$$

3.3.3 Meta-heurística ACO com Evaporação

A segunda variação implementa o modelo *Ant System* clássico, introduzindo o coeficiente de evaporação para evitar a saturação prematura do feromônio. Diferente da abordagem tradicional que atualiza toda a estrutura a cada geração, este modelo utiliza uma estratégia eficiente. Nesse mecanismo, o decaimento do feromônio é calculado sob demanda apenas quando um nó é visitado, projetando o valor armazenado no tempo presente com base na diferença de gerações desde o último acesso. O Algoritmo 2 detalha essa implementação otimizada.

Algoritmo 2: Método baseado na meta-heurística ACO (Versão Refinada)

```

1 Function root, Msg, T, K,  $\alpha$ ,  $\beta$ ,  $\rho$ , Q:
2    $C_{best} \leftarrow \infty$ ;
3   for  $t \leftarrow 1$  to T do
4     for  $k \leftarrow 1$  to K do
5        $C_{path} \leftarrow \text{AntSearch}(\textit{root}, \textit{Msg}, t, 0, 0)$ ;
6        $C_{best} \leftarrow \min(C_{path}, C_{best})$ ;
7   return  $C_{best}$ ;

8 Function AntSearch(u, Msg, t, pos,  $C_{curr}$ ):
9    $C_{node} \leftarrow \text{Hamming}(u.str, \textit{Msg}[\textit{pos}])$ ;
10   $C_{total} \leftarrow C_{curr} + C_{node}$ ;
11  if u is Leaf then
12    return  $C_{total}$ ;
13   $S_{probs} \leftarrow \emptyset$ ;
14  foreach  $v \in \textit{Children}(u)$  do
15     $h \leftarrow \text{Hamming}(v.str, \textit{Msg}[\textit{next\_pos}])$ ;
16     $\eta_v \leftarrow 1.0 / (h + \epsilon)$ ;
17     $gap \leftarrow t - \textit{time}_v$ ;
18     $\tau_{eff} \leftarrow \tau_v \cdot (1 - \rho)^{gap}$ ;
19     $P_v \leftarrow (\tau_{eff})^\alpha \cdot (\eta_v)^\beta$ ;
20    Add  $P_v$  to  $S_{probs}$ ;
21   $v_{next} \leftarrow \text{SelectNode}(\textit{Children}(u), S_{probs})$ ;
22   $C_{final} \leftarrow \text{AntSearch}(v_{next}, \textit{Msg}, t, \textit{pos} + \textit{len}(u), C_{total})$ ;
23   $gap \leftarrow t - \textit{time}_{v_{next}}$ ;
24   $\tau_{curr} \leftarrow \tau_{v_{next}} \cdot (1 - \rho)^{gap}$ ;
25   $\Delta\tau \leftarrow Q / C_{final}$ ;
26   $\tau_{v_{next}} \leftarrow \tau_{curr} + \Delta\tau$ ;
27   $\textit{time}_{v_{next}} \leftarrow t$ ;
28  return  $C_{final}$ ;

```

A inicialização do método ocorre na linha 2, onde o custo global C_{best} é definido como infinito. O núcleo do processo de busca é delimitado pelas linhas 3 a 7, que controlam a evolução do sistema ao longo de T gerações. O laço externo (linha 3) avança o tempo discreto t , enquanto o laço interno (linha 4) deposita K formigas para construir soluções independentes, assim como na implementação anterior. Na linha 5, cada formiga invoca a função *AntSearch* para percorrer o grafo. Imediatamente após, a linha 6 executa a atualização da melhor solução global, onde a função $\min(C_{path}, C_{best})$ garante que C_{best} seja atualizado apenas se a nova solução superar o melhor registro histórico.

O procedimento *AntSearch*, que modela a navegação, inicia-se na linha 8. As linhas 9 e 10 computam o custo acumulado da rota parcial. O critério de parada é estabelecido nas linhas 11 e 12, onde ao atingir um nó folha, a recursão retorna o custo total da solução.

A construção estocástica da rota ocorre no bloco delimitado pelas linhas 15 a 20. Para cada nó candidato v , a linha 16 calcula a informação heurística η_v , incorporando uma constante de estabilidade $\varepsilon = 0,1$ ao denominador, assim como na ACO sem evaporação, evitando a divisão por 0. O mecanismo de evaporação é implementado nas linhas 17 e 18, onde calcula-se o hiato temporal $gap = t - time_v$ apenas para o nó analisado, projetando o feromônio efetivo (τ_{eff}) através do decaimento $(1 - \rho)^{gap}$. Essa estratégia permite estimar a evaporação sem a necessidade de acessar a memória global da árvore. Por fim, a probabilidade de transição P_v é composta na linha 19.

A seleção do próximo nó ocorre na linha 21, seguida pela chamada recursiva na linha 22, que aprofunda a busca até o fim da mensagem. A atualização definitiva do feromônio é realizada durante o retorno da recursão (*backtracking*), entre as linhas 23 e 27. O algoritmo recalcula o intervalo gap para o nó visitado (linha 23) e aplica a evaporação acumulada ao valor físico armazenado (linha 24). O reforço $\Delta\tau$, inversamente proporcional ao custo final (C_{final}), é calculado na linha 25. Por fim, as linhas 26 e 27 persistem o novo valor de feromônio e sincronizam o carimbo de tempo $time_{v_{next}}$ com o instante atual t . A regra de atualização atômica (linhas 24-26) pode ser formalizada como:

$$\tau_v \leftarrow \underbrace{\tau_v \cdot (1 - \rho)^{(t - t_v)}}_{\text{Evaporação Acumulada}} + \underbrace{\frac{Q}{C_{final}}}_{\text{Novo Reforço}} \quad (5)$$

3.3.4 Meta-heurística GA

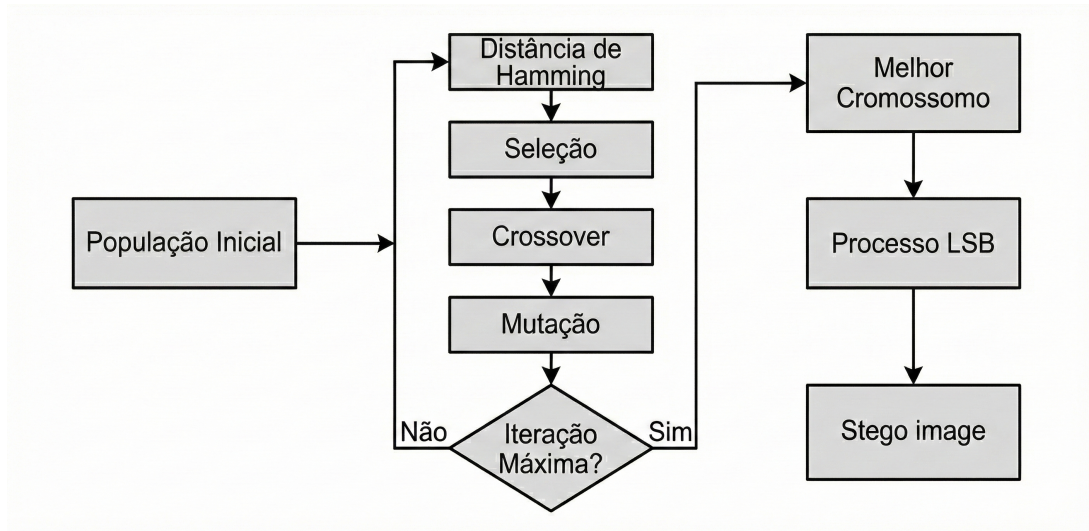
Esta seção descreve a heurística baseada em GA proposta como uma alternativa para além das implementações com o ACO, cujo objetivo também foi de encontrar o melhor par imagem-estratégica, utilizando a árvore de busca descrita na Subseção 3.2.1.

Na implementação proposta, O algoritmo para a otimização da seleção de estratégias na Árvore de busca é uma variação do Algoritmo Genético Canônico (GOLDBERG, 1989), adaptado ao contexto da busca na árvore e expresso na Figura 18. A implementação segue um modelo geracional com elitismo, operando sobre uma codificação inteira e pode ser observado no Algoritmo 3.

O algoritmo inicia nas linhas 2 a 3, onde a aptidão (*fitness*) da população inicial P é calculada. Para cada indivíduo Ind , a função *Evaluate* é invocada para determinar o seu custo, estabelecendo a base para o processo evolutivo. Cabe ressaltar que se trata de um problema de minimização, logo, quanto menor o custo final calculado, maior é a aptidão do indivíduo.

O ciclo geracional ocorre no laço das linhas 4 a 16. A cada iteração t , uma nova população P_{new} é construída. As linhas 6 e 7 implementam o Elitismo, onde uma quantidade k_{elite} , definida pela taxa P_e , determina quantos dos melhores indivíduos (presentes nos índices 1 a k_{elite}) são preservados e copiados diretamente para a próxima geração.

Figura 18 – Implementação da GA proposta



Fonte: Autor.

O preenchimento das vagas restantes ocorre no la o *While* (linhas 8 a 14). A sele o dos genitores, nas linhas 9 e 10, utiliza a Sele o por Truncamento (*Truncation Selection*). Os pais p_1 e p_2 s o escolhidos aleatoriamente, mas sua sele o   restrita   metade superior da popula o ($1 \dots N/2$), garantindo que apenas os indiv duos de melhor desempenho se reproduzam.

Os operadores gen ticos s o aplicados em seguida. A linha 11 gera um descendente atrav s do Cruzamento Uniforme (*Uniform Crossover*), onde cada gene do filho   herdado aleatoriamente de um dos pais. A linha 12 aplica a Muta o, sujeita   taxa P_m . O novo indiv duo tem seu custo calculado imediatamente na linha 13 antes de ser inserido na nova popula o. Ao final da gera o, a popula o principal   atualizada (linha 15).

O cerne da defini o da aptid o reside na fun o *EvaluateCost*, que decodifica o gen tipo do indiv duo, mapeando-o para a topologia da  rvore. O processo inicia no n  raiz ($TotalCost = 0$, $pos = 0$) e, para cada gene pertencente ao cromossomo do indiv duo, executa uma sequ ncia de etapas fundamentais.

Primeiramente, ocorre o c lculo do custo local, onde o algoritmo extrai o trecho da mensagem correspondente   profundidade atual da busca e calcula a Dist ncia de Hamming em rela o   cadeia de caracteres armazenada no n  atual (*u.string*). Essa penalidade local   imediatamente somada ao custo total acumulado (*TotalCost*). Imediatamente ap s a atualiza o do custo, avalia-se o crit rio de parada. Se o n  atual for uma folha da  rvore ou se toda a mensagem *Msg* j  tiver sido percorrida, a decodifica o   interrompida. Nesse cen rio, o *TotalCost* acumulado at  aquele n vel   retornado como o *fitness* definitivo daquele indiv duo.

Por fim, caso o crit rio de parada n o seja atingido, a busca prossegue e realiza a decodifica o da rota via aritm tica modular. Como a  rvore possui uma quantidade vari vel de filhos em cada n vel, o valor atual do gene (um inteiro est tico e arbitr rio) pode n o corresponder

diretamente às opções de caminhos disponíveis. Para solucionar esse problema, o algoritmo aplica uma operação de módulo entre o gene e a quantidade de filhos do nó atual ($k \leftarrow gene \pmod{|Children(u)|}$). Em seguida, o ponteiro da mensagem (pos) é deslocado e o nó atual é atualizado para o filho escolhido, repetindo o ciclo no nível inferior da árvore.

Algoritmo 3: Método baseado na meta-heurística GA

```

1 Function root, Msg, N, T, Pm, Pe:
2   foreach Ind  $\in P$  do
3     Ind.Cost  $\leftarrow$  EvaluateCost(Ind, root, Msg);
4   for  $t \leftarrow 1$  to  $T$  do
5      $P_{new} \leftarrow \emptyset$ ;
6      $k_{elite} \leftarrow \lfloor N \cdot P_e \rfloor$ ;
7     Add  $P[1 \dots k_{elite}]$  to  $P_{new}$ ;
8     while  $|P_{new}| < N$  do
9        $p_1 \leftarrow \text{RandomSelect}(P[1 \dots N/2])$ ;
10       $p_2 \leftarrow \text{RandomSelect}(P[1 \dots N/2])$ ;
11      Child  $\leftarrow \text{UniformCrossover}(p_1, p_2)$ ;
12      Mutate(Child,  $P_m$ );
13      Child.Cost  $\leftarrow \text{EvaluateCost}(\textit{Child}, \textit{root}, \textit{Msg})$ ;
14      Add Child to  $P_{new}$ ;
15      $P \leftarrow P_{new}$ ;
16   return  $P[1]$ ;

17 Function EvaluateCost(Ind, u, Msg):
18   TotalCost  $\leftarrow 0$ ;
19   pos  $\leftarrow 0$ ;
20   foreach gene  $\in \textit{Ind.chromosome}$  do
21     TotalCost  $\leftarrow \textit{TotalCost} + \text{Hamming}(u.string, \textit{Msg}[\textit{pos}])$ ;
22     if u is Leaf or End of Msg then
23       return TotalCost;
24      $k \leftarrow gene \pmod{|Children(u)|}$ ;
25      $u \leftarrow Children(u)[k]$ ;
26     pos  $\leftarrow pos + len(u.string)$ ;
27   return TotalCost;

```

4 RESULTADOS

Os métodos propostos foram implementados utilizando duas linguagens de programação Python e C++. A linguagem Python foi empregada para o pré-processamento de dados, manipulação das imagens, implementação do modelo LSB e análise estatística dos resultados. Os algoritmos de busca e otimização (meta-heurísticas) foram desenvolvidos na linguagem C++.

Nos experimentos para parametrização das meta-heurísticas, foi seguido o padrão ISO C++11, e os algoritmos foram compilados utilizando o G++ (GNU *Compiler Collection*) com a *flag* de otimização máxima (-O3). Os experimentos computacionais foram executados em um computador portátil equipado com processador Intel Core i7-12700H de 12ª Geração (2.30 GHz), 16 GB de memória RAM e sistema operacional Windows de 64 bits.

Nos experimentos finais, após a parametrização das meta-heurísticas e utilizando toda a capacidade dos arquivos de *bitsets*, os métodos propostos foram compilados utilizando o g++ 8.3.0 (opção de compilação -O3). Os experimentos foram executados em um computador com 128 GB de RAM DDR4 3200 MHz, processador Intel Core i9-14900K 3.2GHz (64 bits), 1 TB de SSD NVMe e sistema operacional Ubuntu 23.10.

4.1 Modelo LSB Inicial

O modelo inicial foi desenhado seguindo os parâmetros descritos na Tabela 4. O primeiro passo foi a geração dos arquivos de entrada dos algoritmos de otimização, que foram: o arquivo contendo o *bitset* com 25 milhões de possibilidades de substituição LSB e o arquivo de mensagens, ambos codificados em Base64.

Com isso, o método de Força Bruta foi executado e, durante cada execução, os dados gerados, os valores das distâncias mínimas e o tempo de execução de cada busca foram registrados e usados como linha de base de comparação. Após a execução da Força Bruta, foi encontrada uma média de Distância de Hamming de 273,1, enquanto no método de substituição comum a média encontrada para as 100 mensagens foi de 333,24. Isso demonstrava que o modelo era promissor, uma vez que o modelo, em seu primeiro teste, havia conseguido reduzir os valores absolutos da distancia de Hamming.

Em sequência, foram realizados os testes de desempenho do modelo de esteganografia LSB proposto. Para o teste, foram realizadas duas substituições LSB: uma no modelo tradicional de implementação e outra no modelo proposto. Para isso, a imagem selecionada como a melhor cobertura pelo método de Força Bruta foi utilizada para a Mensagem 1. A implementação comum da substituição LSB altera de maneira sequencial os bits menos significativos dos pixels da imagem de cobertura.

Dada uma imagem representada pela matriz $I = M \times N$, onde M é o número de linhas e N o número de colunas, cada elemento $P(i, j)$ representa o pixel na linha i e coluna j . Nesta implementação, a função submete todo valor de $P(i, j)$ à substituição LSB até que toda a mensagem seja inserida.

