The ISC Int'l Journal of Information Security

January 2026, Volume 18, Number 1 (pp. 1–17)

http://www.isecure-journal.org

Enhancement of LSB Matching Steganography using Multiobjective Optimization Embedding to Improve Security and Imperceptibility

Vajiheh Sabeti [1,*]

[1] *Department of Computer Engineering, Faculty of Engineering, Alzahra University, Tehran, Iran.*

ARTICLE INFO.

Article history:
Received: September 10, 2024
Revised: May 13, 2025
Accepted: September 10, 2025
Published Online: September 17, 2025

Keywords:
Steganography, Steganalysis, Least Significant Bit Matching (LSBM), Non-Dominated Sorting Genetic Algorithm II (NSGA-II).

Type: Research Article

doi: 10.22042/isecure.2025.477842.1172

ABSTRACT

Least Significant Bit Matching (LSBM) is a simple steganography approach that has been detected under multiple attacks. Imperceptibility (i.e., maintenance of high perceptual image quality) and security are significant parameters in steganography. However, most conventional steganography techniques rely on single-objective optimization, which focuses on improving one parameter while often compromising others. This limitation underscores the need for approaches that balance conflicting objectives. To address this, the present study employs the Non-Dominated Sorting Genetic Algorithm II (NSGA-II) to optimize security and imperceptibility. This methodology includes a cover image division into blocks, each with two critical decisions: (1) seed determination for the pseudo-random number generator to simultaneously identify optimal pixels for data embedding and (2) selecting whether the pixel value should be increased or reduced upon a mismatch between the data bit and pixel LSB. Pixels with the highest data bit–LSB correspondence are optimal, and a pixel value change (increase or reduction) is to minimize block histogram variation. This multiobjective optimization is carried out using NSGA-II. It was comparatively revealed that the developed methodology remarkably improved image quality metrics and decreased detection accuracy at different embedding rates. At embedding rates of 0.3, 0.5, and 0.8 bpp, the Peak Signal-to-Noise Ratio (PSNR) was approximately 57.65, 55.55, and 52.75, respectively. This result represents a 1.5-2.5% improvement compared to conventional LSBM techniques.

1 Introduction

Steganography is primarily aimed at hiding confidential data transmission and protecting sensitive data by concealing information in digital media to mitigate any suspicion concerning the existence of hidden content. Unlike cryptographic approaches, which often indicate the presence of sensitive data, steganography hides the existence of the message itself [1]. The data embedding process employs a particular algorithm to embed encrypted and compressed data within a cover media, creating a stego media intended for transmission to the recipient. It is often necessary to utilize one or more embedding keys to improve security, which must be transmitted to the recipient through a secure channel. Images are the

* Corresponding author.
Email address: v.sabeti@alzahra.ac.ir
ISSN: 2008-2045

most favored type of media due to their diverse formats, significant capacity, and extensive applications, making them one of the most widely used cover media in steganography [2].

Steganography methods operate in spatial or transform domains to embed data [3]. Spatial techniques modify pixel values directly using rule-based, deep learning, cost-based, texture- and edge-aware, or adversarial methods. At the same time, transform-domain approaches work on frequency coefficients and offer better robustness [4]. Standard embedding methods include LSB replacement, pixel value difference [5], difference expansion [6], and modification direction [7]. LSB-based schemes are popular due to low visual distortion, with LSB Matching (LSBM) and LSB Flipping (LSBF) being widely used. Despite this, LSBF is vulnerable to several attacks [8–11], prompting enhancements like edge-based embedding [12], AI-driven methods [13], and bit-reversal techniques [14].

In LSBM, the bit is modified by ± 1 to randomly adjust the pixel value when a mismatch occurs between the data bit and the pixel's LSB. This approach offers greater security than LSBF, even though both methods provide the same data embedding capacity. However, advancements in steganalysis have introduced multiple attacks aimed at LSBM detection. The effectiveness of these attacks varies depending on the characteristics of the cover image [15]. The detection of steganography by attacks primarily stems from statistical changes in the image due to data embedding. It is necessary to develop steganography techniques based on strategies that minimize such changes to address this challenge. The literature on this subject has introduced multiple LSBM improvement practices, with some researchers implementing intelligent embedding location selection. In contrast, others focus on purposeful decision-making within the data embedding process. The latter approach pursues particular objectives, such as pixel variation minimization [16], histogram variation reduction (e.g., through GLSBM [17]), and added noise reduction [18] rather than random changes in the image pixel value.

The use of optimization algorithms to enhance existing steganography methods has proven to be a successful idea in this field. Despite significant advancements in optimization-based steganography, most existing approaches focus on enhancing a single parameter—such as security, embedding capacity, or perceptual transparency—without adequately addressing the inherent trade-offs between these conflicting objectives. This limitation often results in suboptimal performance, where improving one aspect compromises another, making current methods either more detectable or less effective in terms of data concealment. While multi-objective optimization has been successfully applied in image watermarking [19–22] and audio/video steganography [23, 24], its potential in image steganography remains underexplored mainly [25, 26]. With the growing sophistication of steganalysis techniques, developing adaptive steganographic frameworks that balance these factors is crucial.

To address this gap, this study develops an LSBM-based technique leveraging multiobjective optimization to enhance security and perceptual transparency while minimizing embedding-induced distortions simultaneously. The proposed method solves a bi-objective optimization problem using the Non-Dominated Sorting Genetic Algorithm II (NSGA-II algorithm). The method optimizes pixel selection and modification by minimizing both the number of pixel changes and histogram variations, thereby enhancing security and perceptual transparency. Experimental results demonstrate that the proposed approach achieves superior stego image quality and lower detection accuracy against steganalysis attacks compared to conventional LSB-based methods. The primary contributions of the present study are as follows:

- Modeling pixel selection and modification process in data embedding using bi-objective optimization.
- Utilizing two objective functions to improve stego image quality and reduce embedding-induced histogram variation.
- Demonstrating the higher Peak Signal-to-Noise Ratio (PSNR) and reduced detection accuracy of the proposed methodology under LSBM attacks relative to earlier works.

The remainder of the paper is organized as follows: Section 2 conducts a review of the literature on optimization-based steganography approaches; Section 3 provides a comprehensive description of the proposed methodology; Section 4 presents the findings, providing comparisons to earlier works; and Section 5 concludes the article, outlining future directions.

2 Related Work

Steganography methods can be categorized in various ways. Given the increasing usage of deep learning approaches in this field, similar to their application in other areas of image processing, one prominent classification includes the following three categories: traditional methods, Convolutional Neural Network-based (CNN-based) methods, and Generative Adversarial Network-based (GAN-based) methods. Traditional methods often rely on techniques, such as LSB, and do not employ machine learning. In contrast, CNN-

based methods utilize deep convolutional neural networks for embedding and extracting hidden messages, while GAN-based methods leverage variants of generative adversarial networks [27].

In the context of steganography, achieving zero extraction error is a fundamental prerequisite for flawless data recovery. Conversely, Deep Learning-based (DL-based) methods, including CNNs [28] and GANs [29], do not ensure zero error. Instead, these techniques concentrate on attaining a high degree of similarity between the extracted and original data, which is frequently assessed using measures such as PSNR or Similarity Index Measure (SSIM). While deep learning techniques emphasize a trade-off between security, imperceptibility, and reconstruction accuracy, traditional methods promote accurate data recovery. The majority of deep learning-based steganography techniques, particularly those embedding one image within another, have significantly increased their embedding capability compared to traditional methods [30]. Nevertheless, the approach suggested in this study aims to embed data into an image while guaranteeing zero extraction error.

Several recent advancements in techniques have aimed to enhance LSB-based approaches without using optimization. For instance, Tseng *et al.* [31] have introduced a system based on a new LSB sorting method and Exploiting Modification Direction (EMD). Kim *et al.* [32] have integrated EMD and optimal LSB replacement, thereby developing a reversible steganography technique with improved performance. Nguyen *et al.* [33] have proposed a novel algorithm that integrates the Lah Transform, Hamming Code, and LSBM Revisited. Furthermore, researchers have integrated Pixel-Value Differencing (PVD) and LSB to develop methodologies boasting large embedding capacities [34].

