LVI SBPO SIMPÓSIO BRASILEIRO DE PESQUISA OPERACIONAL
DE 04 ATÉ 07 NOVEMBRO 2024
FORTALEZA | CEARÁ

otimização do modelo de Esteganografia LSB em imagens utilizando Algoritmo de Colônia de Formigas

Richard Sidney de Souza e Silva
Instituto Federal da Paraíba (IFPB/JP)
João Pessoa, Brasil
richard.silva@academico.ifpb.edu.br

Carlos Thassius Ferreira Freire
carlos.thassius@academico.ifpb.edu.br

Emanuel Eduardo Oliveira Dionisio
emanuel.eduardo@academico.ifpb.edu.br

Thiago Gouveia
thiago.gouveia@ifpb.edu.br

RESUMO

A cibersegurança é um tópico recorrente em trabalhos sobre tecnologia e, dentro desse espectro, destaca-se o transporte seguro de arquivos digitais. Nesse sentido, a esteganografia é uma ferramenta fundamental e amplamente explorada no compartilhamento de informação por meio de imagens. Este trabalho traz uma alternativa de implementação da substituição LSB (*Least Significant Bit*) utilizando estratégias de inserção de bits de forma não sequencial. Esse modelo é implementado como um problema de otimização, buscando a estratégia mais adequada, e para isso foram utilizados os algoritmos DFS (*Depth-First Searc*), Guloso e ACO. Nos resultados, foi demonstrado que o modelo sugerido, em conjunto com o algoritmo de otimização ACO (*Ant Colony Optimization*), apresentou um desempenho promissor ante métodos de esteganálise e melhorou a métrica de diferença de intensidade de pixeis em relação ao LSB convencional.

PALAVRAS CHAVE. Esteganografia LSB, Otimização de colônia de formigas, Meta-heurísticas.

Tópicos: Meta-heurísticas e Cibersegurança.

ABSTRACT

Cybersecurity is a recurring topic in technology-related works, and within this spectrum, secure digital file transport stands out. In this regard, steganography is a fundamental and widely explored tool for sharing information through images. This paper presents an alternative implementation of LSB (Least Significant Bit) substitution using non-sequential bit insertion strategies. This model is implemented as an optimization problem, seeking the most suitable strategy, and for this purpose, Depth-First Search (DFS), Greedy, and Ant Colony Optimization (ACO) algorithms were used. The results demonstrated that the suggested model, together with the ACO optimization algorithm, showed promising performance compared to steganalysis methods and improved the pixel intensity difference metric compared to conventional LSB.

KEYWORDS. LSB Steganography, Ant Colony Optmization, Metaheuristics.

Paper topics: Metaheuristics and Cybersecurity.

DOI: 10.59254/sbpo-2024-193420
https://proceedings.science/p/193420?lang=pt-br

1. Introdução

O uso massivo de arquivos digitais para compartilhamentos de informações traz à tona a grande necessidade de evolução nos métodos de segurança digital para garantia da integridade desses dados, que muitas vezes representam informações sensíveis. Dessa forma, pode-se apontar a Segurança da Informação como um dos principais fatores que pauta o uso atual da tecnologia.

A Esteganografia pode ser definida como uma ferramenta da comunicação segura, cujo objetivo é enviar informações ocultas de forma confiável e secreta. Embora seu conceito seja antigo, anterior à revolução industrial, a Esteganografia como ferramenta digital tem sido explorada e diversas técnicas vêm sendo desenvolvidas com objetivo de ocultar informações em arquivos de imagens, vídeos e áudios [Saini et al., 2012]. É considerada uma subdisciplina no domínio da Segurança da Comunicação de dados [Djebbar et al., 2012].

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

Há uma variedade de técnicas diferentes disponíveis na literatura para realizar esteganografia em imagens. Elas podem ser separadas entre técnicas que manipulam o domínio espacial e o domínio da frequência. As técnicas de esteganografia de domínio espacial são baseadas na alteração das intensidades dos pixeis da imagem hospedeira para esconder a mensagem secreta. A substituição de LSB (*Least Significant Bit*) está entre os métodos de esteganografia de domínio espacial mais populares. Nesse esquema, os dados são ocultados pela substituição do LSB da imagem de cobertura pela mensagem secreta Walia et al. [2018], como ilustrado na Figura 1.

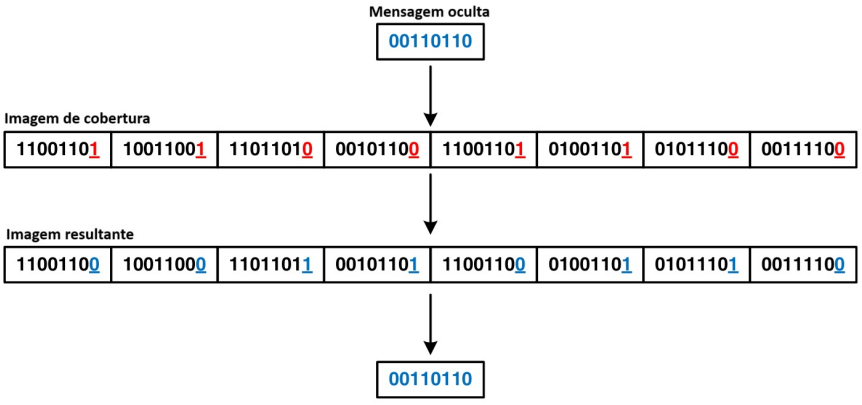

Figura 1: Método de substituição LSB [Banharnsakun, 2018]