A implementação do modelo proposto varreu a imagem seguindo a mesma lógica; porém, o valor de $P(i, j)$ só era submetido à substituição LSB caso sua posição estivesse contida no array da estratégia escolhida pelo Força Bruta, forçando os bits da mensagem a ficarem espalhados na imagem e seguindo o limite de espaço estabelecido pela menor imagem do banco.

As imagens de esteganografia geradas pelas implementações LSB foram submetidas a um método de esteganálise conhecido como Ataque Aural. Inicialmente, as imagens foram submetidas a um algoritmo que saturou os valores dos pixels. Posteriormente, passaram por outro processamento que alterou a escala de cores. Ambos os métodos têm como objetivo expor as anomalias geradas pela substituição LSB. A Equação 6 descreve a função de saturação utilizada, que deve ser executada para todo $P(i, j) \in I$.

$$f(x) = \begin{cases} 0 & \text{se } x = 0 \text{ ou } x \text{ for par;} \\ 255 & \text{se } x \text{ for ímpar;} \end{cases} \quad (6)$$

A análise das imagens geradas pelo primeiro modelo evidenciou que restringir a área de manipulação das imagens do *dataset* às dimensões da menor imagem não foi uma estratégia eficaz. Mesmo com a separação dos bits de informação, a limitação espacial resultou na concentração de ruído. Com isso, embora o modelo proposto tenha apresentado desempenho superior ao método tradicional quanto à suavização das anomalias LSB, constatou-se a necessidade de aprimoramentos ampliando o espaço de trabalho do algoritmo de geração de estratégias.

Assim, um segundo experimento foi realizado, sem as limitações do primeiro descrito anteriormente, onde o campo de atuação do algoritmo que gera as estratégias de substituição LSB não se limitou às dimensões da menor imagem do banco.

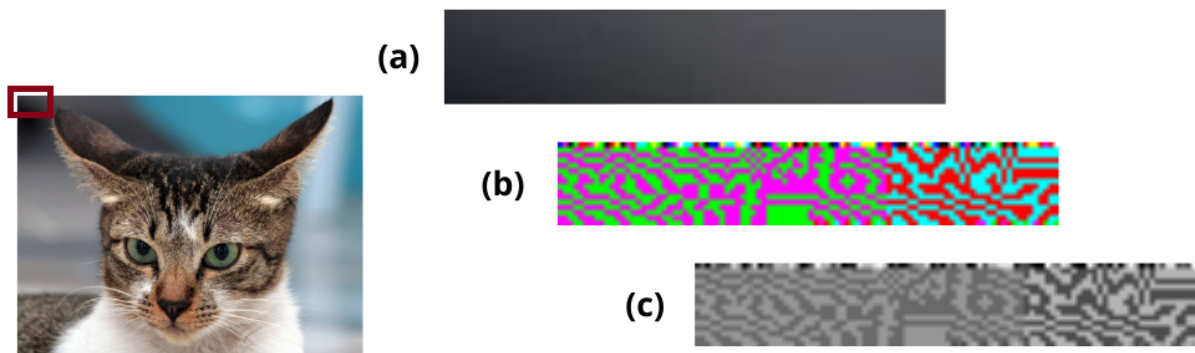
Nessa análise, a maior imagem do banco foi selecionada, com 1023×768 pixels, e foram geradas mil estratégias de inserção. A partir dessas estratégias e das imagens, foram geradas as chaves de substituição LSB e, dentre elas, foi selecionada, pelo algoritmo de Força Bruta, a chave mais adequada para a maior mensagem do banco, com 124 caracteres. O resultado desse teste encontra-se nas Figuras 19 e 20.

Nas Figuras 19 e 20, os recortes (b) foram submetidos a um processo de saturação para realce. É possível notar, no recorte (b) da Figura 19, uma linha de pixels agrupada na parte superior com padrão cromático distinto do restante da imagem. Essa anomalia torna-se ainda mais evidente no recorte (c), indicando o local de depósito dos bits da mensagem. Contudo, ao analisar os recortes (b) e (c) da Figura 20, nota-se que os artefatos gerados pela substituição

foram atenuados devido ao processo de espalhamento aleatório ao qual os bits foram submetidos, tornando-os imperceptíveis. Isso ocorreu porque o algoritmo gerador de estratégias considerou a resolução total da imagem, distribuindo os dados por toda a sua extensão e dificultando a análise visual.

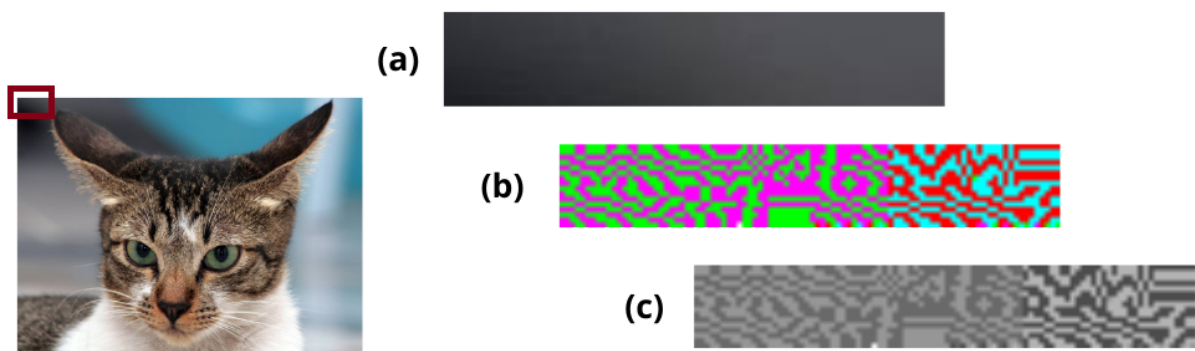
Isso pode ser visto com mais detalhes na Figura 21, onde os recortes (a) e (b) são da imagem que sofreu a substituição LSB simples, os recortes (c) e (d) são da imagem original e os recortes (e) e (f) da imagem que submetida a substituição LSB proposta neste trabalho. Nota-se que nos recortes (a) e (b) está circulado em vermelho a anomalia gerada pelo LSB simples, que ficam mais evidentes quando comparados com os recortes da imagem original. No entanto, o recorte (c) não apresenta diferenças perceptíveis do recorte (e), assim como os recortes (d) e (f).

Figura 19 – Imagem esteganográfica utilizando LSB comum



Fonte: Autor.

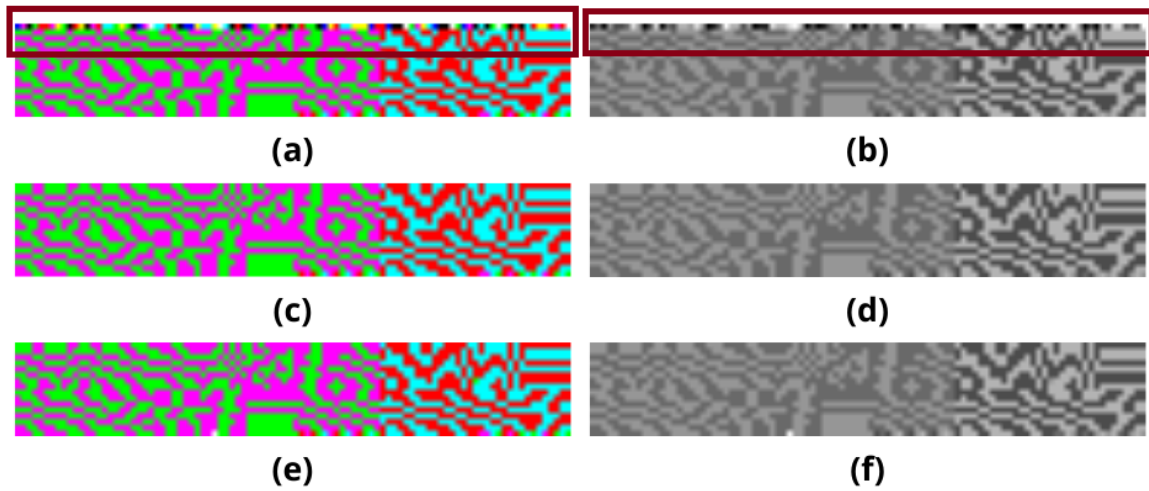
Figura 20 – Imagem esteganográfica utilizando o modelo LSB proposto



Fonte: Autor.

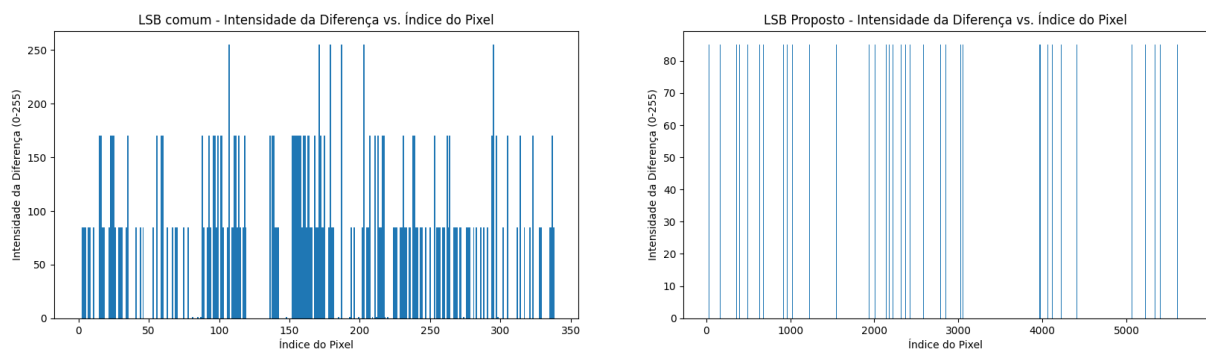
Uma análise da intensidade da diferença entre os pixels da imagem de cobertura e as imagens frutos dos modelos de esteganografia foi realizada. Assim como em (ABBAS et al., 2015), a diferença de pixels foi plotada contra o índice de pixels da imagem de cobertura. Os gráficos da análise do primeiro experimento encontram-se na Figura 22.

Figura 21 – Recortes das imagens submetidas a esteganálise



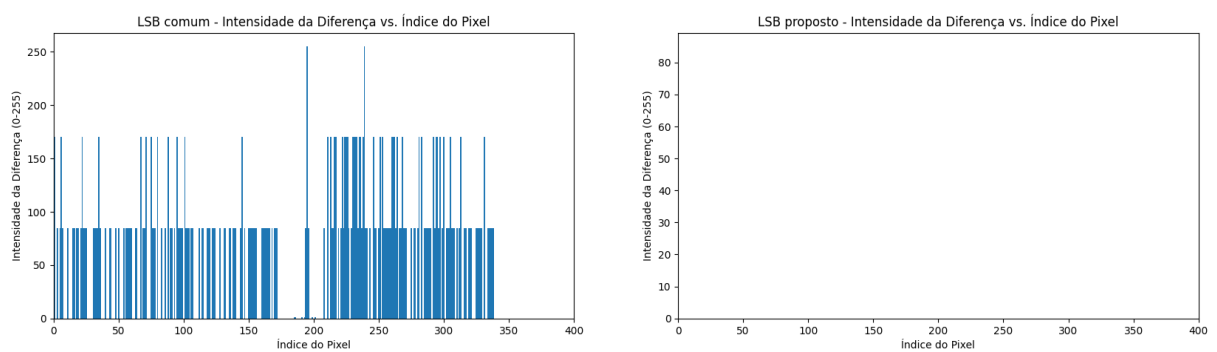
Fonte: Autor.

Figura 22 – Gráficos da intensidade da diferença de píxeis



Fonte: Autor.

Figura 23 – Gráficos da intensidade da diferença de píxeis do segundo experimento



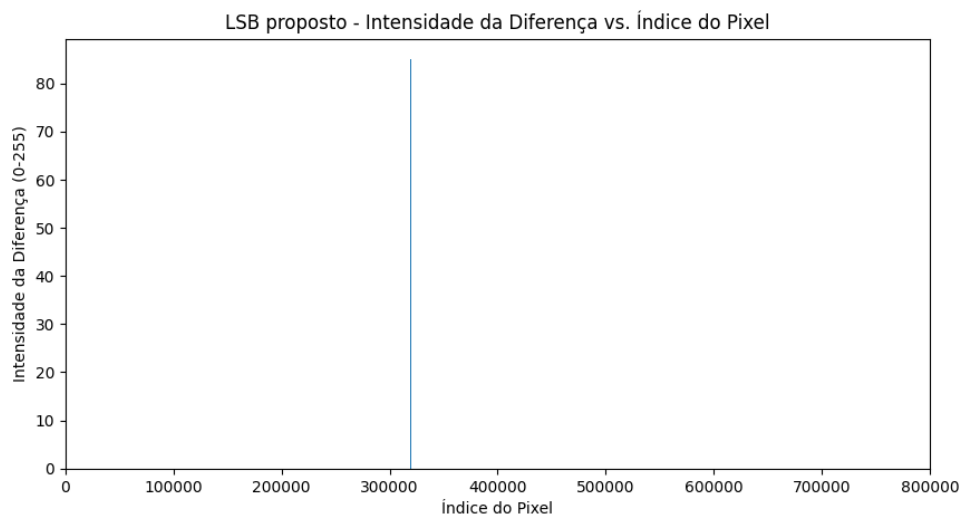
Fonte: Autor.

Nos gráficos da Figura 22, é possível notar que a intensidade das diferenças diminuiu no modelo LSB proposto, enquanto a presença da diferença torna-se mais espaçada. Como exposto anteriormente, no primeiro experimento o algoritmo deixa o ruído mais atenuado, porém a limitação do espaço da imagem restringe os resultados.

Essa análise também foi realizada nas imagens do segundo experimento, utilizando a maior imagem encontrada no banco. O resultado desse teste encontra-se nas Figuras 23 e 24. Os gráficos reforçam o que foi observado nos testes de Ataque Aural, nos quais, no modelo LSB proposto, os bits inseridos na imagem de cobertura estão mais dispersos e o uso de uma estratégia ideal para a mensagem inserida reduz consideravelmente a diferença de intensidade entre os pixels da imagem de cobertura e da imagem resultante.

Na Figura 23, a análise foi feita nas duas imagens geradas pelas substituições LSB; o gráfico limita o eixo x ao índice de pixel 400. Devido a isso, no gráfico do LSB proposto, não é possível notar nenhuma diferença entre os pixels da imagem gerada e da imagem original. Contudo, quando se amplia o índice do pixel na Figura 24, nota-se a presença da diferença, porém em uma intensidade e proporção inferiores ao experimento em que o campo de trabalho foi limitado para o algoritmo gerador de estratégias.

Figura 24 – Gráfico da intensidade da diferença de píxeis na imagem gerada pelo LSB proposto



Fonte: Autor.

4.2 Modelo LSB Final

Com base nos resultados dos experimentos iniciais, foram implementadas alterações visando avaliar a robustez do modelo. Utilizou-se outro banco de imagens, mais robusto e com maior diversidade de imagens. Além disso, o espaço de operação do algoritmo LSB foi ampliado, visto que os testes iniciais indicaram que a aplicação do algoritmo nas dimensões máximas da imagem favorece o seu desempenho. Os novos parâmetros estão definidos na Tabela 5.

As imagens do banco *Real and Fake Image Dataset* possuíam uma dimensão padrão (200x200 pixels), isso cumpre dois requisitos do modelo, amplia o espaço de operação do algoritmo que gera as estratégias de inserção e garante que o espaço corresponde a imagem completa e não a porções dela. Após a geração das chaves, combinações entre imagens e estratégias, o *bitset* com 25 milhões de possibilidades foi ordenado segundo os critérios previamente abordados.

Os passos dos experimentos iniciais foram mantidos e o método de Força Bruta foi executado, selecionando a melhor chave de inserção (estratégia mais imagem) para cada mensagem do banco do Reddit. Após a execução, obteve-se uma média de Distância de Hamming de 348,77, contra uma média de 596,59 do modelo LSB padrão, para um total médio de 829,26 bits por mensagem (cenário de máxima diferença possível).