In addition to conventional LSB-based improvements, several studies have explored optimization techniques to enhance security and capacity, leading to various algorithmic approaches. Optimization-based steganography approaches employ algorithms such as Genetic Algorithm (GA), Ant Colony Optimization (ACO), Particle Swarm Optimization (PSO), Whale Optimization Algorithm (WOA), Haris Hawk Optimization (HHO), and Neural Networks (NN).

Both Spatial-domain and transform-domain steganography systems leverage optimization algorithms, which are implemented in three phases: (I) pre-embedding phase, (II) embedding phase, and (III) post-embedding phase. During the pre-embedding phase, optimization is performed to identify the optimal embedding locations or modify the data bits. During the embedding phase, optimization aims to determine the stego image pixel values. During the post-embedding phase, optimization is used to minimize embedding-induced variations [35]. This study reviews previous spatial-domain techniques that incorporate optimization.

Table 1 presents earlier works on improving LSB-based embedding performance through optimization. In general, previous studies aimed at enhancing the embedding capacity have not reported significant improvements in image quality or sufficient enhancements in security [36, 37]. However, certain studies improved image quality with reduced embedding capacity by a maximum of one bit per image pixel [38, 39]. Additionally, researchers have noted the utilization of LSBM as a base technique [17, 40], as LSBM-based embedding can improve security.

Sabeti *et al.* [17] integrated GA with LSBM to propose two algorithms: GLSBM and MKGM. The GLSBM algorithm integrates GA in LSBM, allowing for the choice between increasing or reducing mismatching pixels. The MKGM algorithm was developed to further enhance GLSBM by dividing the cover image into different blocks and implementing GLSBM on each block with multiple distinct keys. The block with the least histogram variation is then selected as the stego image. The test results indicated that MKGM had a lower detection rate compared to previous approaches and the standard LSBM method. However, MKGM demonstrated almost identical PSNR results, with no significant improvements.

A dual Genetic Algorithm (2GA) approach was developed to enhance the MKGM algorithm [49] further, dividing the cover image into blocks and embedding data into the blocks in two phases. The first phase combines GA with a Linear Congruential Generator (LCG) function, selecting pixels from the block with the highest correspondence with the message sequence for data embedding. The second phase employs GLSBM to embed data in the selected pixels in order to minimize histogram variations within the block. Despite the different security levels and stego image quality rates in LSBM-based techniques, MKGM enhanced GLSBM in terms of security, while 2GA improved MKGM in terms of stego image quality.

While previous LSBM-based methods have enhanced security or stego image quality individually, they have not effectively addressed the trade-off between these objectives. This study introduces an multiobjective optimization approach to simultaneously improve both aspects, marking a significant advancement over previous techniques.

Table 1. Summary of LSB-based spatial domain methods using optimization algorithms

Ref.	Embedding Method	Optimization Algorithm	Objective Function	Optimization Goal*	Advantages (+) Disadvantages (-)**
[41]	LSBF	GA	PSNR	2	+Medium EC -Low SQ -Unproven against strong attacks
[42]	LSBF	GA	PSNR	1	+Medium EC -Low SQ -Unproven against strong attacks
[43]	LSBF	GA	PSNR	4	+Medium EC -Low SQ -Unproven against strong attacks
[44]	LSBF	GA	MSE	4	+Medium EC +Medium SQ -Unproven against strong attacks
[45]	LSBF	ACO	MSE	1	+Medium EC +Medium SQ -Unproven against strong attacks
[38]	LSBF	GA	The number of LSB's pixel-data bit matches	1	+Medium SQ -Low EC -Unproven against strong attacks
[46]	LSBF	ABC	MSE	1	+Medium EC +Medium SQ -Unproven against strong attacks
[39]	LSBF	ACO	Difference between neighboring pixels	1	+Medium SQ -Low EC -Unproven against strong attacks
[36]	LSBF	GA	PSNR	1,2	+High EC -Low SQ -Unproven against strong attacks
[47]	LSBF	PSO	PSNR	1	+Medium EC +High SQ -Unproven against strong attacks
[17]	LSBM	GA	Histogram difference	3	+LSBM-based +Good SS +Medium SQ +Medium EC
[40]	LSBM	GA	The number of LSB's pixel-data bit matches	1	+LSBM-based +High SQ +Good SS +Medium EC
		GA	Histogram difference	3	
[37]	LSBF	GA	PSNR	1,2	+Medium SQ +Medium EC -Unproven against strong attacks
[48]	LSBF	ACO	MSE	1	+Medium EC +Medium SQ -Unproven against strong attacks
[49]	LSBF	WOA	Different error functions	1,4	+Medium EC +Medium SQ -Unproven against strong attacks
[50]	LSBF	HHO	PSNR	2	+Medium EC +Medium SQ +Good SS

* 1. Find embedding locations 2. Modify message bits 3. Determine the stego pixels 4. Modify the stego image
** **EC**: Embedding Capacity **SQ**: Stego Quality (Imperceptibility) **SS**: Stego Security

2.1 NSGA-II Algorithm

Optimization problems involve finding the best solution from a set of feasible candidates, typically by maximizing or minimizing one or more objective functions. Such problems are widely encountered in fields like engineering, finance, and artificial intelligence. Optimization problems can be categorized as single-objective (optimizing a single criterion) or multiobjective (optimizing multiple, often conflicting criteria). Evolutionary optimization algorithms, a subset of stochastic methods, are inspired by natural selection and iteratively improve a population of candidate solutions. GAs, a widely used evolutionary approach, evolve solutions through selection, crossover, and mutation. GAs are particularly effective for single-objective optimization but can be extended to multiobjective problems.

The NSGA-II is a widely used multiobjective optimization algorithm designed to find a set of optimal solutions (Pareto front) for problems involving conflicting objectives. It operates through the following key steps:

- Initialization: A population of solutions is initialized randomly or based on prior knowledge.
- Evaluation: Each solution is evaluated using objective functions to determine its performance.
- Non-dominated Sorting: Solutions are ranked into different fronts based on dominance. A solution dominates another if it is better in at least one objective and no worse in others.
- Crowding Distance: Within each front, solutions are ranked based on diversity (crowding distance) to maintain a well-distributed Pareto front.
- Selection: Parent solutions are selected using a combination of rank and crowding distance.
- Crossover and Mutation: Genetic operators are applied to generate new offspring solutions.
- Elitism: The algorithm combines parent and

offspring populations, retaining the best solutions based on rank and diversity for the next generation.

The flowchart of the NSGA-II algorithm is shown in Figure 1. A detailed explanation of these steps in the proposed method will be provided. Key parameters include the population size, number of generations, crossover probability, mutation probability, and distribution indices for crossover and mutation, which control the spread of solutions.

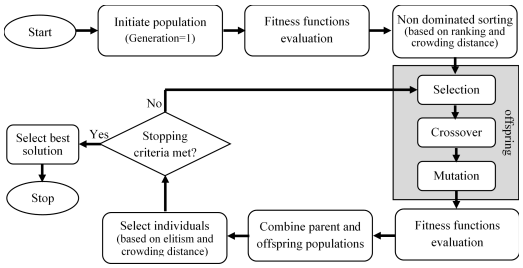


Figure 1. The flowchart of the NSGA-II algorithm

3 Proposed method

Single-objective optimization aims to identify the optimal solution through a single objective function, while real-life problems may require optimizing two or more objective functions that can be conflicting or not directly associated. Steganography mainly includes analysis and optimization of several objectives, and the developed technique adopts NSGA-II for multi-objective optimization so that LSB-M performance can be improved. Figure 2 depicts a block diagram of the Multi-Objective-based LSBM (MOLSBM) method. The initial version of this method was presented in [51], and here, a complete version with practical implementation capability is introduced. Table 2 provides the MOLSBM parameters.

MOLSBM embeds data by implementing the following steps on the cover image for generating the stego image before it is transmitted to the receiver:

(1) Splitting: Partitioning the cover image and secret data based on the block size.
(2) Repeat the following steps for the cover image blocks:
  (2-1) Finding the best seed and optimal embedding pattern using NSGA-II to embed the data into the block of the cover image.
  (2-2) Finding the target embedding pixels through the best seed and calculating pixel values via the optimal pattern for the stego block.
(3) Rebuilding: Arrange the output blocks to reconstruct the stego image.