Diversos trabalhos se debruçam no estudo de novas abordagens da técnica de substituição LSB na tentativa de minimizar os problemas no uso desse método. A substituição LSB possui diversas limitações, como o limite de dados inseridos na imagem e facilidade de detecção em sistemas que modifiquem aspectos da imagem, como cor e saturação, além da fragilidade do uso em imagens de cobertura que contêm mais a cor branca [Boughaci et al., 2016].

Dessa maneira, neste trabalho foi desenvolvido um modelo de esteganografia utilizando a

DOI: 10.59254/sbpo-2024-193420
https://proceedings.science/p/193420?lang=pt-br

substituição LSB. O modelo proposto utiliza abordagem de busca pela imagem de cobertura ideal, uma imagem que recepcione os dados da mensagem de forma que altere a menor quantidade de bits possível. Para isso, foram utilizados algoritmos de otimização que auxiliaram na busca pela imagem de cobertura ideal. O restante deste trabalho está organizado como segue: a Seção 2 discute alguns trabalhos relacionados ao tema; a Seção 3 apresenta os métodos propostos; os experimentos computacionais são discutidos na Seção 4; enquanto as considerações finais e propostas de trabalhos futuros são trazidos na Seção 5.

2. Trabalhos Relacionados

Essa Seção é destinada à exposição de trabalhos relacionados às implementações de técnicas de esteganografia por substituição LSB como problemas de otimização. Tal exposição tem por objetivo mapear o estado da arte dessa área de estudo e demonstrar a contribuição que o método aqui proposto promove nesse âmbito.

No trabalho realizado por Wazirali et al. [2019], foi apontado que a quantidade de dados que são escondidos numa imagem via esteganografia é diretamente proporcional à deformação que esta sofre, o que resulta em um declínio na qualidade. Com efeito, o ato de inserir uma alta quantidade de dados em uma imagem, ao passo que se conserva sua qualidade, pode ser denotado como um problema de otimização. Nesse contexto, o artigo propõe uma busca pela solução ótima do problema supracitado via algoritmos genéticos. O esquema apresentado utiliza operações otimizadas para maximizar a correspondência entre os bits menos significativos. Comparado com os esquemas tradicionais LSB, o apresentado pelos autores demonstrou menores distorções nas imagens esteganografadas, além de ter aprimorado a indetectabilidade.

Nessa linha, Banharnsakun [2018] propõe a utilização de uma forma de aprimoramento da técnica de esteganografia LSB por meio de métodos baseados no algoritmo da colônia de abelhas (ABC). O procedimento exposto usa a ABC para otimizar a atribuição de blocos para a inserção de uma imagem secreta em uma imagem hospedeira. Essa atribuição é considerada como um problema de otimização combinatória. Conforme o fato de que a ABC ser direcionada para a resolução de problemas de otimização numéricos, foi introduzida ao sistema uma lista de atribuição de blocos para representar as soluções no algoritmo ABC. Os resultados demonstraram um aprimoramento tanto na qualidade da imagem esteganografada quanto na sua tolerância a distorções, um ponto fraco nas técnicas de esteganografia LSB. A necessidade de conhecer a permutação usada no processo para resgatar a imagem oculta tornou o método seguro e eficaz.

Continuamente, Araujo e Gramacho [2023] apresentam o uso da matriz de substituição como uma forma de mitigar as restrições que a esteganografia LSB apresenta, devido à sua eficiência, reversibilidade e simplicidade de mapear tais substituições. Dessa forma, é exibido que o ato de encontrar uma configuração ótima para essa matriz é um problema combinatório e deterministicamente inviável. Como efeito, aplica conceitos de Algoritmos Genéticos com Chaves Aleatórias para encontrar soluções em tempos viáveis. Como resultado, o esquema proposto demonstrou melhorias nos resultados conforme um limite determinado pelos parâmetros selecionados.

Nesse meio, Boughaci et al. [2016] propuseram uma busca local estocástica como uma forma de suprimir as limitações expostas pela esteganografia LSB. O trabalho implementou e avaliou três métodos de esteganografia (LSB, LSB com busca local e LSB com busca local estocástica) em uma série de imagens JPEG. As representações binárias das imagens foram decompostas, juntamente com as mensagens, em quatro blocos de bits. Esses blocos foram selecionados da melhor forma possível segundo cada implementação utilizada. Como resultado, os modelos apresentados utilizando busca local, embora mais lentos que o LSB padrão, demonstraram uma manutenção na qualidade e no tamanho da imagem esteganografada, o que denota uma maior indetectabilidade.