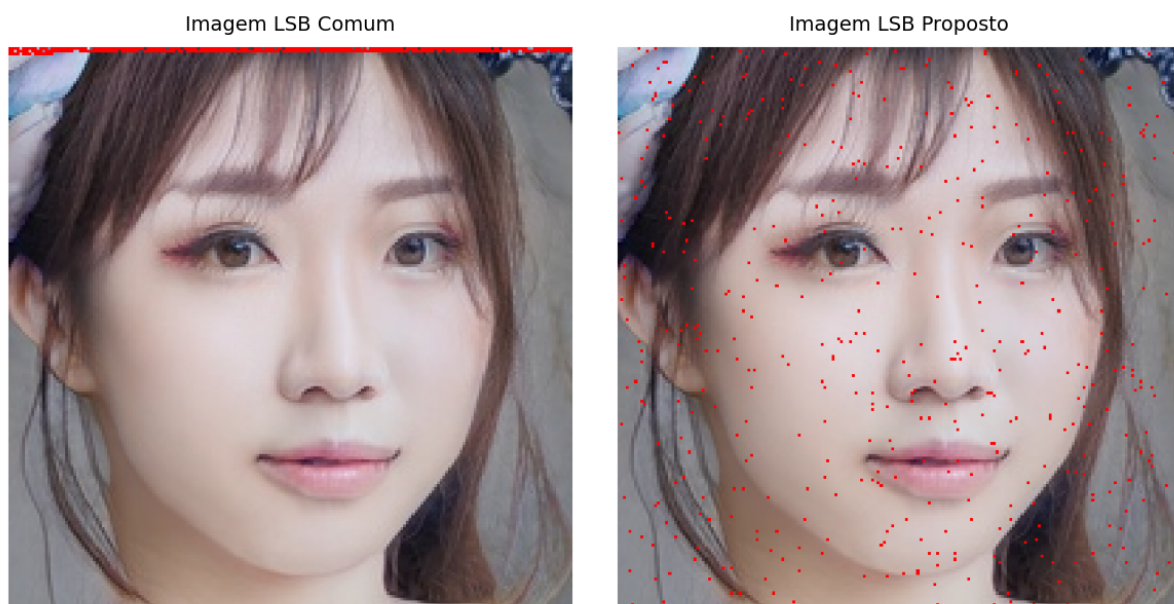
Para exibir os resultados dessa implementação, três mensagens do banco do Reddit foram selecionadas e o processo LSB foi aplicado utilizando a melhor chave para cada mensagem do teste. Também foi aplicado o modelo LSB comum nessas imagens selecionadas, utilizando as mesmas mensagens. Esses pares de imagens LSB gerados foram colocados lado a lado e os bits inseridos foram expostos em vermelho, para que ficasse evidente onde a mensagem estava inserida. Esse resultado pode ser observado nas Figuras 25, 26 e 27.

O modelo demonstra eficácia ao cumprir um dos objetivos iniciais: promover a dispersão da informação inserida por toda a extensão da imagem. Essa característica dificulta a esteganálise, pois evita a concentração de alterações em regiões específicas, ao contrário do LSB padrão, no qual as informações se acumulam no topo da imagem, gerando as distorções observadas na Figura 19.

As imagens resultantes da ocultação das três mensagens do Reddit, bem como as originais, foram submetidas à análise visual (Ataque Visual). Não foi possível notar diferenças perceptíveis mesmo após o processamento realizado pelos algoritmos de esteganálise, ou seja, a esteganálise não gerou nenhuma anomalia visível nas imagens finais. Tais resultados são evidenciados nas Figuras 28, 29, 30, 31, 32 e 33.

Assim como nos experimentos iniciais, as imagens geradas no LSB do modelo final foram submetidas à análise da diferença de pixels, presentes nas Figuras 34, 35, 36, 37, 38 e 39. Tal como nos testes iniciais, notam-se resultados melhores nas diferenças de pixels das imagens geradas pelo LSB proposto. Tal desempenho deve-se à dispersão da informação e à inserção estratégica nas melhores posições, fatores que atenuam ou eliminam grande parte das discrepâncias entre os pixels.

Figura 25 – Exposição da informação inserida nas imagens LSB geradas a partir da Mensagem de Teste 1



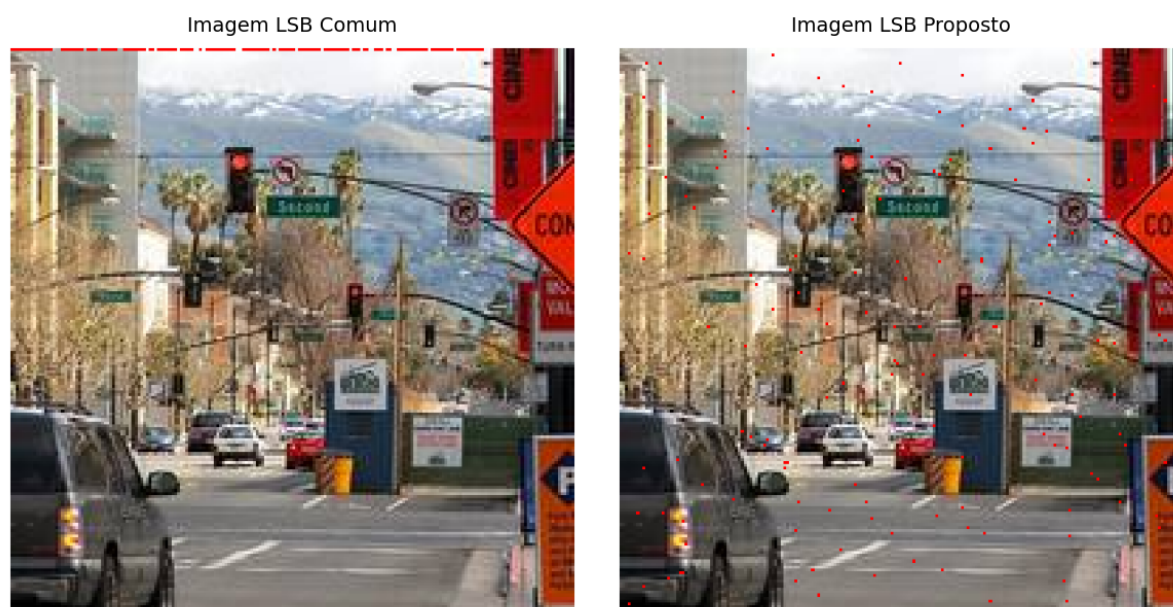
Fonte: Autor.

Figura 26 – Exposição da informação inserida nas imagens LSB geradas a partir da Mensagem de Teste 2



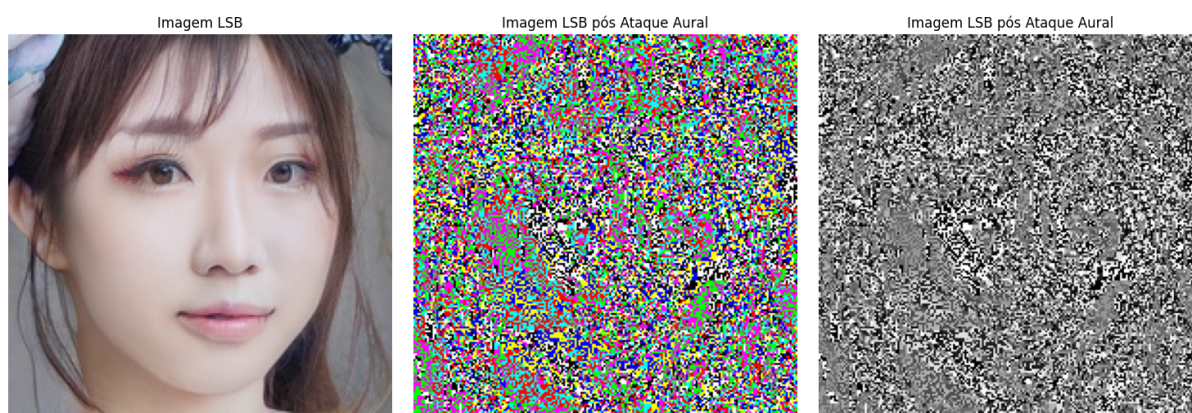
Fonte: Autor.

Figura 27 – Exposição da informação inserida nas imagens LSB geradas a partir da Mensagem de teste 3



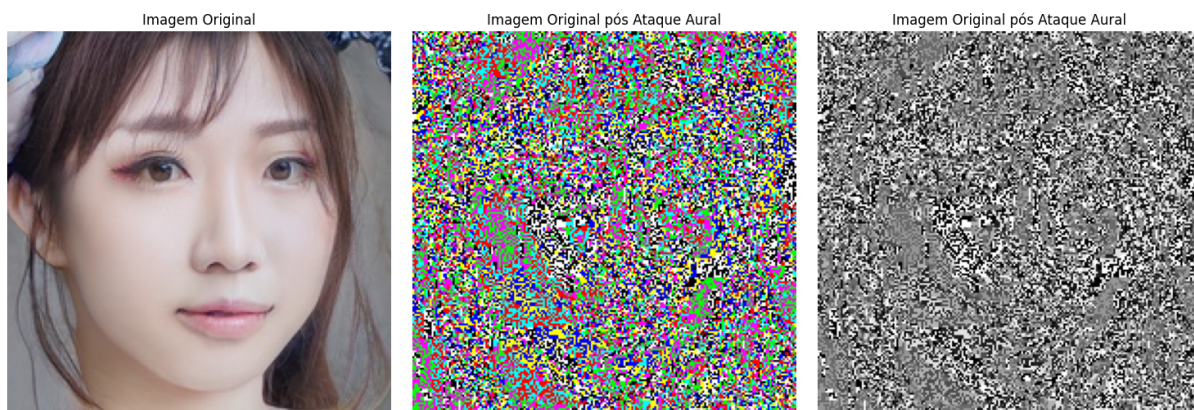
Fonte: Autor.

Figura 28 – Imagem LSB Proposta após esteganálise aural



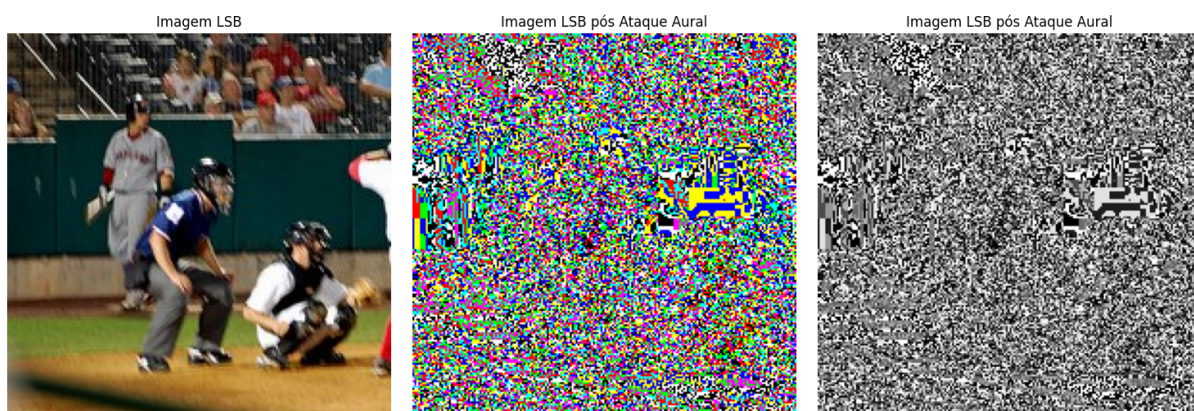
Fonte: Autor.

Figura 29 – Imagem original após esteganálise aural



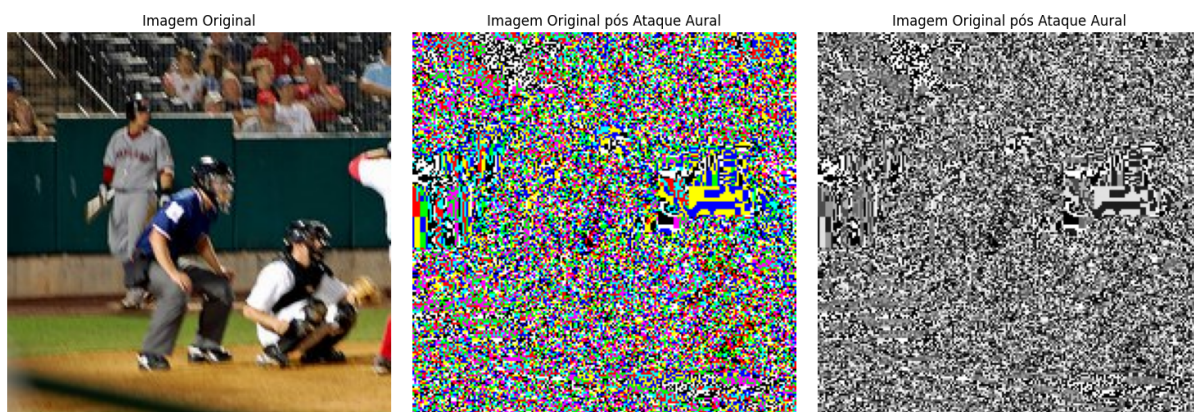
Fonte: Autor.

Figura 30 – Imagem LSB Proposta após esteganálise aural



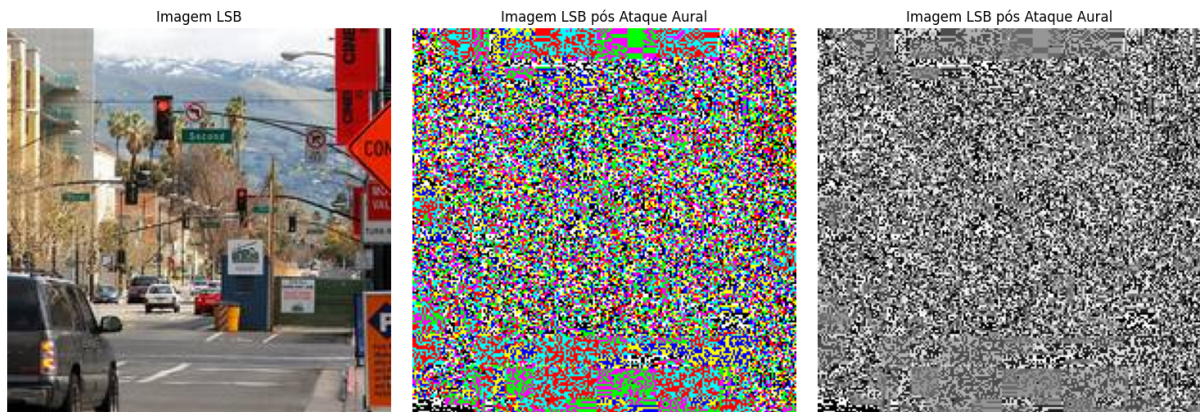
Fonte: Autor.

Figura 31 – Imagem original após esteganálise aural



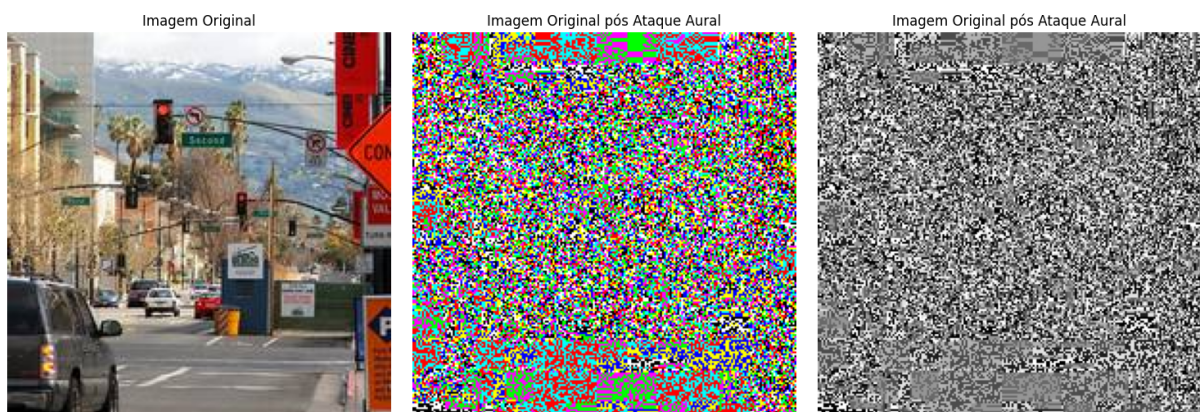
Fonte: Autor.