The best seed value for each block is also sent to enable the receiver to extract the entire data with no errors. MOLSBM models the embedding process of an image block in the form of a bi-objective optimization problem solved through NSGA-II. By employing the NSGA-II algorithm, an optimal key is adaptively selected for each specific cover image based on the proposed objective functions. This image-dependent key selection strategy not only enhances the quality of the resulting stego image but also significantly improves security. Since the key is tightly coupled with the unique characteristics of the cover image, it becomes complicated—if not infeasible—for an adversary to determine or predict the key without access to the original cover image. NSGA-II is a rapid and efficient multiobjective genetic algorithm. Once the two objective functions have been defined, the details of implementing the NSGA-II steps and the extraction algorithm are described.

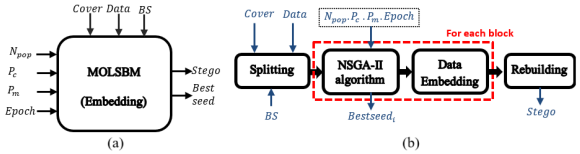


Figure 2. (a) Inputs and outputs, and (b) embedding steps of MOLSBM

Table 2. Abbreviations in MOLSBM

Name	Description
Cover	Cover image
Stego	Stego image
Data	Secret data
C_i	Block i^{th} of the cover image
D_i	Section i^{th} of the data
S_i	Block i^{th} of the stego image
DS	Data length for a block
$P(t)$	Population in epoch t^{th}
$Q(t)$	The population resulting from crossover in epoch t^{th}
N_{pop}	Population number
P_c	Crossover probability
P_m	Mutation probability
Epoch	Iteration number
BS	Block size
N_c	Number of Population resulting from crossover
N_m	Number of Population resulting from mutation
seed_i	Seed for the i^{th} block
Chr_j	j^{th} chromosome

3.1 Objective Functions

It is essential to choose the optimal pixels for data embedding in steganography. Apart from finding such pixels, it is also required to determine the embedding sequence. The entire set of pixels is to be utilized for 100% data embedding; however, the embedding sequence can be different. For an embedding rate below 100%, the optimal pixels and their embedding

sequence should be determined based on the data length. These steps allow the receiver to accurately find the same pixels and embedding sequence as those of the sender, so that the data can be extracted entirely. The use of pseudorandom number generation functions is a typical idea, and different forms of the function *rand*() are available in MATLAB. Such functions should have an initial seed, and their alteration would change the sequence of the generated numbers; i.e., a different set of pixels may be chosen in different cases.

LSBM, however, either changes the pixel value or keeps it unchanged, depending on the pixel value and bit value. As a result, the number of modification-requiring pixels can vary in each pixel sequence. The best embedding sequence is to be selected from the set of available sequences in order to minimize the pixel change rate. A more minor change in the cover image for generating the stego image is expected to provide higher stego image quality. The identification of the best seed value to minimize pixel changes could be modeled in the form of an objective function in multiobjective optimization.

Embedding the message in pixels to maximize stego image security can be the second objective function. To embed data into modification-requiring pixels, LSBM either increases or reduces the pixel value by one unit, each of which might affect the generated stego image. A review of LSBM-revealing attacks has shown that attacks exploit some parameters, e.g., the center of gravity [52] and local extrema [53]. Thus, the reduction of embedding-induced image histogram variations appears to diminish LSBM vulnerability. As MOLSBM is intended to maximize security, it is required to minimize stego image histogram variation in comparison to the cover image; larger histogram variations facilitate steganalysis attacks in detecting the steganography system. Hence, the histogram difference between the cover image and stego image blocks is the second objective function.

3.2 Chromosome Structure and Evaluation

The chromosome structure in MOLSBM and its evaluation based on the objective functions are discussed in this section. Considering the two objective functions, MOLSBM is comprised of two components (Figure 3). The seed for the function *rand*() is the first component of the chromosome that determines the embedding path. The second component determines the required pixel variation. The first eight genes of a chromosome are responsible for seed value encoding, whereas the other genes define pixel modification. The second component's size is dependent on the embedding rate and block size; the data size (DS) for a block size of 16×16 and an embedding rate of 50% is given by: $DS = 16 \times 16 \times 0.5 = 128$. Therefore, the chromosome has a size of 8+128=136 bits. The chromosomes of MOLSBM are written as:

$$Chr_j = [g_1, g_2, \ldots, g_{DS+8}](g_1, \ldots, g_8) \in \{0, 1\}, \\ (g_9, \ldots, g_{DS+8}) \in \{+1, -1\}, j \in [1, N_{\text{pop}}] \quad (1)$$

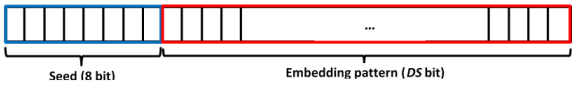


Figure 3. Chromosome structure

NSGA-II requires a random initial population. To initialize a chromosome, an integer from 0 to 255 is to be randomly selected. It is converted into a binary form and introduced to the first eight genes. Then, the number of increments/decrements (i.e., +1 or -1) equaling the data size is selected for the genes in the second chromosome component. The genes in the second component are responsible for determining the pixel values after the data have been embedded.

The randomly generated chromosomes are to be evaluated through the objective functions. To calculate the first objective function, the first 8 bits of the chromosome are used as a seed. Then, a random selection of pixel numbers from a block is made (the number of pixels chosen corresponds to the length of the desired data bit sequence). The first data bit is compared with the LSB of the first selected pixel, the second data bit is compared with the LSB of the second selected pixel, and this process continues until the last data bit. The number of pixels whose LSB does not match the corresponding data bit is counted, and this value is used as the objective function's value. If the kth selected pixel is represented by pixel(k) and the kth data bit is represented by D(k), the value of UnEqbit is calculated according to Equation 2.

$$UnEqbit = \sum_{k=1}^{DS} |LSB(pixel(k) - D(k)| \quad (2)$$

The first fitness function pattern of MOLSBM is given as:

$$F_{i,j}^1 = Mismatch(C_i, D_i, Chr_j) \quad (3)$$

Algorithm 1 depicts the pseudocode for the calculation of the first objective function. This function calculates the value of UnEqbit for each block. A larger value of a given chromosome represents higher desirability.

The second objective function measures the histogram difference between the cover and stego image

Algorithm 1 Mismatch

Require: C_i, D_i, Chr_j
Ensure: $F^1_{i,j}$
$[BS, BS] \leftarrow \text{size}(C_i)$
$DS \leftarrow \text{size}(D_i)$
$F^1_{i,j} \leftarrow 0$
$C_{i\_vec} \leftarrow \text{reshape}(C_i, 1, BS \times BS)$
$Seed \leftarrow \text{bi2dec}(Chr_j(1{:}8))$
$\text{rng}(Seed)$
$Seq \leftarrow \text{randperm}(BS \times BS, DS)$
for $k \leftarrow 1$ to DS **do**
  $pixle \leftarrow C_{i\_vec}(Seq(k))$
  if $\text{mod}(pixle, 2) \neq D_i(k)$ **then**
    $F^1_{i,j} \leftarrow F^1_{i,j} + 1$
  end if
end for

blocks. The stego block is generated using the seed and genes in the second component of chromosomes. Once the embedding pixel sequence has been determined, no change is required for the generation of the stego block when the pixel LSB matches the message bit, and the cover block pixel value is transmitted to the corresponding stego block location. In the case of a mismatch, however, the gene value in the second component of the chromosome would be added to the cover block pixel value, transmitting the resulting pixel value to the corresponding stego block location. Once the stego block has been generated for all chromosomes, the second objective value is calculated. Let H_c and H_s be the histograms of the cover block and stego block generated by Chr_j, respectively. The $H_c - H_s$ difference in a grayscale image is given by:

$$Dif(H_c, H_s) = \sum_{k=0}^{255} |H_c(k) - H_s(k)| \tag{4}$$

The second fitness function pattern is written as:

$$F^2_{i,j} = HistDif(C_i, D_i, Chr_j) \tag{5}$$

Algorithm 2 illustrates the pseudocode for calculating the second objective function. A smaller objective function would represent higher desirability.

Figure 4 shows the calculation of the objective functions for a random chromosome for an 8×8 cover block and a 16-bit data sequence. The first eight genes of the chromosome correspond to the number 100 used for the random selection of sixteen pixels from the block. The mismatch column compares the LSBs of such pixels to the corresponding data bit. The first objective function $F^1_{i,j}$ equals the number of check-marked columns; here, $F^1_{i,j} = 9$.