O trabalho desenvolvido por Siar et al. [2018] utilizou um algoritmo de otimização de

DOI: 10.59254/sbpo-2024-193420
https://proceedings.science/p/193420?lang=pt-br

colônia de formigas (ACO) como estratégia de implementação de esteganografia em imagens. Neste trabalho, é usada uma função de energia reversa que pretende conservar a qualidade da imagem com os dados embutidos e observar a imperceptibilidade necessária para um esquema de esteganografia. O modelo utilizou ACO na busca de partes da imagem menos significativas, de forma que isso reduziu a distorção nas imagens. Nos resultados, o modelo utilizando a otimização ACO mostrou-se superior a outros métodos, como o LSB simples e o DCT-M3.

Walia et al. [2018] trabalharam em um método visando diminuir o comprometimento da qualidade visual da imagem esteganografada usando a técnica LSB. O artigo foi modelado como um problema de busca e otimização. Foi introduzido um novo esquema de substituição de LSB dirigido por uma chave, que não apenas ofereceu alta capacidade de incorporação, mas também mostrou-se resistente à distorção. A chave estego ótima é determinada através da otimização por Busca Cuckoo (CS). Os resultados desse trabalho foram promissores e foram avaliados usando as métricas de Relação Sinal-Ruído de Pico (PSNR) e pela Matriz de Similaridade Estrutural (SSIM).

Por fim, Dasgupta et al. [2013] propôs a utilização de Algoritmos Genéticos para aprimorar a técnica base de LSB. Para efeitos comparativos entre o *frame* original e o esteganografado, são utilizados métodos anti-esteganálise. Como resultado, foi evidenciada uma melhora nos índices de relação Sinal-Ruído de pico (PSNR) e de fidelidade de imagem (IF) na esteganografia otimizada em relação à LSB padrão.

3. Métodos Propostos

Nesta Seção são apresentados os métodos propostos para o desenvolvimento de um novo modelo de esteganografia LSB, assim como os algoritmos utilizados na busca otimizada pela imagem de cobertura ideal. Também foi implementada uma abordagem baseada na meta-heurística de Colônia de Formigas (ACO, do inglês *Ant Colony Optimization*). Em um primeiro momento, apresentamos estruturas e algoritmos utilizados para construção de uma árvore de busca utilizada pelo modelo de esteganografia LSB. Em seguida, apresentamos os algoritmos de otimização do modelo.

3.1. Novo 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 de como 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 2. Adicionalmente, é proposto o uso de um banco de imagens visando selecionar uma imagem que minimize o número de bits alterados.

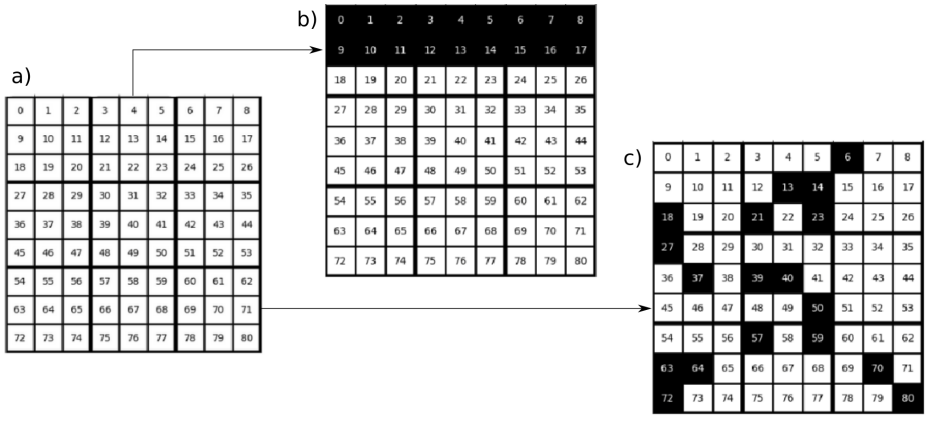

Figura 2: Matrizes LSB

DOI: 10.59254/sbpo-2024-193420
https://proceedings.science/p/193420?lang=pt-br

O primeiro passo para implementação do modelo foi a montagem do *benchmark*. Para as imagens é utilizado um banco de dados disponível no *Kaggle*, *Dogs vs Cats*[1], que possui um total de 25 mil imagens, com dimensões distintas. E para as mensagens é utilizado um banco de mensagens de texto extraídas da rede social X, antigo Twitter, o *Twitter Threads*[2].

A implementação do modelo proposto requer alguns passos de pré-processamento. O primeiro é o mapeamento das dimensões de todas as imagens do banco de dados, na intenção de identificar a capacidade máxima de incorporação, tomando como referência a imagem com a menor capacidade de inserção de dados, que, segundo o banco utilizado, foi uma imagem com dimensões 50x38. Assim, definiu-se que o tamanho máximo das mensagens utilizadas seria de 1024 bits (128 caracteres).

Com a quantidade máxima de informação definida, é realizado o segundo passo do pré-processamento: a geração das estratégias de inserção de bits. Uma estratégia é um *array* de tamanho m, onde $m = 1024$, cujos valores são posições aleatórias que vão de 0 a n, onde n é o tamanho do *array* LSB da imagem com as menores dimensões do banco, no caso $n = 50 \times 38 \times 3 = 5700$. O número de estratégias criadas foi definido como $z = 1000$.