Figura 32 – Imagem LSB Proposta após esteganálise aural



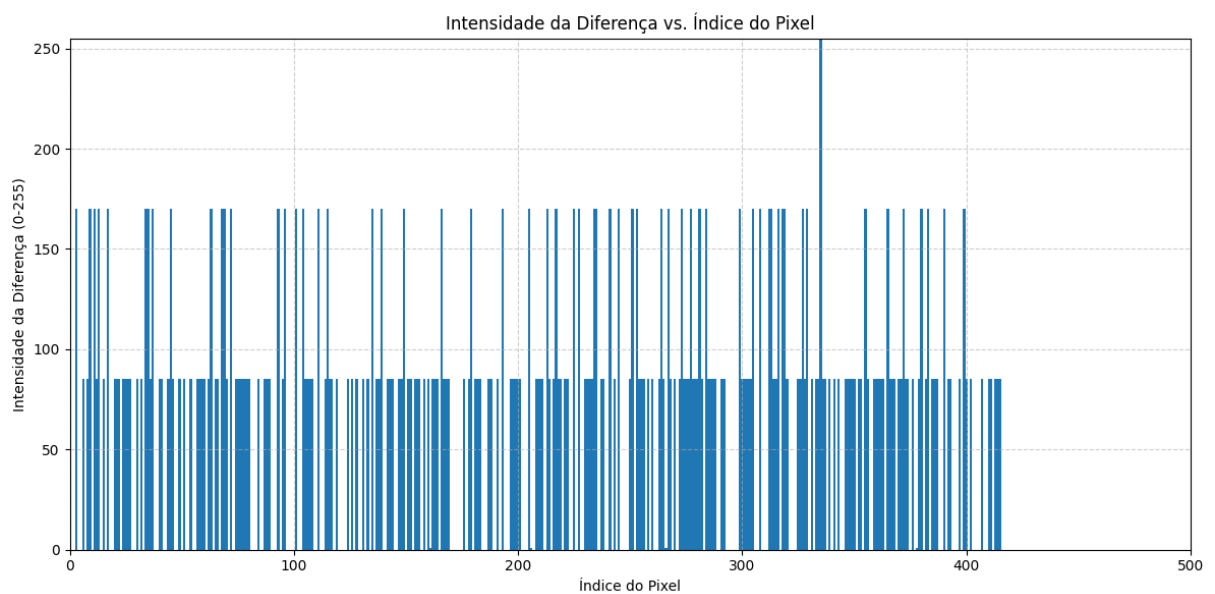
Fonte: Autor.

Figura 33 – Imagem original após esteganálise aural



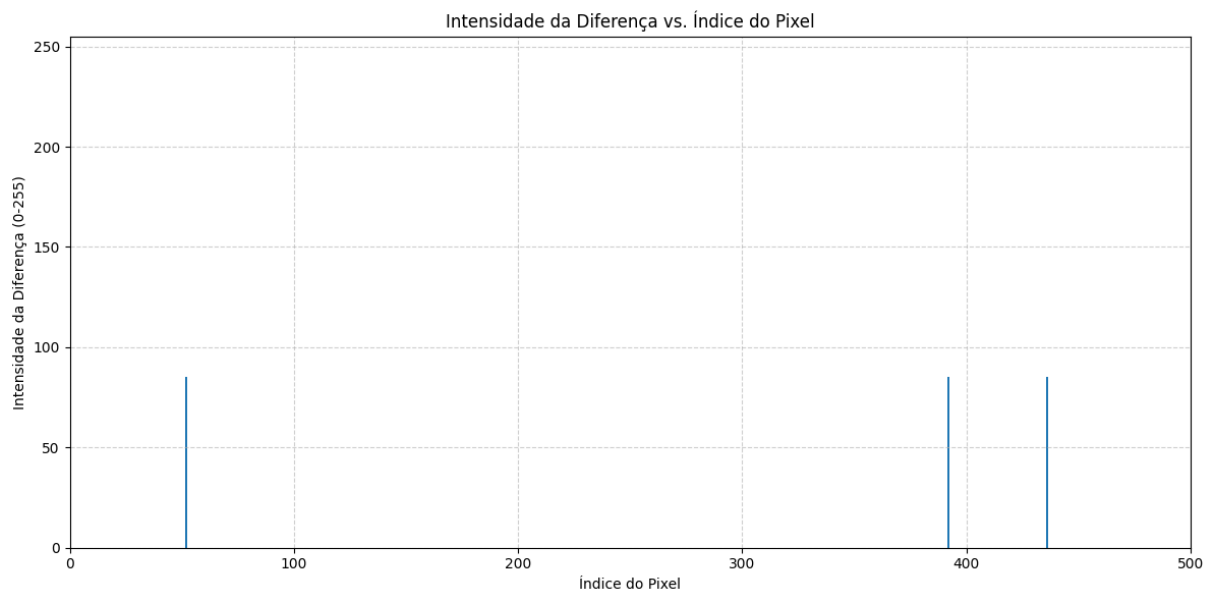
Fonte: Autor.

Figura 34 – Gráfico da intensidade da diferença de pixels na imagem do modelo LSB comum (Teste com a mensagem 1)



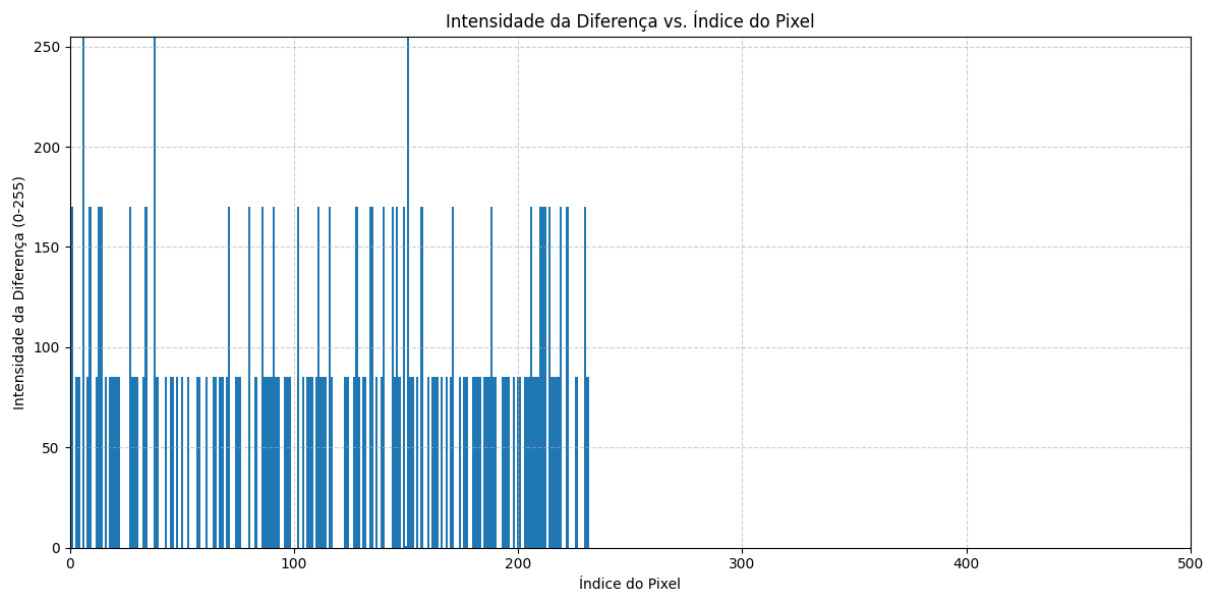
Fonte: Autor.

Figura 35 – Gráfico da intensidade da diferença de pixels na imagem do modelo LSB proposto (Teste com a mensagem 1)



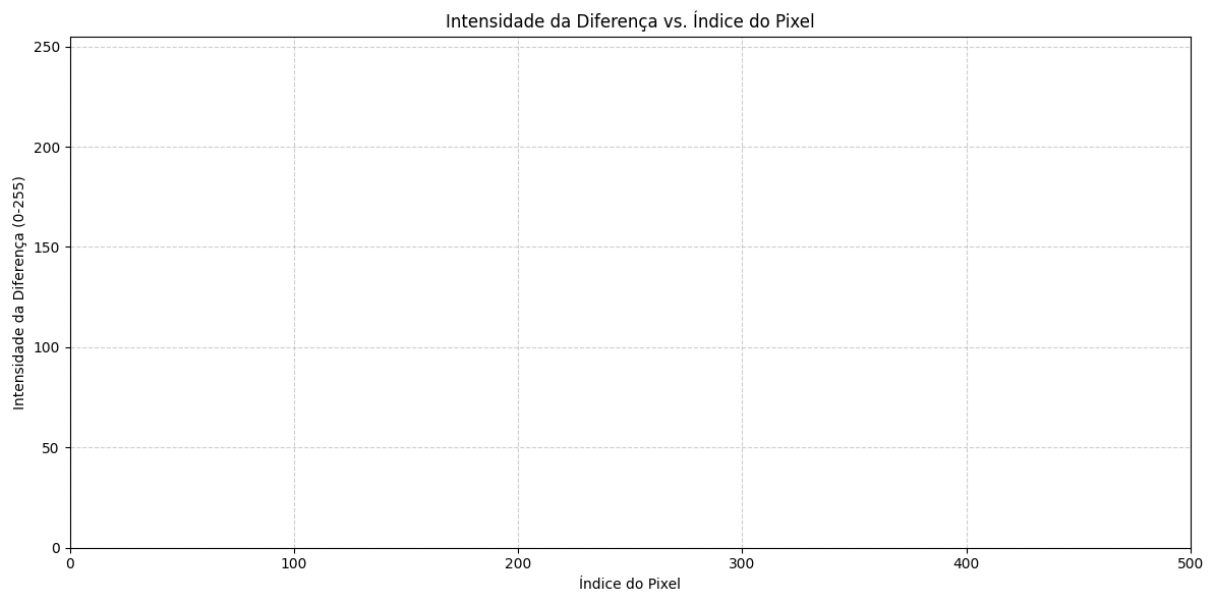
Fonte: Autor.

Figura 36 – Gráfico da intensidade da diferença de pixels na imagem do modelo LSB comum (Teste com a mensagem 2)



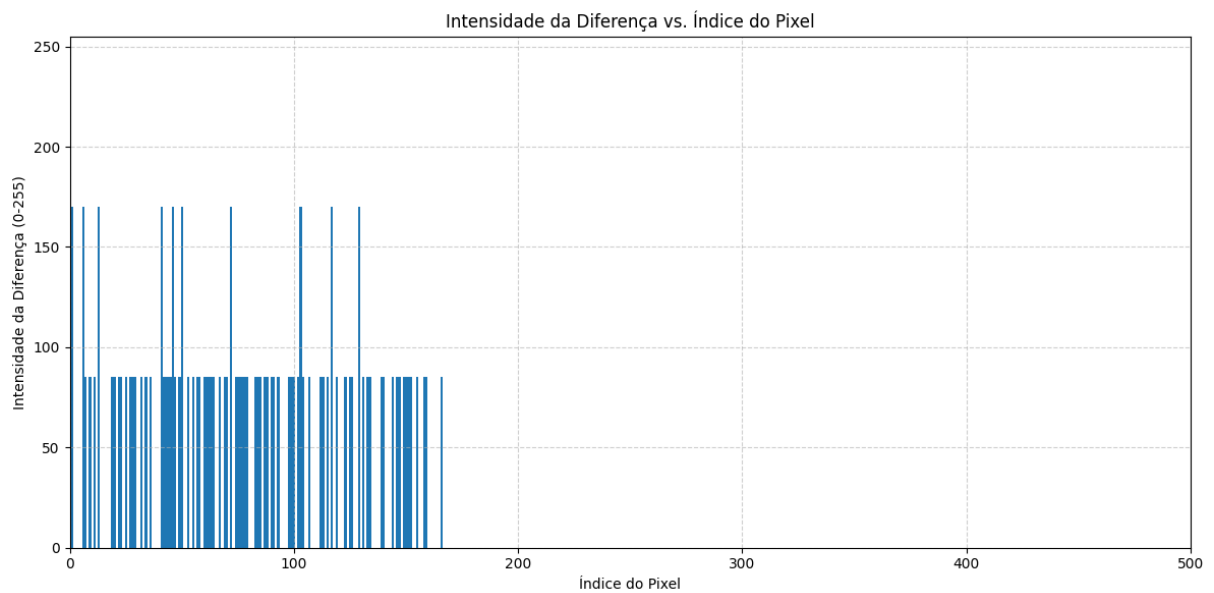
Fonte: Autor.

Figura 37 – Gráfico da intensidade da diferença de pixels na imagem do modelo LSB proposto (Teste com a mensagem 2)



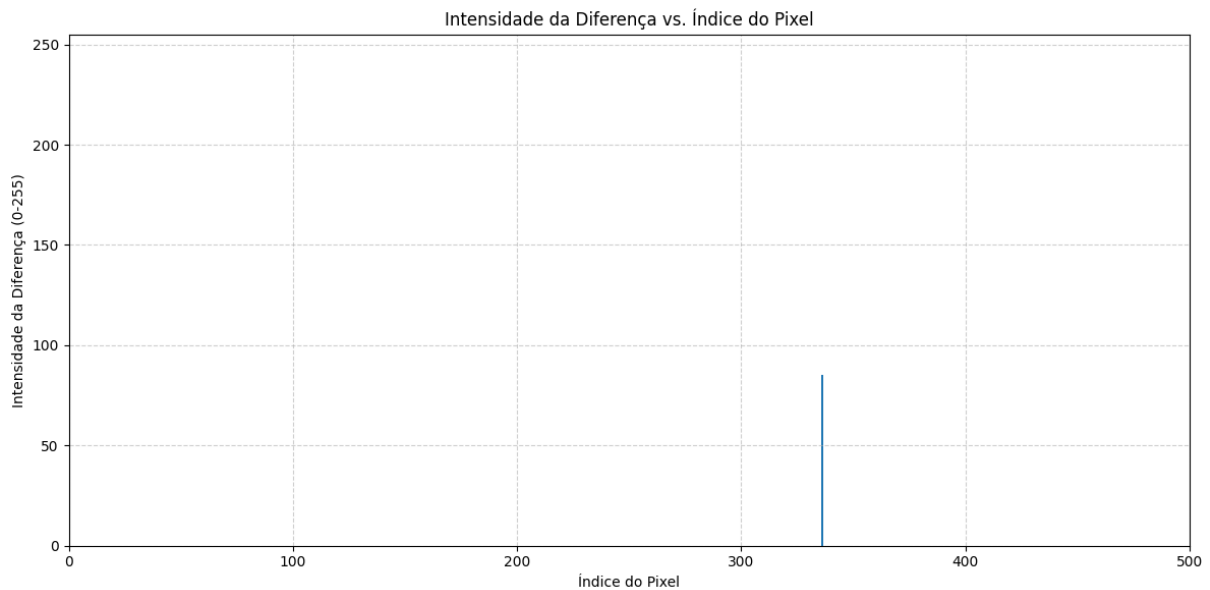
Fonte: Autor.

Figura 38 – Gráfico da intensidade da diferença de pixels na imagem do modelo LSB comum (Teste com a mensagem 3)



Fonte: Autor.

Figura 39 – Gráfico da intensidade da diferença de pixels na imagem do modelo LSB proposto (Teste com a mensagem 3)



Fonte: Autor.