Algorithm 2 HistDif

Require: C_i, D_i, Chr_j
Ensure: $F^2_{i,j}$
$[BS, BS] \leftarrow \text{size}(C_i)$
$DS \leftarrow \text{size}(D_i)$
$C_{i\_vec} \leftarrow \text{reshape}(C_i, 1, BS \times BS)$
$S_{i\_vec} \leftarrow C_{i\_vec}$
$Seed \leftarrow \text{bi2dec}(Chr_j(1{:}8))$
$\text{rng}(Seed)$
$Seq \leftarrow \text{randperm}(BS \times BS, DS)$
for $k \leftarrow 1$ to DS **do**
  $pixel \leftarrow C_{i\_vec}(Seq(k))$
  if $\text{mod}(pixel, 2) \neq D_i(k)$ **then**
    $pixel' \leftarrow pixel + Chr_j(8 + k)$
    $S_{i\_vec}(Seq(k)) \leftarrow pixel'$
  end if
end for
$S_i \leftarrow \text{reshape}(C_{i\_vec}, BS, BS)$
$H_c \leftarrow \text{imhist}(C_i)$
$H_s \leftarrow \text{imhist}(S_i)$
$F^2_{i,j} \leftarrow \text{sum}(\text{abs}(H_c(:) - H_s(:)))$

The stego block is generated by altering merely the pixels with a mismatching LSB, and the new values of these pixels are found by adding the value of the corresponding gene from the second component of the chromosome to the original pixel value. For example, the first pixel is in position 58 and has a value of 60, while bit 1 requires embedding. As a value of 60 has an LSB of 0 (mismatching), the mismatching column is check-marked. To obtain an LSB of 1, the pixel value can be altered to either 59 or 61. As the first gene of the second component is +1, the value is increased by 1, with the pixel value in position 58 of the stego block becoming 61. For the second pixel, on the other hand, no mismatch exists between the LSB of the number 84 and bit 0, and the pixel value of 84 remains unchanged in position 31 of the stego block. This procedure would be performed for all sixteen pixels so that the stego block can be generated.

The histogram difference between the cover and stego blocks stands for the second objective function's value. The second objective function equals 12 for this chromosome.

3.3 Crossover and Mutation

Crossover and mutation are implemented on the population $P(t)$ to generate a new population $Q(t)$. Let P_c and N_{pop} denote the population fraction undergoing crossover and the population size, respectively. The crossover offspring N_c is given by:

$$N_c = \lfloor P_c * N_{pop}/2 \rfloor * 2 \tag{6}$$

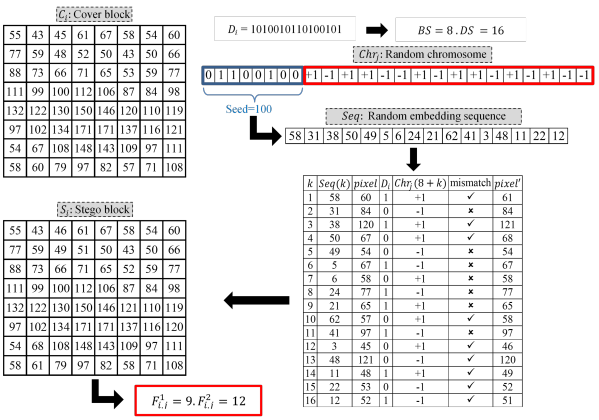


Figure 4. Objective function calculation for a random chromosome

Two members of the population $P(t)$ are parents in each crossover. Two new members are created via single-point crossover to generate the new population $Q(t)$. Once crossover has been completed, it would be required to ensure the absence of duplicate chromosomes. The fitness value of a pair of chromosomes is then obtained and added to $Q(t)$.

Let P_m be the population fraction undergoing mutation. The number of new chromosomes generated in mutation N_m is given by:

$$N_m = \lfloor P_m * N_{pop} \rfloor \tag{7}$$

A member of the population $P(t)$ is chosen in each mutation phase, and mutation is implemented on the chosen member. A random number would be chosen between 1 and DS+8 to implement mutation on a chromosome. The first component of the chromosome is changed when this random number is equal to or smaller than 8. In such a case, if the gene value is 0, it is increased to 1 and vice versa. For a random number larger than 8, on the other hand, the second component of the chromosome is changed. For a gene value of -1, the gene value is changed into +1 and vice versa. The fitness value of the chromosome is then obtained and added to $Q(t)$. Figure 5 represents one crossover and two mutations.

3.4 Selection of a New Population and Final Optimal Solution

Once crossover and mutation have been completed, the chromosomes of the initial population are integrated with the crossover and mutation chromosomes:

$$R(t) = P(t) \cup Q(t) \tag{8}$$

However, solely N_{pop} chromosomes are chosen from the integrated set to generate a new population. Figure 6 depicts the chromosome selection of NSGA-II. The integrated population is arranged based on a non-dominated sorting function, classifying the members into fronts. Then, the crowding distance would be found for all members based on their fronts. The arranged population is employed to choose the first N_{pop} chromosomes as the new population $P(t+1)$.

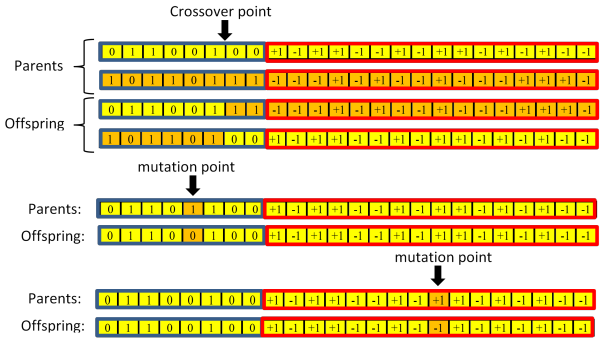


Figure 5. One crossover and two mutations

NSGA-II performs such steps MaxIt times, and the Pareto front is generated. The best chromosomes in the Pareto front should be considered the final solution, directing data embedding within the cover block and leading to a stego block. MOLSBM adopts the Linear Programming Technique for Multidimensional Analysis of Preference (LINMAP) decision-making model [54]:

$$ED_i = \sqrt{(\sum_{j=1}^{n}(F_{ij} - F_j^{ideal}))} \tag{9}$$

where F_{ij} is the objective j of member i in the Pareto front, F_j^{ideal} stands for the ideal solution produced by single-objective optimization for objective j, and n is the number of objectives. The Euclidean distance (ED) is obtained for the Pareto front members, with the member having the shortest ED being the best chromosome. Here, $Bestseed_i$ represents the initial component of the best chromosome and should be shared with the receiver. This process is performed for all the blocks in the cover image to generate the stego image.

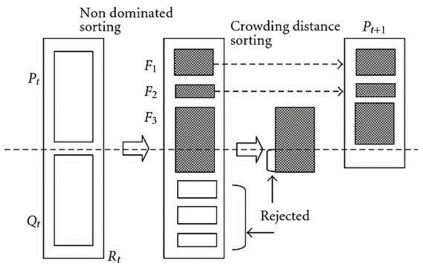


Figure 6. Selection process of NSGA-II [55]

3.5 Extraction Algorithm

An extraction algorithm is to be used in MOLSBM to allow for the accurate retrieval of the embedded data from the stego image by the receiver, as with other embedding approaches. Each block in MOLSBM has an eight-bit key for embedding path identification. It is necessary that the receiver effectively obtain the keys for accurate extraction. In scenarios where a secure channel for key exchange exists between the sender and receiver, this process can be easily handled.

However, due to the absence of such a secure channel in practical applications, the proposed method embeds the key within the cover image after the confidential data has been hidden. This method requires the sender and recipient to agree beforehand on the blocks that will contain the key. For example, the blocks may be selected from the image's first and last rows and columns, or chosen at random using a brief pre-shared key. The length of the data embedded in the cover is therefore equal to the sum of the lengths of the original secret data and the key that is produced by the embedding of the secret data. The key length and, consequently, the length of the embedded data increase when the block size is decreased. The effect of this parameter on the MOLSBM method's performance is evaluated in the evaluation section.