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 estratégia utilizada, como apresentado na Figura 3. 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.

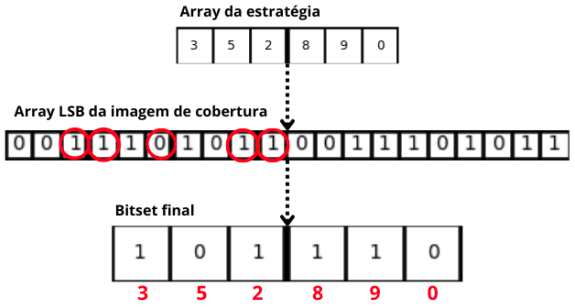

Figura 3: Operação de criação do bitset

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 5 milhões de possibilidades, a busca pelo melhor resultado torna-se um problema de otimização. A Figura 4 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, que são aplicados a cada imagem, gerando 25 milhões de *bitsets* (d), que são ordenados e salvos em um arquivo utilizando codificação de base64 (e).

3.2. Árvore de busca

A próxima etapa proposta neste trabalho é realizar a decomposição do arquivo de *bitsets* construído em células, a partir da qual se pode obter uma estrutura em forma de grafo, que denomi-

[1] https://www.kaggle.com/competitions/dogs-vs-cats
[2] https://www.kaggle.com/datasets/danielgrijalvas/twitter-threads

DOI: 10.59254/sbpo-2024-193420
https://proceedings.science/p/193420?lang=pt-br

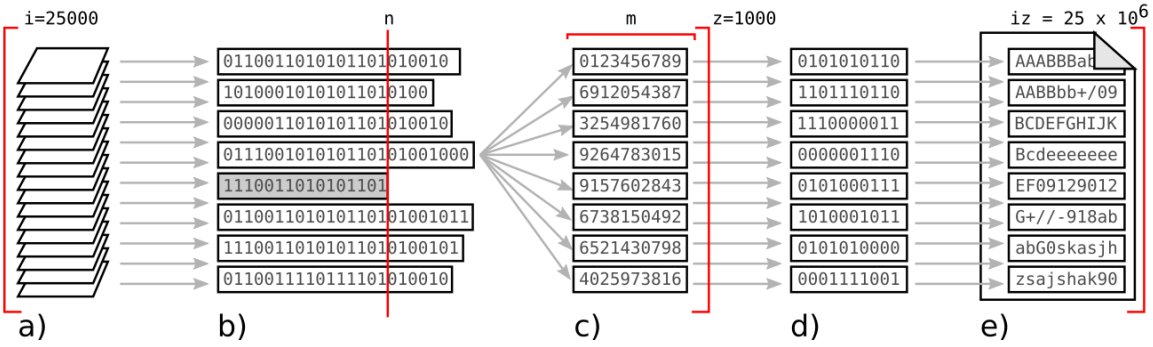

Figura 4: Montagem do array de bitsets

namos árvore de busca. A árvore de busca utilizada é uma estrutura fundamental para aplicação dos algoritmos de busca otimizada que propomos: DFS, Guloso e ACO, sendo construída utilizando o arquivo de bitsets ordenado e codificado em base64. A Figura 5 denota uma representação simplificada da árvore de busca. Os filhos são resultados da diferença entre *bitsets*, enquanto os pais representam a parte comum entre eles. Pode-se observar, à esquerda, 10 conjuntos de *bitsets*, um em cada linha, ordenados como no arquivo original. À medida que avançamos para a direita nos conjuntos, a árvore é ramificada.

A estrutura da árvore da Figura 5 possui apenas um nó no primeiro, uma vez que todos os *bitsets* começam com o mesmo subconjunto de caracteres A. No segundo nível, se tem a primeira diversificação, onde encontram-se três nós, o primeiro com o subconjunto $AAAA$, o segundo $BAAABBBBBCCC$ e o terceiro C. A partir do terceiro nível, apenas o primeiro e o terceiro nó do segundo nível seguem se ramificando e o segundo tornou-se uma folha, visto que os três filhos seriam iguais. O primeiro nó do segundo nível se ramifica em duas folhas e o terceiro nó do segundo nível se ramifica até formar 4 folhas. Esse comportamento também é representado graficamente na Figura 5, onde cada nível foi representado por uma cor diferente.