4.3 Medição Objetiva

O PSNR (*Peak Signal-to-Noise Ratio*) é uma das métricas padrão utilizadas na literatura de processamento de imagens para quantificar a fidelidade estatística do sinal. No contexto da esteganografia, ele avalia a discrepância absoluta entre a imagem original e a imagem resultante, tratando a mensagem oculta como um ruído introduzido no sistema.

Valores altos de PSNR indicam que as alterações nos bits foram mínimas no processo de esteganografia. Assim, a esteganografia torna-se imperceptível para o olho humano se o PSNR for superior a 32dB, Sajasi e Eftekhari-Moghadam (2013). Alguns artigos citados neste trabalho, como o trabalho de Ibanez, Djamal e Najmurokhman (2018) e o artigo elaborado por Sari et al. (2022), utilizam essa métrica na validação de seus modelos.

Entretanto, dado que o sistema visual humano é mais sensível a padrões estruturais do que a erros absolutos de pixels, emprega-se de forma complementar o SSIM (*Structural Similarity Index*). Segundo Li et al. (2017), o SSIM é um indicador amplamente utilizado para mensurar a semelhança entre duas imagens. O SSIM é uma métrica que geralmente opera no intervalo [0, 1]. Quanto mais próximo de 1 for o valor do SSIM, menor a discrepância estrutural, indicando que as alterações podem ser consideradas imperceptíveis.

Os valores resultantes dos testes com quatro mensagens do banco de dados estão apresentados nas Tabelas 6 e 7. Os três primeiros experimentos utilizaram as imagens submetidas à esteganálise visual, apresentadas aqui nos testes anteriores, enquanto o último aplicou a chave selecionada para a maior mensagem do *dataset* do Reddit.

Tabela 6 – Valores de PSNR comparados entre o LSB Proposto e o LSB padrão.

	Tamanho da Mensagem	LSB Proposto (PSNR)	LSB padrão (PSNR)
1	912 bits	73,23 dB	70,89 dB
2	496 bits	76,14 dB	73,47 dB
3	352 bits	77,99 dB	75,19 dB
4	2160 bits	69,16 dB	67,53 dB

Fonte: Autor.

Tabela 7 – Valores de SSIM comparados entre o LSB Proposto e o LSB padrão.

	Tamanho da Mensagem	LSB Proposto (SSIM)	LSB padrão (SSIM)
1	912 bits	0.99999	1.00000
2	496 bits	1.00000	1.00000
3	352 bits	1.00000	1.00000
4	2160 bits	0.99995	0.99989

Fonte: Autor.

4.4 Algoritmos de Busca e Otimização

Com os testes do modelo LSB demonstrando a eficácia na implementação de uma busca pela imagem de cobertura ideal em conjunto com uma estratégia de inserção de bits, seguiu-se para a otimização do sistema de busca da chave LSB mais próxima do ideal. Como o modelo final se mostrou mais eficiente, o conjunto de *bitset* utilizado nessa fase do trabalho foi o gerado pelo modelo mais robusto, com mensagens maiores e chaves de inserção maiores, elevando o custo de processamento para a determinação da chave ideal para cada imagem.

Em virtude desse custo computacional, o espaço de busca foi restringido. Do total original de 25 milhões, as entradas dos algoritmos foram limitadas a 1,5 milhão de possibilidades nos primeiros experimentos. Assim, método de Força Bruta foi executado novamente, para equilibrar os resultados, tendo seu espaço de soluções reduzido as 1,5 milhão de possibilidades iniciais do arquivo de *bitset*.

Nesta etapa, executaram-se o algoritmo Guloso e a DFS, realizaram-se 30 execuções sequenciais para cada algoritmo, em cada uma delas, buscou-se no subconjunto limitado do *bitset* a melhor estratégia para cada uma das 100 mensagens selecionadas do Reddit. A Tabela 8 apresenta os resultados médios finais. As métricas adotadas para a análise foram a menor Distância de Hamming média alcançada e o tempo médio de execução por busca (em segundos).

Tabela 8 – Resultados dos Algoritmos de Busca no Espaço Limitado

	Força Bruta	DFS	Guloso
Menor Distância de Hamming:	356,56	356,56	415,76
Tempo de Execução (s):	28,27	9,47	0,00001

Fonte: Autor.

As implementações das meta-heurísticas foram avaliadas em etapas, considerando-se a importância da parametrização, processo de seleção dos valores ideais para os parâmetros de configuração que regem o comportamento do algoritmo de busca. Trata-se de uma etapa determinante, pois a calibragem correta pode conduzir a soluções de qualidade superior e maior eficiência, ao passo que valores inadequados podem resultar em desempenho insatisfatório ou na falha em convergir para uma boa solução.

Esse processo foi aplicado tanto ao algoritmo baseado em ACO quanto ao GA. Com o objetivo de assegurar uma comparação inicial equitativa, definiram-se parâmetros comuns a ambos. Estabeleceu-se a premissa inicial de que a quantidade de formigas no ACO seria equivalente ao tamanho da população no GA. Assim, referenciando a implementação sugerida por Linden (2008), fixou-se o contingente de 100 agentes (formigas/indivíduos) e determinou-se um teste inicial de 100 ciclos (iterações ou gerações).

Esse critério foi adotado apenas nos experimentos iniciais. Após a coleta dos primeiros resultados, com o número de ciclos e indivíduos (ou formigas) fixados em 100, outros valores foram abordados. Na parametrização dos ACOs, extrapolou-se o número de formigas de 100 para 1000, uma vez que ampliar o número de interações mostrou-se mais custoso computacionalmente. Em contrapartida, no GA, o número de gerações foi ampliado de 100 para 5000.

Tomou-se como guia de implementação o estudo feito por Gaertner e Clark (2005), que analisou os efeitos da variação de β em implementações de ACO, utilizando em seus experimentos o intervalo $0 < \beta < 15$. Seus resultados indicam que um valor elevado para β aproxima o comportamento do algoritmo ao de uma busca gulosa, tornando-o mais rápido, porém inadequado quando o alvo é a solução ótima global. Portanto, para o contexto deste trabalho, valores menores para β mostram-se mais apropriados.

Por outro lado, o *alfa* determina o quanto a formiga deve confiar nos caminhos que foram bem-sucedidos no passado, ou seja, no depósito de feromônio de cada nó. Para valores altos, espera-se uma convergência muito rápida, mas há risco de estagnação prematura. Com valores próximos de 0, o algoritmo se comporta como uma busca gulosa estocástica (se $\beta > 0$).

Os resultados da implementação dos parâmetros do algoritmo ACO sem evaporação, cujos experimentos foram pautados nas variações da quantidade de formigas e nos parâmetros α e β . Os valores empregados e os resultados finais obtidos para cada um deles são apresentados na Tabela 9. Adicionalmente aos resultados absolutos, as Figuras 40 e 41 apresentam gráficos de convergência para cada configuração do ACO sem evaporação.

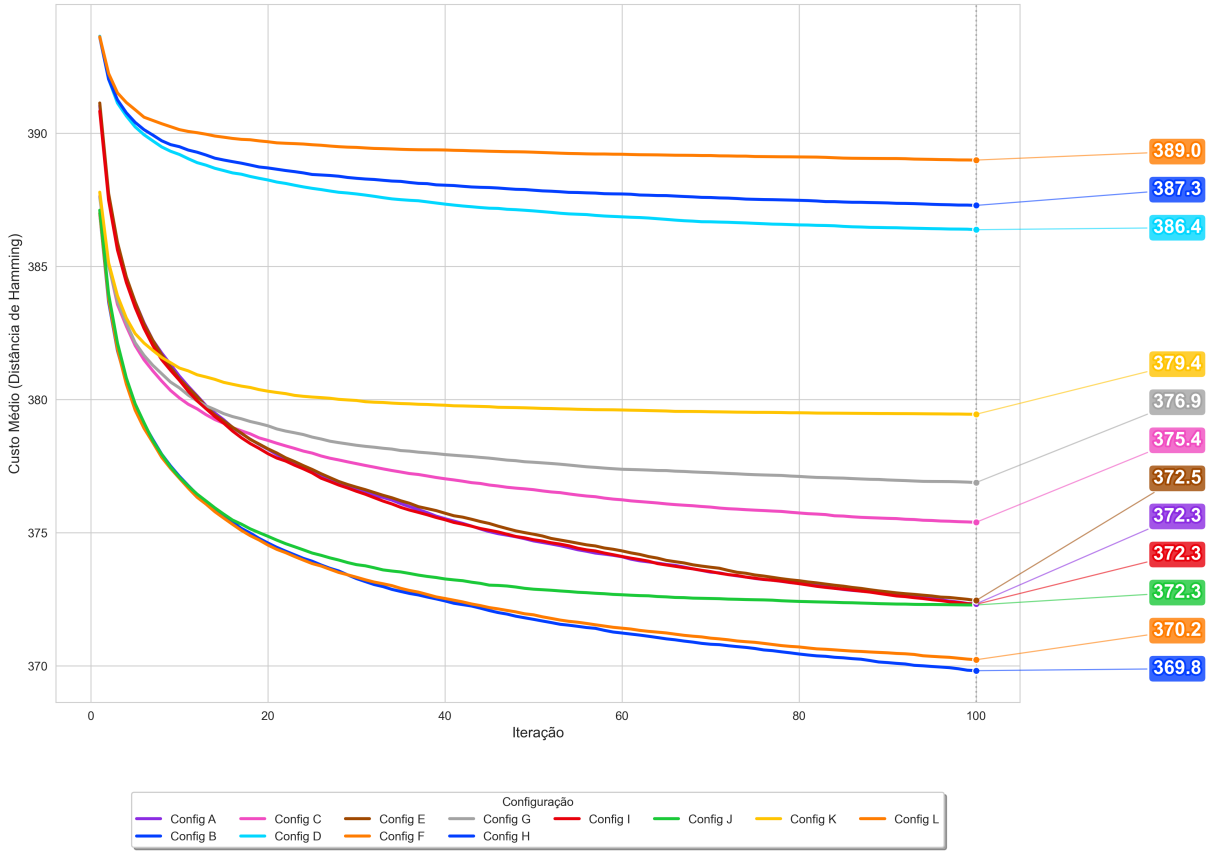
Os dados evidenciam que um β elevado induz à convergência prematura, ao passo que os melhores desempenhos concentram-se nos valores de $\beta = 0$ ou $\beta = 1$. Paralelamente, a análise do parâmetro α revela que sua calibração é fundamental para o equilíbrio da busca, valores inadequados tendem a dispersar excessivamente os agentes ou viciar o algoritmo em ótimos locais.

Tabela 9 – Resultados da parametrização do ACO sem Evaporação (destaque para a melhor configuração).

Grupo	Alfa (α)	Beta (β)	Custo Médio		Tempo Médio (s)	
			100 For.	1000 For.	100 For.	1000 For.
A	0,5	0,0	372,32	365,07	0,73	4,66
B	0,5	1,0	369,82	364,65	1,04	3,21
C	0,5	2,0	375,39	372,36	0,84	3,28
D	0,5	5,0	386,38	384,35	0,85	2,61
E	1,0	0,0	372,45	365,07	0,41	4,00
F	1,0	1,0	370,23	367,36	0,32	4,88
G	1,0	2,0	376,88	375,17	0,28	2,68
H	1,0	5,0	387,29	385,37	0,26	4,54
I	2,0	0,0	372,31	365,02	0,41	8,60
J	2,0	1,0	372,28	371,76	0,30	2,85
K	2,0	2,0	379,45	379,35	0,27	2,44
L	2,0	5,0	388,99	387,44	0,25	3,35

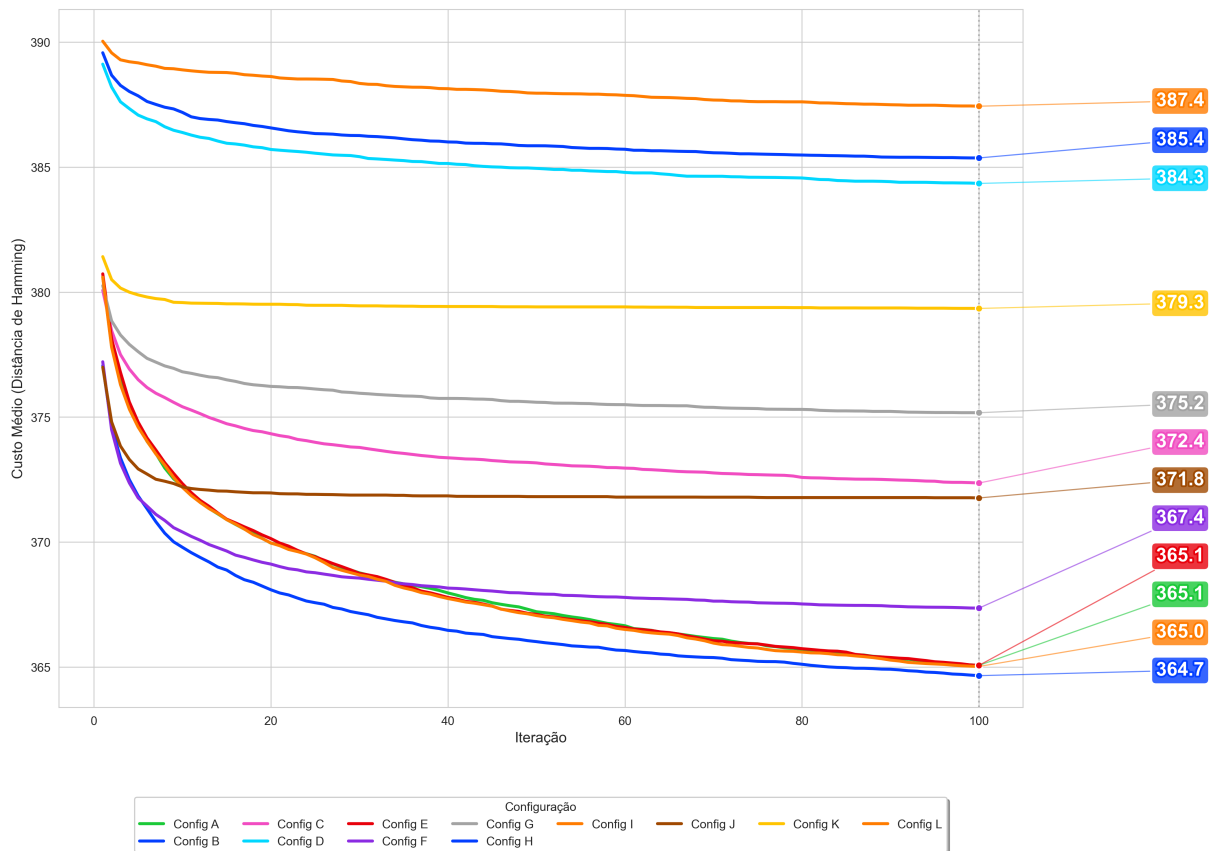
Fonte: Autor.

Figura 40 – Curva de Convergência Média da parametrização do ACO sem evaporação utilizando 100 Formigas a cada iteração



Fonte: Autor.

Figura 41 – Curva de Convergência Média da parametrização do ACO sem evaporação utilizando 1000 Formigas a cada iteração



Fonte: Autor.

Demonstra-se, assim, que a ponderação correta do feromônio é tão crítica quanto a decisão heurística imediata. Tal fato é confirmado pelos dados, visto que a elevação de α ocasionou uma leve degradação nos resultados, sugerindo que o excesso de confiança no feromônio depositado pelas formigas artificiais limitou a busca a soluções ótimas.

Observando a Tabela 9, nota-se que o aumento no número de formigas disponíveis por ciclo melhora os resultados obtidos nos valores absolutos da Distância de Hamming Média, mas piora consideravelmente o custo computacional, o que acarreta no aumento do tempo médio para encontrar uma solução viável. É importante avaliar que o custo de tempo é elevado em decorrência de o poder computacional da máquina utilizada nos primeiros experimentos ser inferior.

Nos testes da implementação de ACO com a adição de evaporação de feromônio, a variação dos parâmetros utilizados seguiu a mesma lógica de variação dos testes da ACO sem evaporação. Adicionalmente, variou-se o fator de evaporação ρ , visando analisar sua influência sobre os resultados finais e a velocidade de convergência. Os resultados dos conjuntos de evaporação utilizados podem ser observados na Tabela 10.