4 Experimental Results

MOLSBM performs decision-making using NSGA-II in two steps: (1) choosing embedding pixels and (2) determining how to modify pixel values during embedding. The method aims to improve the quality and security of the stego image relative to the LSBM baseline and earlier methodologies by developing practical objective functions in multiobjective optimization. To enhance the practical relevance of the approach, a key transmission strategy has been integrated in this version. As this strategy introduces additional modifications to the stego image, the evaluation results reported below differ from those presented in the earlier version [51] and more accurately reflect real-world usage scenarios. Moreover, a broader, more comprehensive set of evaluation experiments has been conducted to assess the proposed method from multiple perspectives. These results, presented in the following subsections, provide deeper insight into the performance and robustness of the approach. The algorithm was performed in MATLAB 2020 based on the NRCS image dataset. This dataset involves nearly 2,000 images that cover several subjects, mostly nature scenes. A total of 1,000 random images were chosen for testing.

A random sequence of 0s and 1s generated by function *rand*() was used to embed data in the test image. This data is an efficient embedding sequence for secret data and does not require prior data encryption or compression. For each embedding stage, the embedding key is also added to the generated data. Therefore, the data being embedded is slightly larger than the original secret data. The tests were primarily conducted at three embedding rates. The embedding rate is the ratio of the data bits to the total image pixels (bits per pixel); e.g., an embedding rate of 0.5 bpp refers to embedding one data bit in half of the pixels. A population of 20 was employed in NSGA-II, with crossover and mutation probabilities of 0.7 and 0.1, respectively, and a total of twenty iterations.

Stego image quality, security, and embedding capacity are the main criteria to compare steganography methods. As all the results can be presented, only the primary outcomes are discussed. Once the optimal block size has been identified for MOLSBM, the performance of the proposed algorithm is evaluated using various criteria in comparison to earlier methodologies.

4.1 MOLSBM Performance Versus Block Size

The block size (BS) is a major evaluation criterion and can significantly affect algorithm performance. Testing is performed to verify the effect of BS on performance and choose the optimal block size for MOLSBM. Four images were employed as samples, including Lena, Baboon, Peppers, and Boat (Figure 7). Table 3 provides the PSNR and SSIM of MOLSBM for various block sizes and an embedding rate of 0.5 bpp. A smaller block size was found to produce higher stego image quality. This result was expected as more efficient decisions are enabled for smaller regions in data embedding. The reduction of the embedding space and the use of each region's characteristics contribute to data embedding decision-making. However, MOLSBM generates an eight-bit key ($Bestseed_i$) for blocks to be embedded in the stego image itself, and a rise in the number of blocks would result in a longer final key. As the number of blocks increases, the final key length also increases, which in turn enlarges the amount of embedded data in the image and consequently raises the level of distortion in the stego image.

For a trade-off between the key length and block size, the sender may set the optimal block size under the existing circumstances. This study employed block sizes of 8×8 and 16×16 for further evaluation. An eight-bit key would be necessary to embed 256 bits per block, leading to a key/data length ratio of 0.03 for a block size of 16×16. At a block size of 8×8,

however, an eight-key bit is to be used to embed 64 bits, and the key/data length ratio becomes 0.125. This method ensures secure and efficient key embedding while managing the trade-off between image quality and data capacity.

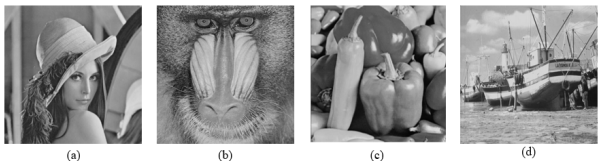


Figure 7. Images of (a) Lena, (b) Baboon, (c) Peppers, and (d) Boat

4.2 Comparative Analysis of Stego Image Quality

Improving stego image quality was a primary objective of the proposed algorithm. Four sample images were employed to evaluate the performance of the MOLSBM algorithm in stego image improvement (Figure 7). Given the similarity between MOLSBM and other methods like LCG [37], MKGM [17], and 2GA [49], the proposed approach was compared to these algorithms for further comparative analysis.

Table 4 provides a comparison between the PSNRs at embedding rates of 0.3, 0.5, and 0.8 bits per pixel (bpp). The average PSNR was calculated using 1,000 sample images sourced from the NRCS database for further comparative analysis (Figure 8). As presented in Table 4 and Figure 8, the MOLSBM algorithm outperformed LSBM, LCG, MKGM, and 2GA in stego image quality.

Furthermore, MOLSBM was compared with the optimization-based techniques, followed by comparisons with non-optimized LSB approaches, including the models proposed by Tseng [45], Kim [38], and Nguyen [46]. Figure 9 illustrates the average PSNRs across the three embedding rates. The results suggest that MOLSBM outperformed earlier non-optimized LSB models in stego image quality improvement. This improvement can be attributed to the role of NSGA-II in identifying optimal embedding locations with minimal cover image variation.

In addition to PSNR, other metrics such as Image Fidelity (IF), Pearson Correlation Coefficient (PCC), Correlation Coefficient (CC), and SSIM were employed to evaluate the quality of stego images. These metrics provide values ranging from 0 to 1, where values approaching 1 suggest a closer resemblance between the cover and stego images, thus complicating the task of locating embedded data for steganalysis algorithms. A set of 200 images was selected for this study from the NRCS dataset, with the metrics being computed using both the LSBM and MOLSBM methods following embedding at a rate of 0.5 bpp. The results are illustrated within the charts of Figure 10, where the horizontal axis represents the image number, and the vertical axis represents the value of the respective metric. Analysis of these charts reveals that the MOLSBM method improved all parameters compared to LSBM across all the studied images, demonstrating that stego images produced by the MOLSBM method exhibit greater similarity to the cover images.

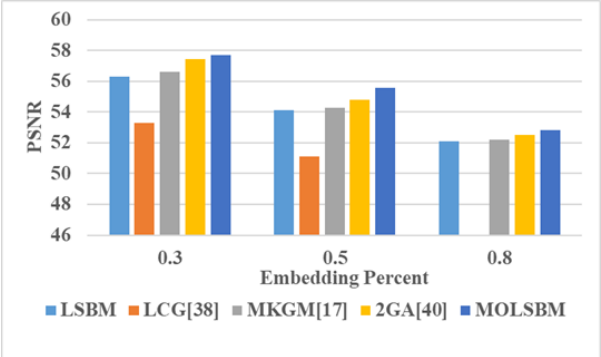


Figure 8. PSNR comparison of optimization-based embedding methods at different embedding rates on NRCS images

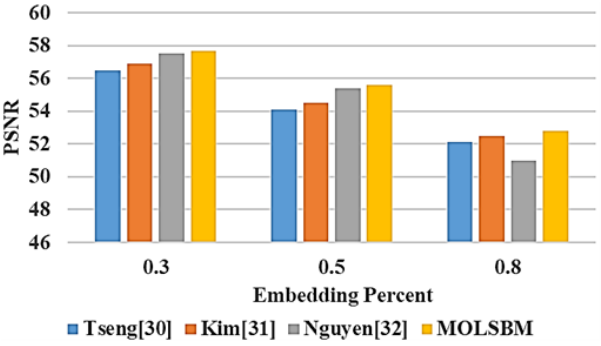


Figure 9. PSNR comparison of non-optimized embedding methods at different embedding rates on NRCS images

4.3 Comparative Analysis of the Embedding Capacity

The embedding capacity of a steganography technique represents the maximum number of data bits that can be embedded in an image. The embedding capacity of spatial-domain approaches is often expressed using bits per pixel (bpp), referring to the ratio of the maximum embeddable bits to the total number of image pixels. Table 5 provides a comparison of embedding capacity between MOLSBM and earlier methods for grayscale images. MOLSBM and 2GA methods, due to the need to embed the key along with the secret data, the actual maximum embedding capacity is slightly less than one bpp.