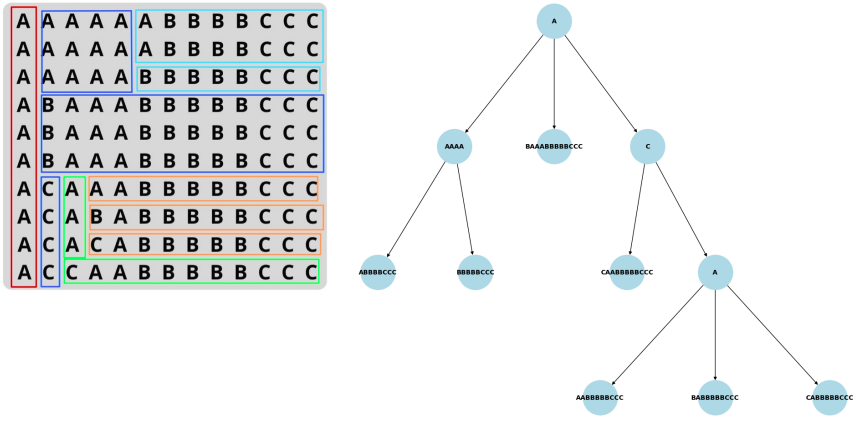

Figura 5: representação como árvore de busca

Assim, fica definida a estrutura da árvore utilizada neste trabalho, onde as estratégias codificadas em base64, ordenadas no arquivo de origem, vão se ramificando à medida que subconjuntos de caracteres vão se diferenciando entre elas. Então, definiu-se que o resultado ideal da busca na árvore é o caminho percorrido com a menor distância de Hamming, onde a distância é calculada

DOI: 10.59254/sbpo-2024-193420
https://proceedings.science/p/193420?lang=pt-br

entre a representação binária do caminho percorrido e da mensagem binarizada.

3.3. Busca Otimizada

Este trabalho propõe quatro algoritmos de busca, cujo objetivo é encontrar dentro do 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. O primeiro é baseado em DFS (do inglês, *Depth-First Search*, ou busca em profundidade), 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

O algoritmo de Colônia de Formigas ACO (do inglês, *Ant Colony Optimization*) é uma meta-heurística inspirada pelo comportamento de forrageamento das Colônias de formigas reais. As formigas depositam feromônio no solo para marcar a rota e identificar os caminhos do ninho até a comida [Chandra Mohan e Baskaran, 2012]. Esta seção descreve a heurística baseada em ACO que propomos para encontrar o melhor par imagem-estratégica utilizando a árvore de busca descrita na Subseção 3.2.

Na implementação proposta, cada formiga deve percorrer um caminho da raiz da árvore de busca até uma de suas folhas, depositando feromônio nos seus vértices, o que influencia probabilisticamente a escolha de caminhos das formigas das próximas gerações. O Algoritmo 1 detalha a implementação proposta, enquanto a Tabela 1 descreve os parâmetros utilizados.

Algoritmo 1: Método baseado na Meta-heurística ACO

```
1  Procedure ACO(r, M, G, Q):
2      c* ← ∞;
3      F ← initFer();
4      for g ← 1 to G do
5          for f ← 1 to Q do
6              node ← r;
7              c_f ← cost(node, M);
8              while node is not a leaf do
9                  node ← EscolhaFilho(node, F);
10                 c_f ← c_f + cost(node, M);
11             c* ← best(c_f, c*);
12         F ← updateFer(c_1, c_2, ..., c_f);
13     return c*;
```

Em detalhes, as linhas 1 e 2 realizam as inicializações necessárias, sendo F a quantidade de feromônio em cada nó, inicialmente 0. Os laços das linhas 4 e 5 fazem para cada geração até G,

DOI: 10.59254/sbpo-2024-193420
https://proceedings.science/p/193420?lang=pt-br

sejam liberadas Q formigas, de modo que cada uma construa uma solução. O laço das linhas 6 a 10 descreve o processo de construir um caminho da raiz até uma folha da árvore de busca realizado por uma formiga, enquanto o custo deste caminho é atualizado nas linhas 7 e 10. A função $cost$ retorna a Distância de Hamming entre parte da mensagem M e a *string* armazenada no nó corrente. A linha 11 mantém a melhor solução encontrada e a linha 12 realiza a atualização do valor do feromônio de cada nó para cada solução encontrada nesta geração. Adicionalmente, a escolha do filho a ser visitado (linha 9) é dada aleatoriamente conforme a seguinte probabilidade:

$$p_{ij} = \frac{[\tau_i]^\alpha \cdot [\eta_i]^\beta}{\sum_{k \in \mathrm{N}(i)} [\tau_k]^\alpha \cdot [\eta_k]^\beta}, \tag{1}$$

sendo τ_i a quantidade de feromônio no nó i; η_i é o custo de visitar o nó i; α é o parâmetro que controla a influência do feromônio; β é o parâmetro que controla a influência do custo do nó; e $\mathrm{N}(i)$ é o conjunto de filhos do nó i. Por fim, dada uma solução com custo c_f, a quantidade de feromônio adicionado em cada nó desta solução (linha 12) é dada pela fórmula $10^3 c_f^{-1}$.

Tabela 1: Parâmetros utilizados para a ACO.

Parâmetro	**Valor**
Número de formigas Q	1900
Número máximo de iterações G	100
Fator de evaporação ρ	-
Fator heurístico α	0,4
Fator heurístico β	0

4. Experimentos Computacionais

Os métodos propostos foram implementados utilizando duas linguagens de programação, Python e C++. O algoritmo de construção do arquivo com os *bitsets* de estratégias e com as mensagens foi desenvolvido em Python, assim como os algoritmos de substituição LSB. Os algoritmos de busca e otimização foram implementados na linguagem C++, utilizando o compilador g++ 8.3.0 - 6 (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 (64bits), 1 TB de SSD NMVE, e sistema operacional Ubuntu 23.10.