A representação gráfica da convergência de cada grupo de parâmetros é apresentada nas

Figuras 42 e 43. Para esta implementação com evaporação, valores reduzidos de β e ρ resultaram em um desempenho absoluto superior, bem como em uma curva de convergência mais adequada.

Conforme observado nos resultados do ACO com evaporação, o aumento no número de formigas também resultou no aumento considerável do tempo de execução na busca de um resultado adequado para cada mensagem. Além disso, os resultados obtidos nessa implementação não se mostraram estatisticamente superiores à implementação sem evaporação, uma vez que o tempo de execução se mostrou muito semelhante, assim como os resultados da menor Distância de Hamming.

Na etapa de parametrização do GA, avaliaram-se diferentes intervalos para os operadores de mutação ($0,01 < P_m < 0,30$) e elitismo ($0,05 < P_e < 0,20$). Segundo Linden (2008), a mutação é crucial para a manutenção da diversidade genética. Com base nessa premissa, buscou-se equilibrar a exploração do espaço de busca, via mutação, com a preservação dos melhores indivíduo, via elitismo. Adicionalmente, a análise foi expandida para cinco mil gerações para verificar o comportamento de convergência a longo prazo. Os resultados obtidos constam na Tabela 11.

As curvas de convergência ao longo das cinco mil gerações são apresentadas na Figura 44. Observa-se que os melhores desempenhos foram obtidos com taxas de mutação mais elevadas aliadas a um elitismo moderado. Isso indica que uma maior exploração do espaço de busca favorece a obtenção de melhores resultados, confirmando o observado nos resultados dos algoritmos baseados no ACO.

Após o processo de parametrização para cada algoritmo, conduziu-se um novo experimento abrangendo o espaço completo de estratégias do gerador de chaves. O objetivo consistiu em avaliar o desempenho dos métodos ao processarem a totalidade das estratégias de inserção disponíveis, contendo 25 milhões de chaves, utilizando uma outra máquina com infraestrutura de alto desempenho, mencionada no início desta seção.

Observou-se uma melhora geral no desempenho dos algoritmos. O acesso a um espaço de busca ampliado permitiu encontrar soluções com menor Distância de Hamming, enquanto o maior poder computacional reduziu significativamente o tempo de execução.

A análise da parametrização indicou que, para os algoritmos baseados em ACO, 100 ciclos são suficientes para a convergência. Em contrapartida, para o GA, limitar a execução a 100 gerações restringe seu desempenho, visto que este método possui um custo computacional por ciclo inferior, permitindo mais iterações no mesmo intervalo de tempo.

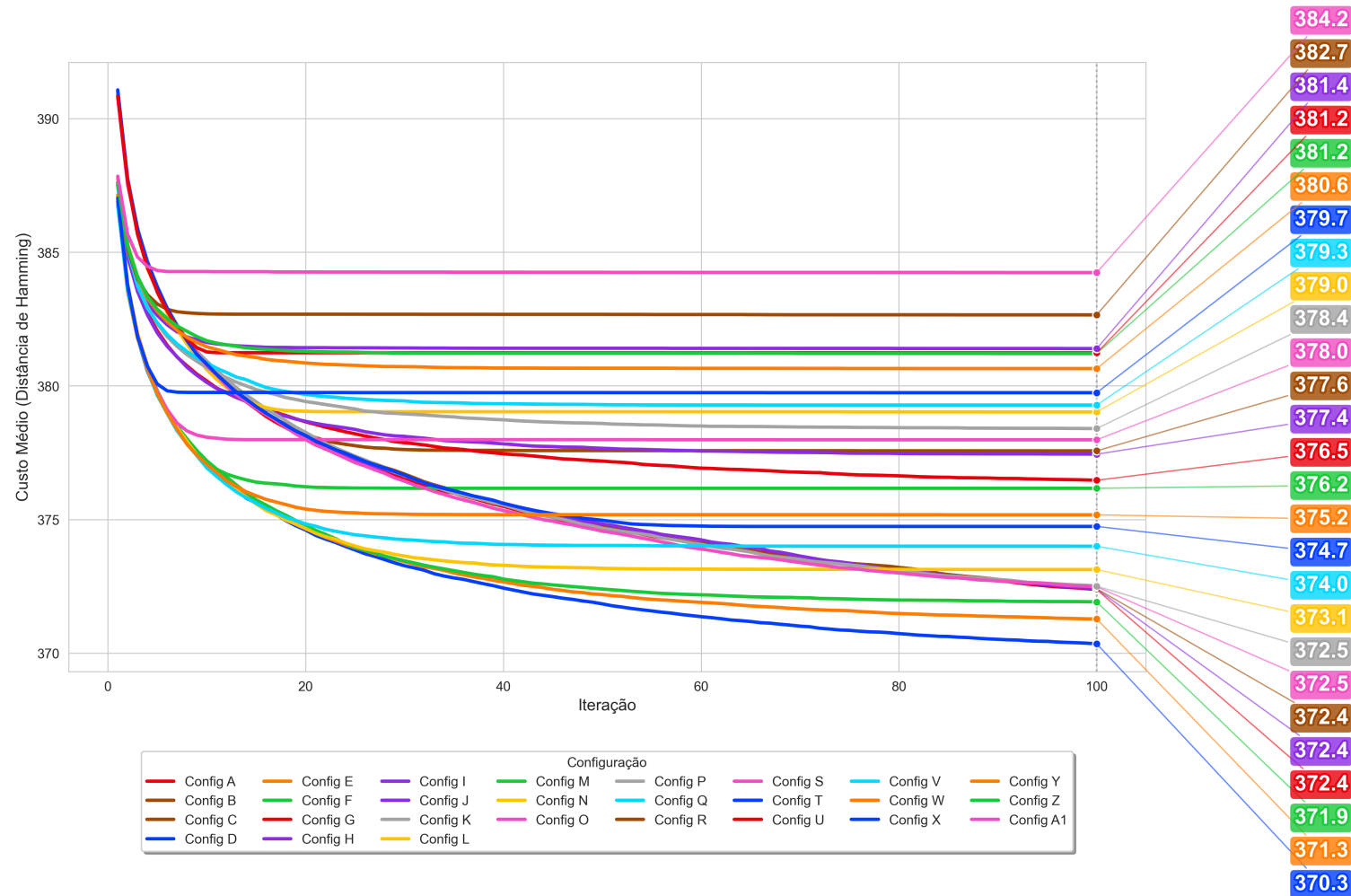
Ao explorar o potencial máximo de cada método, estendendo o GA para 5 mil gerações com 100 indivíduos, notou-se uma evolução significativa na qualidade das soluções. De forma análoga, no ACO, o aumento do número de formigas por iteração, mantendo-se os 100 ciclos, resultou em melhores soluções finais. Os melhores desempenhos de cada algoritmo estão detalhados nas Tabelas 12 e 13.

Tabela 10 – Resultados da parametrização do ACO com Evaporação, comparando o desempenho entre iterações utilizando 100 e 1000 formigas.

Grupo	Alfa (α)	Beta (β)	Rho (ρ)	Custo Médio		Tempo Médio (s)	
				100 For.	1000 For.	100 For.	1000 For.
A	0,5	0,0	0,05	372,38	364,99	0,68	4,91
B	0,5	0,0	0,10	372,41	364,96	0,54	4,91
C	0,5	0,0	0,20	377,57	366,43	0,66	4,79
D	0,5	1,0	0,05	370,35	365,59	0,43	4,06
E	0,5	1,0	0,10	371,28	366,31	0,42	4,01
F	0,5	1,0	0,20	376,17	367,85	0,38	3,83
G	0,5	5,0	0,05	376,47	384,55	1,19	3,53
H	0,5	5,0	0,10	377,44	384,92	0,86	9,31
I	0,5	5,0	0,20	381,40	385,39	1,14	6,84
J	1,0	0,0	0,05	372,40	365,16	0,52	4,96
K	1,0	0,0	0,10	372,50	365,55	1,46	4,94
L	1,0	0,0	0,20	379,03	367,31	1,13	4,80
M	1,0	1,0	0,05	371,91	368,49	0,41	3,80
N	1,0	1,0	0,10	373,13	369,22	1,29	3,73
O	1,0	1,0	0,20	377,98	370,68	1,07	3,52
P	1,0	5,0	0,05	378,40	385,93	0,37	3,52
Q	1,0	5,0	0,10	379,28	386,26	0,94	3,52
R	1,0	5,0	0,20	382,66	386,77	0,63	3,36
S	2,0	0,0	0,05	372,48	366,85	0,50	4,52
T	2,0	0,0	0,10	374,74	368,33	0,83	4,30
U	2,0	0,0	0,20	381,23	369,90	0,40	4,03
V	2,0	1,0	0,05	374,00	372,16	0,37	3,48
W	2,0	1,0	0,10	375,17	372,85	0,36	3,46
X	2,0	1,0	0,20	379,74	373,50	0,51	3,30
Y	2,0	5,0	0,05	380,65	387,63	0,36	3,39
Z	2,0	5,0	0,10	381,21	387,73	0,35	3,37
A1	2,0	5,0	0,20	384,24	388,33	0,33	3,21

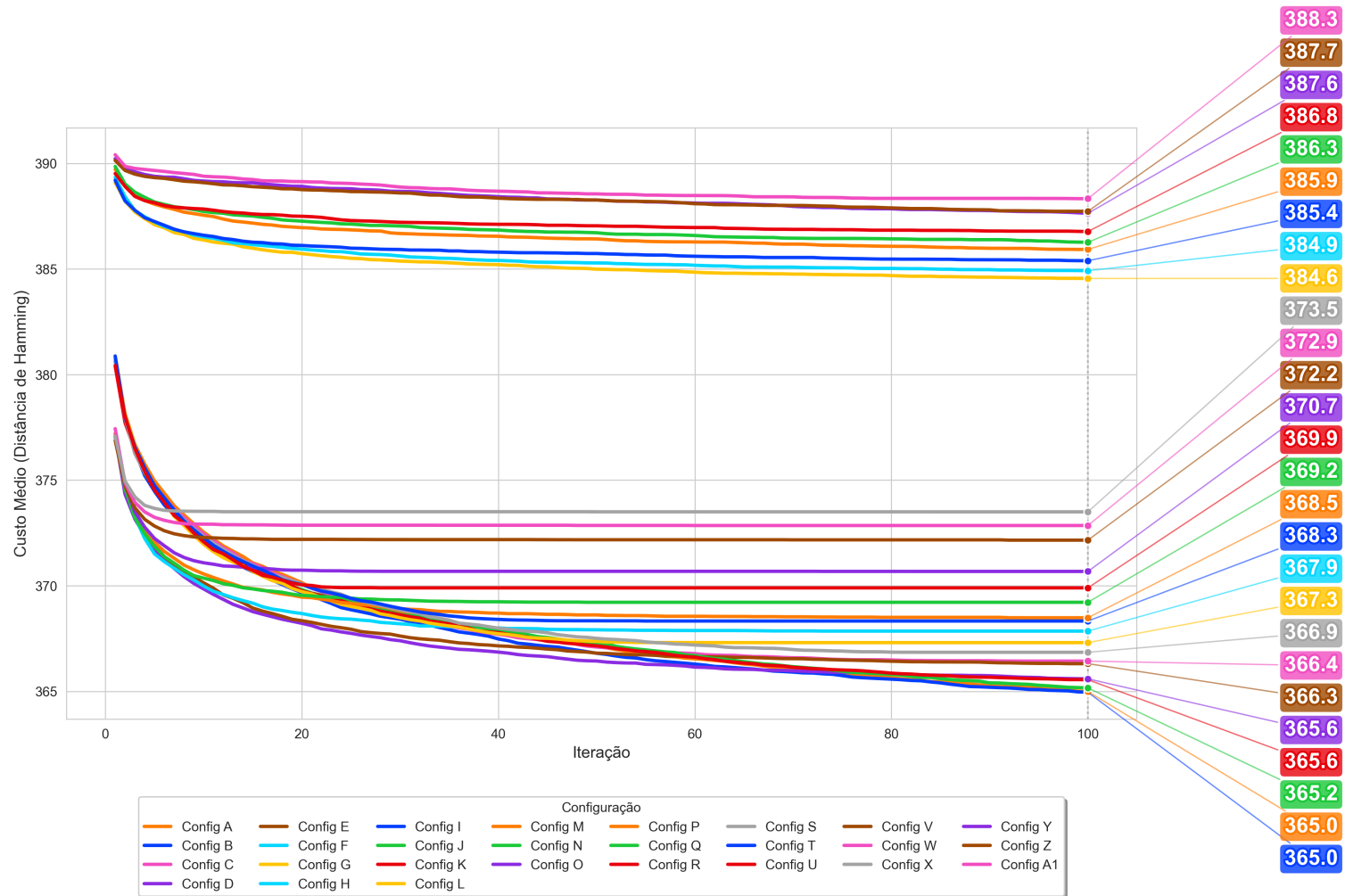
Fonte: Autor.

Figura 42 – Curva de Convergência Média da parametrização do ACO com evaporação utilizando 100 Formigas a cada iteração



Fonte: Autor.

Figura 43 – Curva de Convergência Média da parametrização do ACO com evaporação utilizando 1000 Formigas a cada iteração



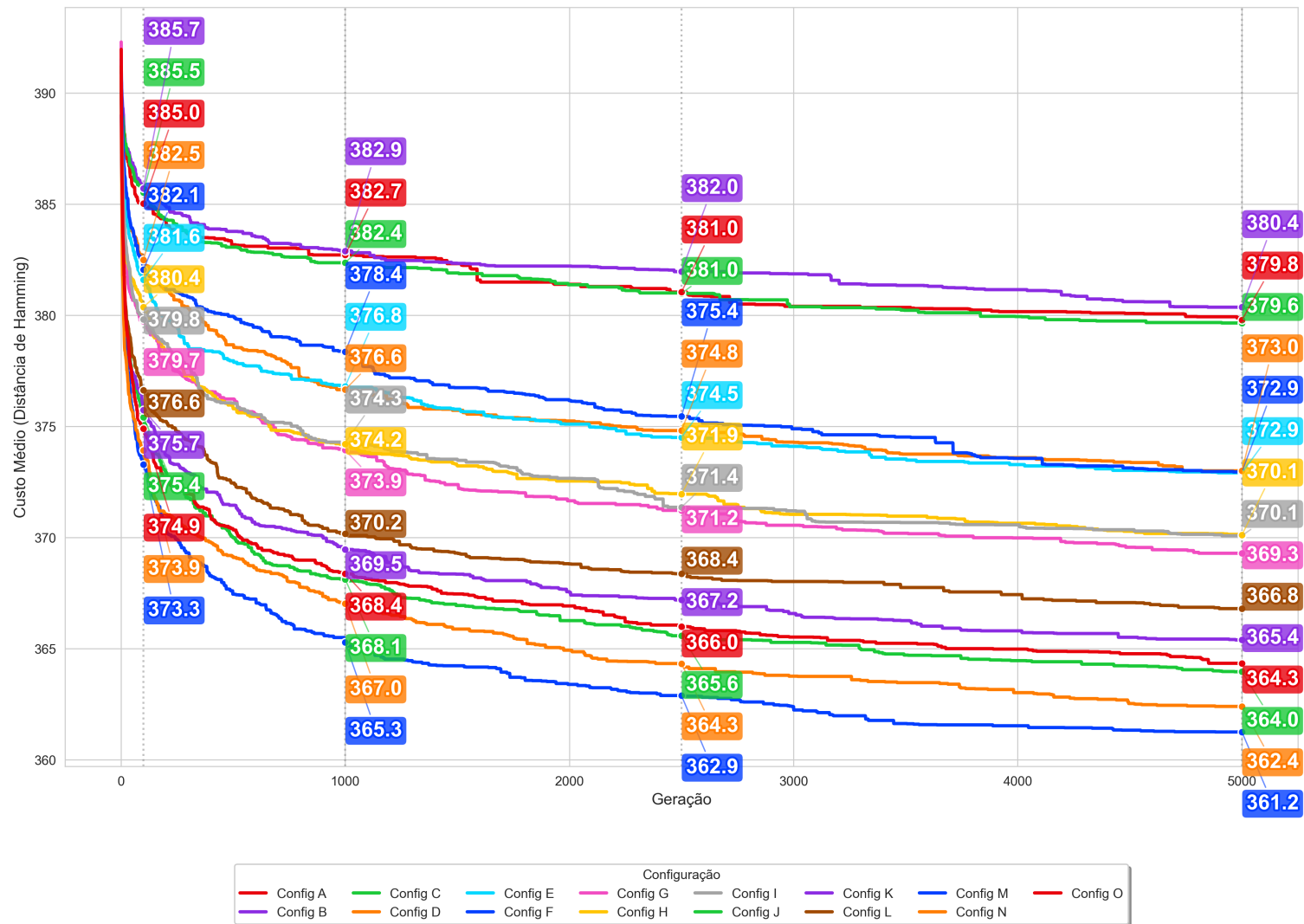
Fonte: Autor.