Several steganography models, including GA2017 [53] and GA2019 [47], enjoy a huge embedding capacity,

Table 3. PSNR and SSIM for the test images at different block sizes (0.5 bpp)

	Lena		Baboon		Peppers		Boat	
BS	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
512	54.16	0.9980	54.16	0.9994	54.17	0.9981	54.16	0.9991
256	54.25	0.9981	54.23	0.9995	54.22	0.9981	54.26	0.9992
128	54.25	0.9981	54.24	0.9995	54.24	0.9982	54.27	0.9992
64	54.34	0.9982	54.33	0.9995	54.33	0.9982	54.34	0.9993
32	54.51	0.9982	54.52	0.9996	54.52	0.9982	54.50	0.9993
16	55.22	0.9983	55.22	0.9997	55.21	0.9983	55.20	0.9993
8	55.52	0.9984	55.58	0.9997	55.51	0.9984	55.56	0.9994

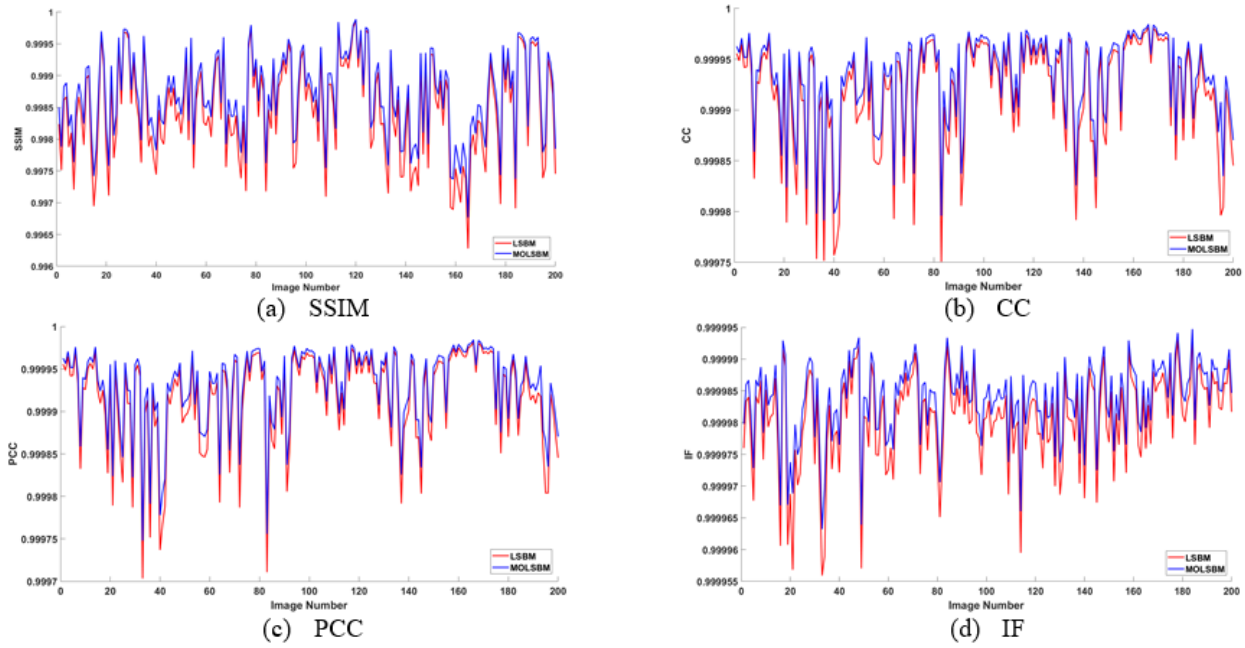


Figure 10. Image quality metrics comparison between LSBM and MOLSBM at an embedding rate of 0.5 bpp

whereas other techniques, such as ACO2018 [48] and LCG [37], exhibit a low embedding capacity. Most steganography methodologies, including MOLSBM, provide a maximum embedding capacity close to 1 bpp. It is important to note that a large embedding capacity is not necessarily an advantage, as specific detection attacks can accurately detect steganography techniques at high embedding rates. Consequently, an efficient steganography model should embed a small dataset while minimizing detection likelihood, thus emphasizing security as a critical parameter within steganography. This aspect was also comparatively evaluated with the same data embedding rates.

All the methods in Table 5 are classical. These classic steganography approaches have a far lower embedding capacity than the majority of DL-based methods, as outlined in Section 2. The embedding capacities of the MOLSBM method and several other DL-based approaches are presented in Table 6. The majority of DL-based techniques can embed a whole image (secret data) within a cover image, while the suggested MOLSBM method is restricted to embedding one bit per pixel. As a result, the embedding capacity of these techniques is usually eight bpp for grayscale images and 24 bpp for RGB color images. It is crucial to remember that these techniques do not guarantee the recovery of all embedded data. The third column in Table 6 depicts the PSNR of the stego image relative to the cover image. In contrast, the fourth column represents the SSIM of the recovered data compared to the embedded data across various methods. The MOLSBM method is a classical approach that is capable of successfully retrieving all the embedded data, while the other methods listed in this table exhibit errors in data recovery. Therefore, these methods are not suitable for transmitting encrypted textual data, as even a single bit error can prevent decoding.

Table 4. PSNR comparison at three embedding rates

	Method	Lena	Baboon	Boat	Peppers
0.3 bpp	LSBM	56.37	56.37	56.39	56.38
	LCG [38]	53.37	53.37	53.37	53.36
	MKGM [17]	56.57	56.68	56.53	56.61
	2GA [40]	57.45	57.42	57.45	57.43
	MOLSBM	**57.66**	**57.65**	**57.66**	**57.67**
0.5 bpp	LSBM	54.15	54.15	54.15	54.14
	LCG [38]	51.16	51.15	51.14	51.16
	MKGM [17]	54.32	54.41	54.26	54.34
	2GA [40]	54.78	54.79	54.76	54.77
	MOLSBM	**55.52**	**55.58**	**55.51**	**55.56**
0.8 bpp	LSBM	52.11	52.11	52.11	52.10
	LCG [38]	-	-	-	-
	MKGM [17]	52.26	52.28	52.16	52.25
	2GA [40]	52.46	52.48	52.47	52.48
	MOLSBM	**52.75**	**52.77**	**52.76**	**52.78**

Moreover, a subset of steganography methods termed Steganography Without Embedding (SWE) exhibits extremely low embedding capacities, with only a few achieving complete data recovery. For instance, methods [56–58] demonstrate embedding capacities of 1.95×10^{-3}, 1×10^{-1}, 6.2×10^{-2} bits per pixel, respectively, while achieving error-free data recovery. However, the significantly lower embedding capacities of these methods remain their primary limitation compared to the MOLSBM method.

Table 5. Maximum embedding capacity for grayscale images

Embedding Method	Embedding Capacity (bpp)
GA2017 [44]	4
ACO2018 [39]	< 0.5
LCG [38]	0.5
GA2019 [36]	4
LCG-GA [37]	1
MKGM [17]	1
2GA [40]	$\approx$1
Tseng [31]	1
Kim [32]	1.1
Nguyen [33]	1
MOLSBM	$\approx$1

4.4 Comparative Analysis of Security

The proposed method aims to produce stego images with a lower likelihood of detection by reducing the number of pixel changes (first objective function) and histogram variations (second objective function). Since stego image identification relies on a collection of features extracted from the pixels and image histogram, lowering these two objective functions ensures that the features of the stego and cover images remain more comparable, thereby lowering the possibility of successful detection attacks. Prior to evaluating the performance against various attacks, the effectiveness of the proposed method in achieving these two objectives is examined. For 200 images sourced from the NRCS dataset, the Mismatch and HistDif values resulting from embedding at 0.5 bpp using both the LSBM and MOLSBM methods were calculated, with the results presented in Figure 11. In section (a), the vertical axis represents the Mismatch value, while section (b) represents the HistDif value, with the horizontal axis indicating the image number. This chart demonstrates that the NSGA-II algorithm effectively lowers the Mismatch and HistDif values across all images when compared to the LSBM technique.

It is essential to minimize detection likelihood in steganography. Considering that LSBF-based techniques exhibit lower security, MOLSBM was expected to outperform such techniques in terms of security. To assess this hypothesis, the MOLSBM approach was compared to LCG [38] as an LSBF-based model, as well as LSBM, MKGM [17], and 2GA [40] for security analysis on 500 images from the NRCS dataset. The attacks included ker1 [52], ker2 [52], CNGL [66], and ALE [53], which are specifically designed to detect LSBM-based embedding models. Table 7 provides a comparison of detection accuracy across the three data embedding rates. Detection accuracy is measured by the area under the ROC curve (AUC), with curves closer to the diagonal indicating higher security or, conversely, lower detection success by the adversarial methods.