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 linha base de comparação. Após a execução da Força Bruta, foi encontrada uma média da distância de Hamming de 273,1, enquanto no método de substituição comum a média encontrada para as 100 mensagens foi de 333,24.

Em sequência, foram realizados os testes de desempenho do modelo de esteganografia LSB proposto. Para o teste do modelo de esteganografia, foram realizadas duas substituições LSB, uma no modelo tradicional de implementação e outra no modelo proposto. Para isso, a imagem selecionada como a melhor imagem de cobertura pelo método de *Força Bruta* foi utilizada como imagem de cobertura da mensagem 1. A implementação comum da substituição LSB substitui de maneira sequencial os bits menos significativos dos pixeis da imagem de cobertura. Dada uma imagem representada pela matriz $I = MXN$, onde M é o número de linhas de N o número de

DOI: 10.59254/sbpo-2024-193420
https://proceedings.science/p/193420?lang=pt-br

colunas, cada elemento $P(i, j)$ representa o pixel na linha i e coluna j. Nesta implementação, a função submete todo valor de $P(i, j)$ a substituição LSB até que toda a mensagem seja inserida na imagem de cobertura. A implementação do modelo proposto varreu a imagem seguindo a mesma lógica, porém o valor de $P(i, j)$ só era submetido a substituição LSB caso sua posição estivesse contida no *array* da estratégia escolhida pelo *Força Bruta* como a melhor, forçando os bits da mensagem a ficarem espalhados na imagem, 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 pixeis das imagens. Posteriormente, as imagens passaram por outro processamento que alterou a escala de cores. Ambos os métodos usados têm como objetivo expor as anomalias geradas pela substituição LSB. As Figuras 6 e 7 apresentam a imagem utilizada no teste da mensagem 1. Observa-se que foi feito um recorte da porção superior da imagem, esse recorte foi submetido aos métodos de Ataque Aural. A Equação 2 descreve a função de saturação utilizada, que deve ser executada para todo $P(i, j) \in I$.

$$f(x) = \begin{cases} 0 & \text{se } x = 0 \text{ ou } x \text{ for par;} \\ 255 & \text{se } x \text{ for ímpar;} \end{cases} \tag{2}$$

Nas Figuras 6 e 7, os recortes (b) passaram pelo processo de saturação. É possível notar, no recorte (b) da Figura 6, que na parte superior há uma linha de pixeis agrupada com padrão de coloração diferente do resto do recorte. Essa anomalia fica ainda mais clara quando observa o recorte (c) dessa Figura. Essa anomalia indica que os bits da mensagem oculta foram depositados ali. Contudo, quando se observam os recortes (b) e (c) da Figura 7, nota-se que as anomalias geradas pela substituição encontram-se atenuadas devido ao processo de espalhamento aleatório a que os bits foram submetidos. Isso ocorre porque essas anomalias se confundem com o próprio padrão da imagem de cobertura.

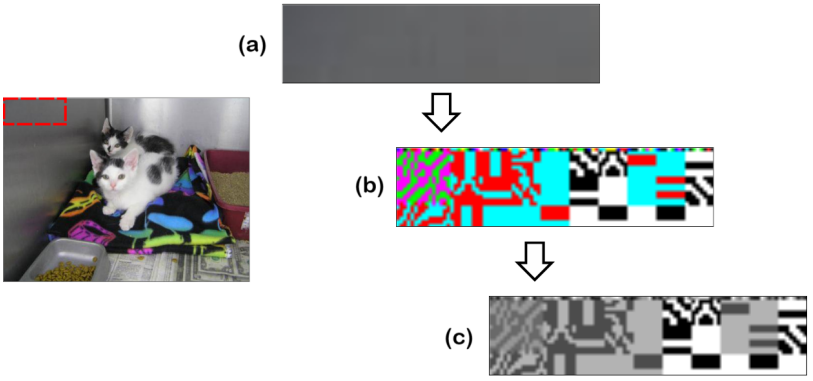

Figura 6: Imagem esteganográfica utilizando LSB comum

Uma análise da intensidade da diferença entre os pixeis da imagem de cobertura e as imagens fruto dos modelos de esteganografia foi realizada. Assim como Abbas et al. [2015], a diferença de pixeis foi plotada contra o índice de pixeis da imagem de cobertura. Os gráficos da análise encontram-se na Figura 8, neles é possível notar que a intensidade das diferenças diminui no modelo LSB proposto, enquanto a presença da diferença torna-se mais espaçada. Isso reforça o que foi observado nos testes de Ataque Aural, onde no modelo LSB proposto, os bits inseridos