Tabela 11 – Detalhamento da evolução do Algoritmo Genético, comparando Custo e Tempo ao longo das 5000 gerações,

Grupo	Mutação (P_m)	Elistismo (P_e)	Custo Médio				Tempo Acumulado (s)			
			100 Gen.	100 Gen.	2500 Gen.	5000 Gen.	100 Gen.	1000 Gen.	2500 Gen.	5000 Gen.
A	0,01	0,05	385,03	382,71	381,04	379,78	0,03	0,27	0,66	1,32
B	0,01	0,10	385,71	382,89	381,96	380,36	0,03	0,25	0,63	1,25
C	0,01	0,20	385,52	382,36	380,98	379,65	0,02	0,23	0,56	1,12
D	0,05	0,05	382,49	376,65	374,81	373,00	0,04	0,34	0,84	1,68
E	0,05	0,10	381,60	376,80	374,50	372,91	0,03	0,32	0,80	1,59
F	0,05	0,20	382,05	378,35	375,45	372,95	0,03	0,29	0,71	1,42
G	0,10	0,05	379,73	373,93	371,21	369,29	0,04	0,42	1,04	2,07
H	0,10	0,10	380,36	374,20	371,95	370,11	0,04	0,40	0,99	1,97
I	0,10	0,20	379,81	374,27	371,36	370,08	0,04	0,36	0,89	1,78
J	0,20	0,05	375,41	368,12	365,58	363,96	0,07	0,66	1,65	3,29
K	0,20	0,10	375,74	369,46	367,19	365,39	0,06	0,58	1,43	2,86
L	0,20	0,20	376,63	370,17	368,37	366,80	0,05	0,48	1,19	2,38
M	0,30	0,05	373,29	365,28	362,89	361,25	0,08	0,76	1,89	3,78
N	0,30	0,10	373,92	367,03	364,32	362,40	0,07	0,70	1,75	3,49
O	0,30	0,20	374,90	368,37	365,99	364,33	0,06	0,59	1,48	2,96

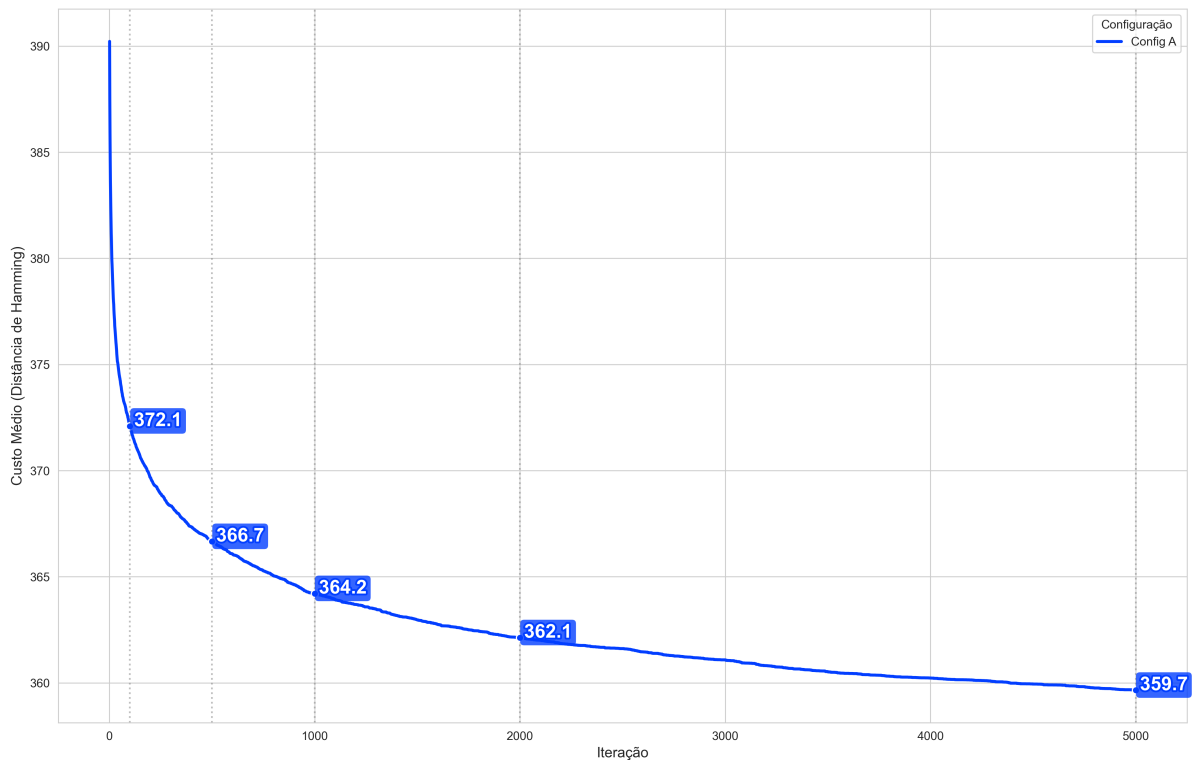
Fonte: Autor.

Figura 44 – Curva de Convergência Média do Grupo de Configurações do GA



Fonte: Autor.

Figura 45 – Curva de Convergência Média do GA com a melhor parametrização no experimento com 25 milhões de possibilidades ao longo de 5 mil gerações



Fonte: Autor.

A partir dos resultados obtidos, definindo que uma solução promissora para a otimização com os algoritmos bioinspirados apresenta bons resultados dentro de 1 segundo, as melhores configurações de cada algoritmo bioinspirado foram selecionadas e seus resultados apresentados na Tabela 14 em conjunto com os resultados dos outros algoritmos, utilizando o método de Força Bruta como linha de base, uma vez que este garante a solução ótima global. A coluna Perda representa o percentual de aumento na Distância de Hamming em relação ao ótimo.

A coluna *Speedup* na Tabela 14 quantifica quantas vezes o algoritmo foi mais rápido que o Força Bruta. Observa-se um *trade-off* claro entre qualidade e velocidade. Enquanto o algoritmo DFS obteve a menor perda de qualidade (0,57%), seu ganho de tempo foi modesto (2,3×). Em contrapartida, as meta-heurísticas (ACO e GA) ofereceram um equilíbrio atrativo, acelerando o processo em cerca de 100 vezes com uma perda de qualidade inferior a 4%.

Para concluir a análise, a Figura 46 apresenta o comparativo de eficiência entre o limite superior do modelo proposto (resultado ótimo), o desempenho das meta-heurísticas implementadas (GA, ACO com Evaporação e ACO sem Evaporação) e a abordagem convencional. A análise fundamentou-se nas menores Distâncias de Hamming obtidas para cada uma das 100 mensagens do banco de teste, considerando as configurações de parâmetros mais eficazes de cada algoritmo. Essa sistematização permite visualizar a proximidade dos algoritmos bioinspirados em relação ao teto teórico de otimização, a consistência e a variabilidade dos resultados.

Tabela 12 – Resultados final dos algoritmos inspirados na ACO, comparando o desempenho no experimento com 25 milhões de possibilidades utilizando 100 ciclos.

ACO	Alfa (α)	Beta (β)	Rho (ρ)	Custo Médio		Tempo Médio (s)	
				100 For.	1000 For.	100 For.	1000 For.
Sem Evaporação	0.5	1,0	-	366,92	362,13	0,11	0,93
Com Evaporação	0,5	0,0	0,05	371,90	363,93	0,54	1,61

Fonte: Autor.

Tabela 13 – Resultado final da melhor hiperparametrização do algoritmo inspirado no GA ao longo de 5 mil gerações.

Geração	Custo Médio	Tempo Médio (s)
100	372,10	0,05
500	366,66	0,25
1000	364,21	0,50
2000	362,13	1,00
5000	359,65	2,51

Fonte: Autor.

Tabela 14 – Comparativo de perda de qualidade e ganho de velocidade em relação ao método de Força Bruta

Algoritmo	Qualidade (Hamming)		Desempenho (Tempo)	
	Média	Perda (%)	Tempo (s)	Speedup ($\times$)
Brute (Base)	348,77	—	98,21	1,0 $\times$
DFS	350,77	0,57%	41,81	2,3 $\times$
ACO	362,13	3,83%	0,91	108,0 $\times$
ACO Evap	363,30	4,17%	1,61	61,0 $\times$
GA	362,10	3,82%	1,00	98,2 $\times$
Guloso	414,81	18,94%	10 ⁻⁵	9,8 · 10 ⁶ $\times$

Fonte: Autor.

A métrica de eficiência, representada no eixo vertical, quantifica a preservação da integridade dos bits da imagem original durante o processo de ocultação. Nesse contexto, um índice de 80% indica que o algoritmo foi capaz de inserir a mensagem preservando 80% dos bits de cobertura originais, ou seja, sem gerar ruído nesses pontos específicos. Portanto, o valor de 100% representa uma inserção ideal (sem alterações nos bits de cobertura), enquanto 0% denota a inversão de todos os bits correspondentes à mensagem.

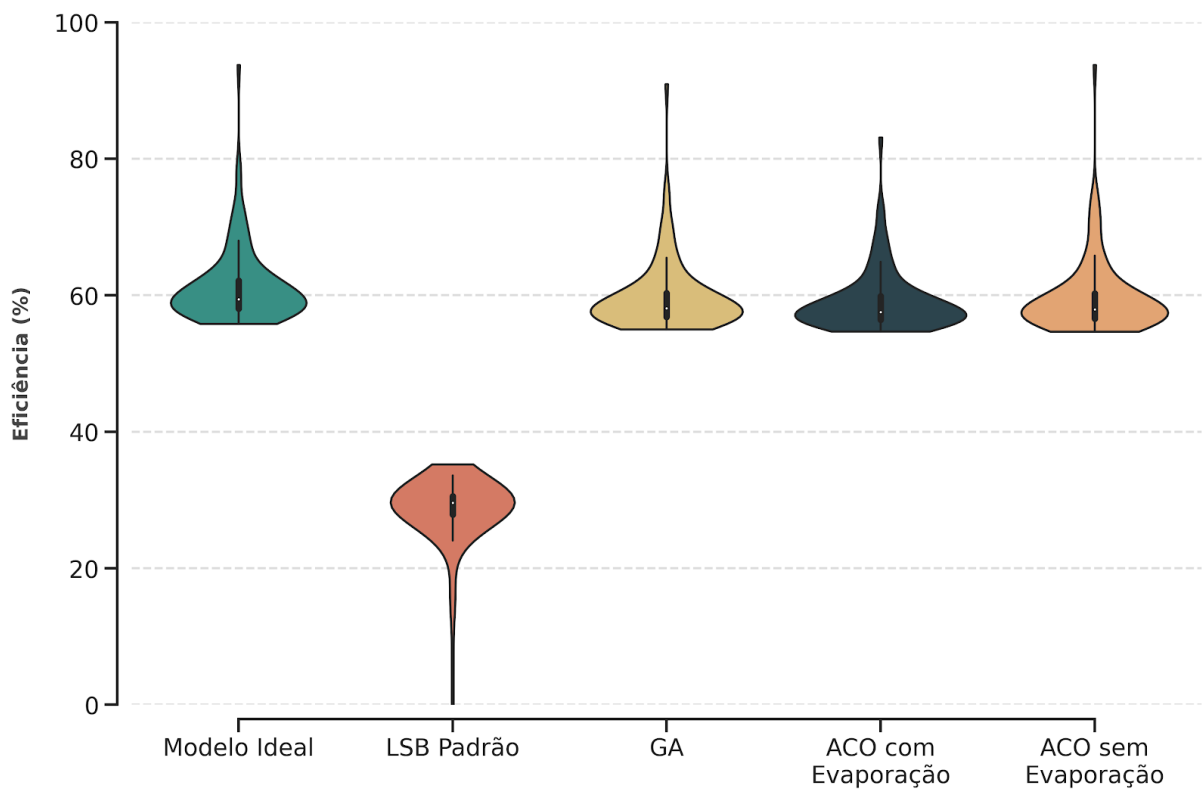
Evidencia-se uma superioridade significativa das abordagens baseadas em otimização quando comparadas ao método clássico. Os modelos bioinspirados obtiveram resultados que convergem para o desempenho do modelo proposto (Ideal), com medianas de eficiência situadas próximas a 60%. Tal comportamento sugere que estas meta-heurísticas são eficazes em identificar os melhores conjuntos de imagem e estratégia de inserção, selecionando posições onde os bits

da mensagem coincidem com os pixels originais, mitigando assim a introdução de ruído.

O método LSB Padrão, como esperado, apresentou desempenho inferior dentre as estratégias avaliadas, registrando uma eficiência mediana de aproximadamente 30%. Este dado ratifica as limitações inerentes à abordagem sequencial, a qual, desprovida de mecanismos de pré-processamento ou otimização de custo, acarreta uma taxa elevada de alterações na imagem de cobertura.

Os métodos de otimização propostos apresentaram baixa variância nos resultados, conforme evidenciado pela reduzida amplitude interquartil e pela curta extensão dos *whiskers*. Essa concentração dos dados indica que os algoritmos são estáveis e robustos, mantendo altos índices de eficiência de forma consistente, independentemente das variações no conteúdo da mensagem oculta.

Figura 46 – Comparação de eficiência (%)



Fonte: Autor.

5 CONSIDERAÇÕES FINAIS

Neste trabalho, é proposto um novo modelo de esteganografia por substituição LSB. O modelo baseia-se na implementação de chaves de esteganografia, onde os dados são inseridos na imagem de maneira aleatória e não sequencial. Para sua implementação, foi construído um arquivo com 25 milhões de possibilidades de substituição LSB. Também é proposto um método para otimizar a busca pelo conjunto de imagem mais estratégia de cobertura ideal para uma mensagem. Esse método consistiu na implementação de algoritmos bioinspirados, como o ACO e o GA, em uma árvore de busca estruturada a partir do arquivo de *bitsets*.

No que tange aos resultados, o modelo de esteganografia LSB proposto revelou-se promissor, beneficiando-se da inserção não sequencial dos bits da mensagem. Essa abordagem alinha-se ao demonstrado por Elshare e EL-Emam (2018), o qual sustenta que a inserção não linear amplia a imperceptibilidade do método LSB. O modelo demonstrou superioridade frente às técnicas de esteganálise, reduzindo a detectabilidade da ocultação em comparação ao LSB tradicional. A validação do método foi corroborada pela análise da diferença de pixels entre a imagem de cobertura e a estego-imagem (imagem resultante), métrica também adotada por Abbas et al. (2015).

Em conjunto, os resultados obtidos nas métricas PSNR e SSIM, utilizadas pelos autores Sari et al. (2022), Ibanez, Djamal e Najmurokhman (2018) e Li et al. (2017), mostraram-se satisfatórios. O modelo proposto apresentou bons valores no teste de PSNR, com uma média de 3 dB acima do resultado das imagens geradas pelo LSB comum. Também obteve índices altos nos testes de SSIM, comprovando a eficácia do modelo em diminuir as alterações geradas pelo LSB e mantendo a qualidade da imagem. Além disso, o uso de chaves adiciona um nível de segurança extra ao modelo proposto.