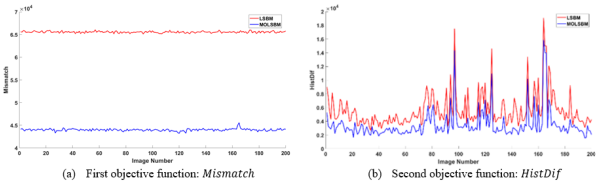
The findings in Table 7 indicate that MOLSBM outperformed its counterparts in steganalysis security, except the CNGL attack at an embedding rate of 0.8 bpp. Notably, the detection accuracy of LCG was left blank, as more than 50% of the data could not be embedded [38]. The highest scores are bolded in Table 7, demonstrating the higher security exhibited by MOLSBM compared to MKGK [17] and 2GA [40]. Additionally, MOLSBM outperformed LSBF-based methodologies, including LCG [38].

Universal steganalysis, which leverages machine learning and feature extraction approaches, is effective against a wide range of steganography methods, including unidentified techniques. This study tested the security of MOLSBM using 1,000 images from the NRCS dataset employing Subtractive Pixel Adjacency Matrix (SPAM) with 686 features and Spatial Rich Model (SRM) with 34,671 features, as well as two SRM variants, maxSRM and maxSRMd2 [67], for

Table 6. A performance comparison with some DL-based techniques

Embedding Method	Approach	Cover vs. Stego-Image PSNR	Secret vs. Retrieved Image Similarity	Embedding Capacity (bpp)
Chen *et al.* [59]	CNN	40.3	0.98	24
Duan *et al.* [60]	FC-DenseNet	40.19	0.98	23.96
Baluja *et al.* [61]	CNN	28.96	0.73	24
Kalifa *et al.* [62]	CNN	44.33	0.93	24
Li *et al.* [63]	GAN	42.3	0.95	8
Liu *et al.* [64]	U-Net	39.77	0.98	8
Gan *et al.* [65]	GAN	38.74	0.97	8
MOLSBM	LSB+NSGA-II	≈52.7	1	≈1

feature extraction, alongside the Ensemble classifier for the classification stage. The findings presented in Table Table 8 indicate that lower test errors and Area Under Curve (AUC) values nearer 0.5 show stronger security. According to these results, the proposed method is quite secure at low embedding rates but demonstrates diminished security as embedding rates rise. As embedding rates rise and more changes occur within the stego image, the AUC increases while test error decreases. This trend emphasizes the unfeasible nature of high embedding rates, as skilled attackers can more readily detect them.



(a) First objective function: *Mismatch* (b) Second objective function: *HistDif*

Figure 11. The values of the objective functions in LSBM and MOLSBM at an embedding rateof 0.5 bpp

4.5 Computational Time

One of the criteria for comparing different algorithms is their computational time. Although this criterion is less significant compared to other metrics in academic papers, it holds particular importance in practical online applications. The proposed method comprises two algorithms: embedding and extraction. While the extraction process is relatively straightforward, the embedding algorithm requires the execution of NSGA-II, which results in a larger computing load. The hardware and the effectiveness of the NSGA-II implementation determine the embedding algorithm's runtime. The runtime of both techniques for three distinct embedding levels is shown in Table 9. The following are the hardware specifications that were used: Processor: 3.70 GHz Intel® CoreTM i7-8700K CPU; 16 GB of main memory; graphics NVIDIA GeForce GTX 1080 Ti processor.

Table 7. Detection accuracy comparison under various attacks at different embedding rates

	Method	ker1	ker2	CNGL	ALE
0.3 bpp	LSBM	0.1522	0.0795	0.0995	0.2889
	LCG [38]	0.8986	0.6881	0.0899	1
	MKGM [17]	0.0867	0.0579	0.0890	0.2724
	2GA [40]	0.0860	0.0572	0.0846	0.2721
	MOLSBM	**0.0114**	**0.0095**	**0.0497**	**0.0098**
0.5 bpp	LSBM	0.2923	0.1443	0.1445	0.4099
	LCG [38]	0.9443	0.8228	0.2350	1
	MKGM [17]	0.1587	0.0748	0.1252	0.3055
	2GA [40]	0.2083	0.1220	0.0992	0.3111
	MOLSBM	**0.0711**	**0.0491**	**0.0543**	**0.0325**
0.8 bpp	LSBM	0.4841	0.2475	0.2724	0.5911
	LCG [38]	-	-	-	-
	MKGM [17]	0.3408	0.2166	0.2315	0.4485
	2GA [40]	0.2385	0.1727	**0.1285**	0.4003
	MOLSBM	**0.1576**	**0.1419**	0.2001	**0.0554**

5 Conclusion

In classical steganography algorithms, two primary issues must be addressed: selecting the pixels for data embedding and determining how these pixels should be modified. In the proposed method, after dividing the cover image into blocks, the data embedding process for each block is modeled as a bi-objective optimization problem and solved using the NSGA-II algorithm. To use a pseudo-random function for pixel selection, the best seed must be identified. The optimization method seeks to minimize the number of pixel changes (the first objective function) to identify the optimal seed. The embedding algorithm considers both increasing and decreasing the value of each pixel that needs modification. These modifications

Table 8. Results of universal steganalysis methods for different embedding levels

	0.05 bpp		0.1 bpp		0.2 bpp		0.25 bpp		0.3 bpp	
Features	**AUC**	**Test error**	**AUC**	**Test error**	**AUC**	**Test error**	**AUC**	**Test error**	**AUC**	**Test error**
SPAM	0.5378	0.4725	0.5729	0.4470	0.6549	0.4072	0.7021	0.3655	0.7603	0.3390
SRM	0.5678	0.4654	0.5909	0.4201	0.6638	0.3779	0.7135	0.3365	0.7663	0.3118
maxSRM	0.5689	0.4682	0.5981	0.4311	0.6709	0.3702	0.7117	0.3528	0.7435	0.3325
maxSRMd2	0.5432	0.4705	0.5893	0.4445	0.6682	0.3692	0.7100	0.3455	0.7421	0.3201

Table 9. Computation time

	0.3 bpp		0.5 bpp		0.8 bpp	
	Embedding	**Extraction**	**Embedding**	**Extraction**	**Embedding**	**Extraction**
Time	585 sec	5 sec	725 sec	6 sec	925 sec	8 sec

are made in such a way as to minimize histogram variations (the second objective function). After executing the NSGA-II algorithm, the target pixels for embedding are identified based on the optimal seed, and the pixel values of the stego block are computed using the best embedding pattern.

Results from various tests indicate that the proposed method, with a maximum embedding capacity of one bit per pixel, achieves lower detection accuracy against various attacks compared to LSB-based methods. Furthermore, the stego images produced by the proposed method exhibit significantly better quality than those generated by these methods. As future research directions, transitioning from LSBM to more advanced base methods to enhance embedding capacity, exploring alternative multiobjective optimization techniques for improved optimization outcomes, incorporating higher-order statistical features as objective functions, and increasing the number of objectives to guide the embedding process better are suggested.

Acknowledgment

The authors used AI-based tools (e.g., ChatGPT) solely for improving the English language and grammar in this manuscript. The scientific content and all analyses were conceived and written by the author without AI assistance.

References

[1] O. Torki, M. Ashouri-Talouki, and M. Mahdavi. Hierarchical deterministic wallets for secure steganography in blockchain. *ISC Int. J. Inf. Secur.*, 15(1):73–81, jan 2023.

[2] I. J. Kadhim, P. Premaratne, P. J. Vial, and B. Halloran. Comprehensive survey of image steganography: Techniques, evaluations, and trends in future research. *Neurocomputing*, 335:299–326, mar 2019.

[3] V. Sabeti and A. Aghabagheri. Developing an adaptive dct-based steganography method using a genetic algorithm. *Multimed. Tools Appl.*, 82(13):19323–19346, may 2023.

[4] B. Song, P. Wei, S. Wu, Y. Lin, and W. Zhou. A survey on deep-learning-based image steganography. *Expert Syst. Appl.*, 254:124390, nov 2024.

[5] P. N. Andono and D. R. I. M. Setiadi. Quantization selection based on characteristic of cover image for pvd steganography to optimize imperceptibility and capacity. *Multimed. Tools Appl.*, 82(3):3561–3580, jan 2023.

[6] P. Maniriho and T. Ahmad. Information hiding scheme for digital images using difference expansion and modulus function. *J. King Saud Univ. - Comput. Inf. Sci.*, 31(3):335–347, jul 2019.

[7] N. Cevik, T. Cevik, O. Osman, A. Gurhanli, S. Nematzadeh, and F. Sahin. Improved exploiting modification direction steganography for hexagonal image processing. *J. King Saud Univ. - Comput. Inf. Sci.*, 34(10):9273–9283, nov 2022. Part B.