DOI: 10.59254/sbpo-2024-193420
https://proceedings.science/p/193420?lang=pt-br

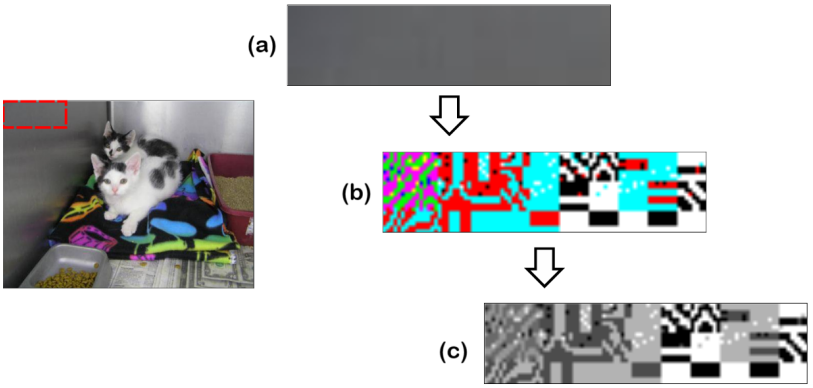

Figura 7: Imagem esteganográfica utilizando o modelo desenvolvido

na imagem de cobertura estão mais dispersos e o uso de uma estratégia ideal para a mensagem inserida reduz consideravelmente a diferença na intensidade entre os pixeis da imagem cobertura e da imagem resultante da esteganografia LSB proposta.

Com os testes do modelo LSB mostrando 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. Foram executados, nessa etapa, o algoritmo Guloso e a DFS. Na Tabela 2, encontram-se os resultados finais das médias das execuções de cada algoritmo. Os valores retirados para análise foram os da menor distância de Hamming média alcançada pelo algoritmo e a média de tempo de execução por busca em segundos.

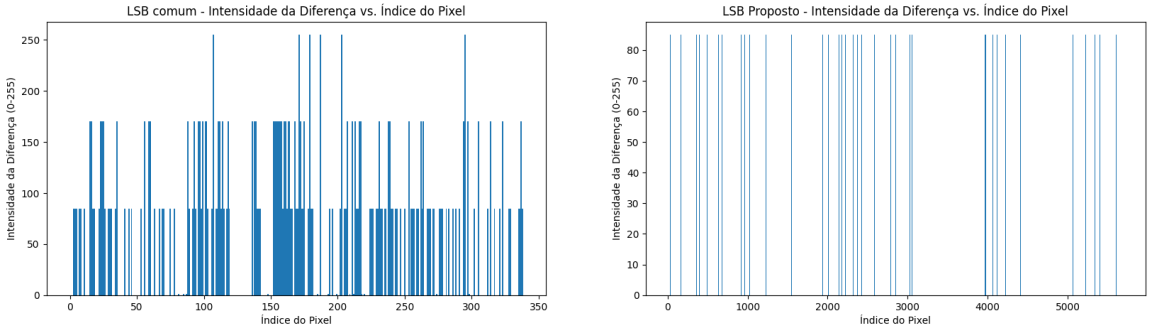

Figura 8: Gráficos da intensidade da diferença de pixeis

Tabela 2: Resultados dos Algoritmos de Busca

	Força Bruta	**DFS**	**Guloso**	**ACO**
Menor Distância de Hamming:	271,3	271,3	354,5	282,63
Tempo de Execução:	79,55 s	82,89 s	0,35 s	1,26 s

O último experimento computacional realizado foi a implementação da ACO, onde os valores adotados para os parâmetros que definem a probabilidade p_{ij} de uma formiga escolher uma aresta foram definidos empiricamente, observando o resultado e comparando com os resultados da DFS. Para simplificar a implementação, o fator de evaporação foi desconsiderado e o parâmetro β definido como 0. Foram feitas 15 execuções, variando o parâmetro α no intervalo entre $0,1$ e

DOI: 10.59254/sbpo-2024-193420
https://proceedings.science/p/193420?lang=pt-br

Tabela 3: Resultados do ACO por Alfa

	Fator heurístico α	**Menor Distância de Hamming**	**Tempo**
Melhor resultado	0,4	282,63	1,25 s
Pior resultado	1,3	283,72	1,26 s

Tabela 4: Resultados do ACO em comparação com os demais algoritimos.

	Força Bruta	**DFS**	**Guloso**
Aumento % na distância de Hamming	4,18%	4,18%	-20,27%
Ganho de Velocidade	63,14	65,78	0,28

$1,5$. O melhor e o pior resultados podem ser vistos na Tabela 3 e o α escolhido foi o $0,4$. São apresentados dados de comparação direta entre o ACO e os demais, destacando-se o ganho na velocidade de execução em relação aos métodos de referência, DFS o Força Bruta, com uma perda de desempenho um pouco maior que 4% em relação a eles na Tabela 4.

5. Considerações finais

Neste trabalho, é proposto um novo modelo de esteganografia por substituição LSB. O modelo foi baseado na implementação de chave de esteganografia, na qual os dados são inseridos na imagem de maneira aleatória e não sequencial. Em sua implementação, foi construído um arquivo de 25 milhões de possibilidades de substituição LSB. também foi proposto um método para otimizar a busca pela estratégia de cobertura ideal para uma mensagem, consistindo na implementação de uma ACO em um grafo de árvore de busca estruturado a partir do arquivo de *bitsets*.