O uso das técnicas de otimização ACO e GA enriquece o modelo. Analisando a implementação do ACO, os resultados apontam para um desempenho melhor da lógica sem evaporação do feromônio. O ACO sem evaporação alcançou resultados levemente superiores à implementação com evaporação, mostrando-se mais rápido, especialmente quando o número de formigas por ciclo cresce. Portanto, a implementação sem evaporação mostrou-se mais adequada para o contexto deste trabalho.

O algoritmo inspirado no GA mostrou-se igualmente promissor. No contexto do problema apresentado, a mutação surgiu inicialmente como um obstáculo, pois o algoritmo gerava soluções inválidas, fora do espaço de busca. Com a adequação do algoritmo, que, independentemente da solução gerada pela mutação, mantinha o indivíduo dentro do espaço possível, obteve-se um ótimo desempenho na exploração da Árvore de Busca. Com um número de indivíduos fixado em 100, o algoritmo mantém bons resultados com o operador de elitismo moderado (5%), enquanto

força a população a explorar as possibilidades com uma alta taxa de mutação (30%) em conjunto com a lógica de *crossover*. O aumento das gerações mostrou que o algoritmo tende a estagnar parcialmente a partir de 2 mil gerações, já que continua evoluindo de forma gradual e muito mais lenta a partir desse ponto.

Embora os algoritmos bioinspirados nem sempre alcancem a distância de Hamming absoluta, os experimentos comprovaram que a combinação de estratégias aleatórias com um banco de imagens vasto e variado, permitindo ao algoritmo operar em toda a extensão das coberturas, proporciona um resultado esteganográfico aprimorado. Observou-se que, mesmo com o aumento no tamanho das mensagens e do espaço de busca elevando o custo computacional, as meta-heurísticas ACO e GA mantiveram uma diferença de apenas 4% em relação ao ótimo global, apresentando um tempo de processamento até 100 vezes menor que a busca exaustiva. Esse equilíbrio valida a eficiência do modelo, especialmente em cenários de alta complexidade onde a redução da latência operacional é prioritária frente ao custo mínimo de qualidade.

A abordagem apresentada neste trabalho diferencia-se de métodos convencionais encontrados na literatura ao antecipar a etapa de geração das chaves de inserção, dissociando-a do processo de ocultação da mensagem. Ao realizar o processamento das estratégias em uma fase prévia, o método tende a reduzir a latência durante a escrita dos dados e favorece a escalabilidade, viabilizando a utilização de conjuntos de imagens mais amplos. Tal característica sugere que a estrutura de árvore de busca desenvolvida pode servir como base para que trabalhos futuros avaliem o desempenho de outras meta-heurísticas ou técnicas de otimização.

Adicionalmente, a base estabelecida neste trabalho abre precedentes para investigações subsequentes que busquem elevar o nível de inteligência do modelo de pré-processamento. Como perspectiva futura, vislumbra-se a integração de técnicas de aprendizado de máquina (*Machine Learning*) para automatizar a identificação de regiões de interesse em conjuntos massivos de imagens, visando otimizar os resultados obtidos nas distâncias absolutas de Hamming e aprimorar a imperceptibilidade do sistema.

REFERÊNCIAS BIBLIOGRÁFICAS

- ABBAS, S. A. et al. Optimized video steganography using cuckoo search algorithm. In: *2015 IEEE Seventh International Conference on Intelligent Computing and Information Systems (ICICIS)*. [S.l.: s.n.], 2015. p. 572–577. Citado 2 vezes nas páginas 53 e 77.
- AKHTAR, A. Evolution of ant colony optimization algorithm: A brief literature review. *arXiv preprint arXiv:1908.08007*, 2019. Disponível em: <https://arxiv.org/abs/1908.08007>. Citado na página 26.
- BABU, A. G. S.; CHIRANJEEVI, B. T. Implementation of fractional order pid controller for an avr system using ga and aco optimization techniques. In: *IFAC-PapersOnLine*. [S.l.]: Elsevier, 2016. v. 49, n. 1, p. 456–461. Citado 2 vezes nas páginas 26 e 27.
- BANHARNSAKUN, A. Artificial bee colony approach for enhancing lsb based image steganography. *Multimedia Tools and Applications*, v. 77, 2018. ISSN 0030-4026. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0030402618306399>>. Citado 3 vezes nas páginas 15, 21 e 37.
- BOUGHACI, D.; KEMOUCHE, A.; LACHIBI, H. Stochastic local search combined with lsb technique for image steganography. In: *2016 13th Learning and Technology Conference (L&T)*. [S.l.: s.n.], 2016. p. 1–9. Citado 3 vezes nas páginas 15, 35 e 36.
- CHOUDHARY, K. Ant colony optimization-based models for agriculture price forecasting: Innovations, case studies, and future prospects. In: _____. [S.l.: s.n.], 2024. ISBN 978-1-83769-180-7. Citado na página 26.
- D., I. S.; CHAKRABORTY, U. K. Genetic algorithm: An approach on optimization. In: *Proceedings of the Fourth International Conference on Communication and Electronics Systems (ICCES 2019)*. [S.l.]: IEEE, 2019. p. 1–4. ISBN 978-1-7281-1261-9. Citado 2 vezes nas páginas 27 e 28.
- DASGUPTA, K.; MONDAL, J. K.; DUTTA, P. Optimized video steganography using genetic algorithm (ga). *Procedia Technology*, v. 10, p. 131–137, 2013. ISSN 2212-0173. First International Conference on Computational Intelligence: Modeling Techniques and Applications (CIMTA) 2013. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S2212017313004994>>. Citado na página 37.
- DESALE, S. et al. Heuristic and meta-heuristic algorithms and their relevance to the real world: A survey. *INTERNATIONAL JOURNAL OF COMPUTER ENGINEERING IN RESEARCH TRENDS*, v. 351, p. 2349–7084, 01 2015. Citado na página 24.
- DORIGO, M.; STÜTZLE, T. Ant colony optimization: Overview and recent advances. In: *Handbook of Metaheuristics*. [S.l.]: Springer, 2019. p. 311–351. Citado 2 vezes nas páginas 25 e 27.
- ELSHARE, S.; EL-EMAM, N. Modified multi-level steganography to enhance data security. *International Journal of Communication Networks and Information Security*, v. 10, p. 509–525, 12 2018. Citado na página 77.

Florida State University. *Digital Image Basics*. s.d. National High Magnetic Field Laboratory. Acesso em: 1 fev. 2026. Disponível em: <<https://hamamatsu.magnet.fsu.edu/articles/digitalimagebasics.html>>. Citado na página 20.

G, G. et al. Enhanced data encryption in iot using ecc cryptography and lsb steganography. In: *2021 International Conference on Design Innovations for 3Cs Compute Communicate Control (ICDI3C)*. [S.l.]: IEEE, 2021. Citado 3 vezes nas páginas 14, 35 e 36.

GAERTNER, D.; CLARK, K. On optimal parameters for ant colony optimization algorithms. In: . [S.l.: s.n.], 2005. v. 1, p. 83–89. Citado na página 65.

GARCIA, N. Digital steganography and its existence in cybercrime. *dimensions*, v. 640, p. 480, 2018. Citado na página 15.

GE, S.; GAO, Y.; WANG, R. Least significant bit steganography detection with machine learning techniques. In: *Proceedings of the 2007 International Workshop on Domain Driven Data Mining*. New York, NY, USA: Association for Computing Machinery, 2007. (DDDM '07), p. 24–32. ISBN 9781595938466. Disponível em: <<https://doi.org/10.1145/1288552.1288556>>. Citado 2 vezes nas páginas 34 e 35.

HASSAN, N.; HIJAZI, R. *Data Hiding Techniques in Windows OS: A practical approach to investigation and defense*. 1. ed. [S.l.]: Syngress, 2017. 1-22 p. Citado na página 18.

HUSSAIN, A. Z.; KHODHER, M. A. A. Securing medical images using chaotic map encryption and lsb steganography. *Revue d'Intelligence Artificielle*, v. 38, n. 1, p. 313–321, 2024. Citado 2 vezes nas páginas 15 e 35.

HUSSEIN, M. K.; ALHIJAJ, A. A. Protection of images by combination of vernam stream cipher, aes, and lsb steganography in a video clip. *Bulletin of Electrical Engineering and Informatics*, v. 12, n. 3, p. 1578–1585, 2023. Citado 4 vezes nas páginas 15, 19, 35 e 36.

IBANEZ, A. L.; DJAMAL, E. C.; NAJMURROKHMAN, A. Optimization of least significant bit steganography using genetic algorithm to improve data security. In: *2018 10th International Conference on Information Technology and Electrical Engineering (ICITEE)*. [S.l.]: IEEE, 2018. Citado 7 vezes nas páginas 14, 15, 22, 32, 33, 63 e 77.

JASPER, N. A.; PEREIRA, S. do L. Classificação de técnicas esteganográficas. In: *11º Congresso de Tecnologia da FATEC-SP*. São Paulo, Brasil: [s.n.], 2009. Citado na página 19.

JUNG, M. et al. A dynamic construction algorithm for the compact patricia trie using the hierarchical structure. *Information Processing Management*, v. 38, n. 2, p. 221–236, 2002. ISSN 0306-4573. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0306457301000310>>. Citado na página 44.

Júnior, D. C. N. *Grafos e problemas de caminhos*. Dissertação (Dissertação de Mestrado) — Universidade Federal de Viçosa, Florestal, Minas Gerais, Brasil, 2017. Citado na página 25.

KARAMPIDIS, K.; KAVALLIERATOUA, E.; PAPADOURAKIS, G. A review of image steganalysis techniques for digital forensics. *ELSEVIER*, 2018. Citado na página 23.

KHALIL, N.; SARHAN, A.; ALSHEWIMY, M. A. M. A secure image steganography based on lsb technique and 2d chaotic maps. *Computers and Electrical Engineering*, v. 119, n. 109566, 2024. Citado 4 vezes nas páginas 15, 21, 33 e 34.

KHUDHER, I. Lsb steganography strengthen footprint biometric template. *Eastern-European Journal of Enterprise Technologies*, v. 1, p. 58–65, 02 2021. Citado 2 vezes nas páginas 35 e 36.

KING, D. J.; LAUNCHBURY, J. Structuring depth-first search algorithms in haskell. In: *Proceedings of the 22nd ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*. New York, NY, USA: Association for Computing Machinery, 1995. (POPL '95), p. 344–354. ISBN 0897916921. Disponível em: <<https://doi.org/10.1145/199448.199530>>. Citado na página 24.

KUMBHAKAR, D.; SANYAL, K.; KARFORMA, S. An optimal and efficient data security technique through crypto-stegano for e-commerce. *Multimedia Tools and Applications*, v. 82, p. 21005–21018, 2023. Citado 3 vezes nas páginas 18, 33 e 34.

LI, C. et al. Facial feature based secure information transmission. In: *Proceedings of the International Conference on Big Data and Internet of Thing*. New York, NY, USA: Association for Computing Machinery, 2017. (BDIOT '17), p. 141–144. ISBN 9781450354301. Disponível em: <<https://doi.org/10.1145/3175684.3175705>>. Citado 3 vezes nas páginas 14, 63 e 77.

LI, T. et al. Genetic algorithm for building optimization - state-of-the-art survey. *School of Automation, Chongqing University*, 2014. Review Paper co-authored with University of Miami. Citado 2 vezes nas páginas 27 e 28.

LINDEN, R. *Algoritmos Genéticos: Teoria e Implementação*. 2. ed. Rio de Janeiro: Ciência Moderna, 2008. ISBN 9788573937389. Citado 2 vezes nas páginas 65 e 68.

LUO, X.; LIU, F.; LIU, B. Detecting lsb steganography based on dynamic masks. In: *Proceedings of the 5th International Conference on Intelligent Systems Design and Applications (ISDA'05)*. [S.l.]: IEEE Computer Society, 2005. p. 251–255. Citado na página 33.

MARE, S. F.; VLADUTIU, M.; PRODAN, L. Decreasing change impact using smart lsb pixel mapping and data rearrangement. In: *2011 11th IEEE International Conference on Computer and Information Technology*. [S.l.]: IEEE, 2011. Citado 2 vezes nas páginas 35 e 36.

NASREEN, S. An efficient seac system for different data transmission over internet. In: *2016 International Conference on Automatic Control and Dynamic Optimization Techniques (ICACDOT)*. [S.l.]: IEEE, 2016. Citado na página 35.

OLEWI, H. I. et al. Enhanced security through integrated morse code encryption and lsb steganography in digital communications. *Traitement du Signal*, v. 41, n. 1, p. 519–524, 2024. Citado 2 vezes nas páginas 20 e 34.

PELOSI, M. et al. Positive identification of lsb image steganography using cover image comparisons. *Annual ADFSL Conference on Digital Forensics, Security and Law*, 2018. Citado na página 18.

RACHMAWATI, D.; EFENDI, S.; SITUMORANG, A. S. Comparative analysis of depth-first search algorithm and greedy algorithm at nearest atm search in padang sidempuan city. *Journal of Theoretical and Applied Information Technology*, v. 98, n. 17, September 2020. 15th September 2020. Citado na página 25.

RAFRASTARA, F. A. et al. Image steganography using inverted lsb based on 2nd, 3rd and 4th lsb pattern. In: *2019 International Conference on Information and Communications Technology (ICOIAC)*. [S.l.]: IEEE, 2019. Citado 2 vezes nas páginas 35 e 36.

SAJASI, S.; EFTEKHARI-MOGHADAM, A.-M. A high quality image hiding scheme based upon noise visibility function and an optimal chaotic based encryption method. In: *2013 3rd Joint Conference of AI Robotics and 5th RoboCup Iran Open International Symposium*. [S.l.: s.n.], 2013. p. 1–7. Citado 3 vezes nas páginas 35, 36 e 63.

SARI, C. A. et al. An enhancement of des, aes based on imperceptibility along with lsb. In: *2022 International Seminar on Application for Technology of Information and Communication (iSemantic)*. [S.l.]: IEEE, 2022. Citado 4 vezes nas páginas 35, 36, 63 e 77.

SGURSKY, L. F. F. *ANÁLISE E IMPLEMENTAÇÃO DE TÉCNICAS DE ESTEGANOGRAFIA*. 119 p. Monografia — UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ, CORNÉLIO PROCÓPIO, PR, 2015. Citado na página 23.

SHAOZHANG, N. et al. Detecting lsb steganography based on noise function. *Chinese Journal of Electronics*, v. 18, n. 2, 2009. Citado na página 33.

SINGH, B. P. *The project is based on emerging field Image Processing, In this project A Graphical User Interface has been designed using the software Labwindows. which can process both type of Real time Image Processing and Also many formats of images like .JPG, .DAT, .BMP etc.* 2015. Citado na página 20.

SINGHAL, A.; BEDI, P. Steganography using cuckoo optimized wavelet coefficients. In: *Proceedings of the Third International Symposium on Women in Computing and Informatics*. New York, NY, USA: Association for Computing Machinery, 2015. (WCI '15), p. 365–370. ISBN 9781450333610. Disponível em: <<https://doi.org/10.1145/2791405.2791500>>. Citado 3 vezes nas páginas 14, 35 e 36.

WALIA, G. S. et al. Robust stego-key directed lsb substitution scheme based upon cuckoo search and chaotic map. *Optik*, v. 170, p. 106–124, 2018. ISSN 0030-4026. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0030402618306399>>. Citado 3 vezes nas páginas 15, 21 e 37.

WAZIRALI, R. et al. An optimized steganography hiding capacity and imperceptibly using genetic algorithms. *IEEE Access*, v. 7, p. 133496–133508, 2019. Citado na página 37.

ZIN, W. W.; SOE, T. N. Implementation and analysis of three steganographic approaches. In: *2011 3rd International Conference on Computer Research and Development*. [S.l.: s.n.], 2011. v. 2, p. 456–460. Citado 2 vezes nas páginas 35 e 36.