Em relação aos resultados, o modelo de esteganografia LSB proposto mostrou-se promissor devido a inserção não sequencial dos bits da mensagem na imagem de cobertura. Elshare e EL-Emam [2018] mostra em seu trabalho que a inserção não linear dos bits da imagem de cobertura melhora a imperceptibilidade do modelo LSB. O modelo melhorou o desempenho frente as técnicas de esteganálise utilizadas, diminuindo a percepção da presença da esteganografia em relação ao LSB comum. A boa avaliação do modelo foi reforçada pelo teste de diferença de pixeis entre a imagem de cobertura e a imagem fruto do modelo, métrica utilizada por Abbas et al. [2015]. Além disso, o modelo de chave adiciona um nível de segurança extra no modelo proposto.

O uso da técnica de otimização ACO enriquece o modelo. Embora ele nem sempre encontre a estratégia de inserção com menor distância de Hamming da mensagem, em média ele encontra resultados com uma diferença de apenas 4,1% no melhor resultado, com um tempo de processamento mais de 63 vezes menor que o da DFS e da Força Bruta. E, quando comparado à substituição LSB comum, houve uma redução de 15,05% na distância de Hamming média. Esses resultados tornam-se ainda mais expressivos quando se nota que nos testes desse modelo o tamanho das mensagens foi limitado, assim como o espaço de manipulação das imagens. Um aumento no tamanho das mensagens e no espaço de manipulação elevaria o custo de processamento dos algoritmos de busca, destacando o tempo de execução 63 vezes melhor do ACO com custo aproximado de 4%.

Como trabalhos futuros, se propõe refinar as estratégias de substituição LSB, diminuindo o componente de aleatoriedade delas, direcionando mais a inserção dos bits das mensagens em pontos definidos na imagem. Do mesmo modo, avaliar a capacidade de incorporação de informação, ampliando o tamanho das mensagens e o espaço de manipulação nas imagens de cobertura. Isso levará mais tempo e poder de processamento, o que foi um fator limitante neste trabalho. Além disso, há que se explorar outras meta-heurísticas utilizadas nesse tipo de otimização.

DOI: 10.59254/sbpo-2024-193420
https://proceedings.science/p/193420?lang=pt-br

Agradecimentos

Os autores agradecem à FAPESQ-PB pela contribuição com este artigo por meio dos Editais 10/2021-FAPESQ/PB-MCTIC/CNPq e 11/2023-FAPESQ/SECTIES.

Referências

Abbas, S. A., El Arif, T. I. B., Ghaleb, F. F. M., e Khamis, S. M. (2015). Optimized video steganography using cuckoo search algorithm. In *2015 IEEE Seventh International Conference on Intelligent Computing and Information Systems (ICICIS)*, p. 572–577.

Araujo, G. e Gramacho, W. (2023). Esteganografia usando algoritmo genético de chaves aleatórias viciadas. *Academic Journal on Computing, Engineering and Applied Mathematics*, 4:69–72.

Banharnsakun, A. (2018). Artificial bee colony approach for enhancing lsb based image steganography. *Multimedia Tools and Applications*, 77. ISSN 0030-4026. URL https://www.sciencedirect.com/science/article/pii/S0030402618306399.

Boughaci, D., Kemouche, A., e Lachibi, H. (2016). Stochastic local search combined with lsb technique for image steganography. In *2016 13th Learning and Technology Conference (L&T)*, p. 1–9.

Chandra Mohan, B. e Baskaran, R. (2012). A survey: Ant colony optimization based recent research and implementation on several engineering domain. *Expert Systems with Applications*, 39(4): 4618–4627. ISSN 0957-4174. URL https://www.sciencedirect.com/science/article/pii/S0957417411013996.

Dasgupta, K., Mondal, J. K., e Dutta, P. (2013). Optimized video steganography using genetic algorithm (ga). *Procedia Technology*, 10:131–137. ISSN 2212-0173. URL https://www.sciencedirect.com/science/article/pii/S2212017313004994. First International Conference on Computational Intelligence: Modeling Techniques and Applications (CIMTA) 2013.

Djebbar, F., Ayad, B., Meraim, K. A., e Hamam, H. (2012). Comparative study of digital audio steganography techniques. *EURASIP Journal on Audio, Speech, and Music Processing*.

Elshare, S. e EL-Emam, N. (2018). Modified multi-level steganography to enhance data security. *International Journal of Communication Networks and Information Security*, 10:509–525.

Saini, H., Rao, Y. S., e T.C.Panda (2012). Cyber-crimes and their impacts. *International Journal of Engineering Research and Applications*, 2:202–209.

Siar, F., Alirezazadeh, S., e Jalali, F. (2018). A novel steganography approach based on ant colony optimization. In *2018 6th Iranian Joint Congress on Fuzzy and Intelligent Systems (CFIS)*, p. 215–219.

Walia, G. S., Makhija, S., Singh, K., e Sharma, K. (2018). Robust stego-key directed lsb substitution scheme based upon cuckoo search and chaotic map. *Optik*, 170:106–124. ISSN 0030-4026. URL https://www.sciencedirect.com/science/article/pii/S0030402618306399.

Wazirali, R., Alasmary, W., Mahmoud, M. M. E. A., e Alhindi, A. (2019). An optimized steganography hiding capacity and imperceptibly using genetic algorithms. *IEEE Access*, 7:133496–133508.

DOI: 10.59254/sbpo-2024-193420
https://proceedings.science/p/193420?lang=pt-br