[8] A. Westfeld and A. Pfitzmann. Attacks on steganographic systems. In A. Pfitzmann, editor, *Information Hiding*, volume 1768 of *Lecture Notes in Computer Science*, pages 61–76. Springer Berlin Heidelberg, Berlin, Heidelberg, 2000.

[9] J. Fridrich, M. Goljan, and R. Du. Detecting lsb steganography in color, and gray-scale images. *IEEE Multimed.*, 8(4):22–28, oct 2001.

[10] S. Dumitrescu, X. Wu, and Z. Wang. Detection of lsb steganography via sample pair analysis. In F. A. P. Petitcolas, editor, *Information Hiding*, volume 2578 of *Lecture Notes in Computer Science*, pages 355–372. Springer Berlin Heidelberg, Berlin, Heidelberg, 2003.

[11] A. D. Ker and R. Böhme. Revisiting weighted stego-image steganalysis. In E. J. Delp Iii, P. W. Wong, J. Dittmann, and N. D. Memon, editors, *Electronic Imaging 2008*, page 681905, San Jose, CA, feb 2008.

[12] S. N. M. Al-Faydi, S. K. Ahmed, and H. N. Y. Al-Talb. Improved lsb image steganography with high imperceptibility based on cover-stego matching. *IET Image Process.*, 17(7):2072–2082, may 2023.

[13] S. Hossain, S. Mukhopadhyay, B. Ray, S. K. Ghosal, and R. Sarkar. A secured image steganography method based on ballot transform and genetic algorithm. *Multimed. Tools Appl.*, 81(27):38429–38458, nov 2022.

[14] S. Rustad, D. R. I. M. Setiadi, A. Syukur, and P. N. Andono. Inverted lsb image steganography using adaptive pattern to improve imperceptibility. *J. King Saud Univ. - Comput. Inf. Sci.*, 34(6):3559–3568, jun 2022. Part B.

[15] G. Cancelli, G. Doerr, M. Barni, and I. J. Cox. A comparative study of ± steganalyzers. In *2008 IEEE 10th Workshop on Multimedia Signal Processing*, pages 791–796, oct 2008.

[16] M. Fateh, M. Rezvani, and Y. Irani. A new method of coding for steganography based on lsb matching revisited. *Secur. Commun. Netn.*, 2021(1):6610678, 2021.

[17] V. Sabeti, S. Faiazi, and H. Shirinkhah. Improving security of lsbm steganography using of genetic algorithm, multi-key and blocking. Available: https://www.sid.ir/paper/228423/en, 2020.

[18] C. Wang, X. Li, B. Yang, X. Lu, and C. Liu. A content-adaptive approach for reducing embedding impact in steganography. In *2010 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 1762–1765, mar 2010.

[19] K. Loukhaoukha, J. Chouinard, and M. H. Taieb. Optimal image watermarking algorithm based on lwt-svd via multi-objective ant colony optimization. Available: https://www.semanticscholar.org/paper/Optimal-Image-Watermarking-Algorithm-Based-on-via-Loukhaoukha-Chouinard/ddd620a5b9a78f5926924451a367134208e193f2, 2011. Accessed: Aug. 04, 2024.

[20] K. Loukhaoukha. Image watermarking algorithm based on multiobjective ant colony optimization and singular value decomposition in wavelet domain. *J. Optim.*, 2013:1–10, 2013.

[21] M. Mubeen, S. A. M. Gilani, and K. Zafar. Robust image watermarking in contourlet domain using multi objective genetic algorithm. In *Eighth International Conference on Digital Information Management (ICDIM 2013)*, pages 67–72, sep 2013.

[22] J.-S. Lee, J.-W. Wang, and K.-Y. Giang. A new image watermarking scheme using multi-objective bees algorithm. *Appl. Math. Inf. Sci.*, 8(6):2945–2953, nov 2014.

[23] M. Suresh and I. Shatheesh Sam. Optimized interesting region identification for video steganography using fractional grey wolf optimization along with multi-objective cost function. *J. King Saud Univ. - Comput. Inf. Sci.*, 34(6):3489–3496, jun 2022.

[24] M. H. Noor Azam, F. H. Mohd Ridzuan, and M. N. S. Mohd Sayuti. Optimized cover selection for audio steganography using multi-objective evolutionary algorithm. *J. Inf. Commun. Technol.*, 22, 2023.

[25] Z. Yin, Y. Ji, and B. Luo. Reversible data hiding in jpeg images with multi-objective optimization. *IEEE Trans. Circuits Syst. Video Technol.*, 30(8):2343–2352, aug 2020.

[26] M. Kaur, V. Kumar, and D. Singh. An efficient image steganography method using multi-objective differential evolution. In *Digital Media Steganography*, pages 65–79. Elsevier, 2020.

[27] N. Subramanian, O. Elharrouss, S. Al-Maadeed, and A. Bouridane. Image steganography: A review of the recent advances. *IEEE Access*, 9:23409–23423, 2021.

[28] A. G. Devi, A. Thota, G. Nithya, S. Majji, A. Gopatoti, and L. Dhavamani. Advancement of digital image steganography using deep convolutional neural networks. In *2022 International Interdisciplinary Humanitarian Conference for Sustainability (IIHC)*, pages 250–254, nov 2022.

[29] A. Martín, A. Hernández, M. Alazab, J. Jung, and D. Camacho. Evolving generative adversarial networks to improve image steganography. *Expert Syst. Appl.*, 222:119841, jul 2023.

[30] W. Luo, W. Huang, R. Fan, R. Shi, and Y. Q. Shi. A comprehensive survey of digital image steganography and steganalysis. *APSIPA Trans. Signal Inf. Process.*, 13(1), nov 2024.

[31] H.-W. Tseng and H.-S. Leng. A reversible modified least significant bit (lsb) matching revisited method. *Signal Process. Image Commun.*, 101:116556, feb 2022.

[32] C. Kim, L. Cavazos Quero, K.-H. Jung, and L. Leng. Advanced dual reversible data hiding: A focus on modification direction and enhanced least significant bit (lsb) approaches. *Appl. Sci.*, 14(6):2437, mar 2024.

[33] T. D. Nguyen and H. Q. Le. A novel secure image steganography scheme based on hamming encoding and lsb matching revisited using lah transform. *Multimed. Tools Appl.*, feb 2024.

[34] W. Wu and H. Li. A novel scheme for random se-

quential high-capacity data hiding based on pvd and lsb. *Signal Image Video Process.*, 18(3):2277–2287, apr 2024.

[35] V. Sabeti. Unmistakable information embedding into the integer wavelet transform domain of an image using an xor function and a genetic algorithm. *Multimed. Tools Appl.*, 83(8):23655–23688, aug 2023.

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

[37] P. D. Shah and R. Bichkar. Genetic algorithm-based imperceptible image steganography technique with histogram distortion minimization. In V. E. Balas, A. E. Hassanien, S. Chakrabarti, and L. Mandal, editors, *Lecture Notes on Data Engineering and Communications Technologies*, volume 62, pages 267–278. Springer Singapore, Singapore, 2021.

[38] P. D. Shah and R. S. Bichkar. A secure spatial domain image steganography using genetic algorithm and linear congruential generator. In *International Conference on Intelligent Computing and Applications*, pages 119–129, 2018.

[39] S. Khan. Ant colony optimization (aco) based data hiding in image complex region. *Int. J. Electr. Comput. Eng. IJECE*, 8(1):379, feb 2018.

[40] V. Sabeti and S. Faiazi. Secure image steganography with high visual image quality based on lsbm and genetic algorithm. *J. Soft Comput. Inf. Technol.*, 9(4):70–82, 2020.

[41] R.-Z. Wang, C.-F. Lin, and J.-C. Lin. Image hiding by optimal lsb substitution and genetic algorithm. *Pattern Recognit.*, 34(3):671–683, mar 2001.

[42] H. R. Kanan and B. Nazeri. A novel image steganography scheme with high embedding capacity and tunable visual image quality based on a genetic algorithm. *Expert Syst. Appl.*, 41(14):6123–6130, oct 2014.

[43] R. Roy and S. Laha. Optimization of stego image retaining secret information using genetic algorithm with 8-connected psnr. *Procedia Comput. Sci.*, 60:468–477, jan 2015.

[44] A. Khamrui, D. D. Gupta, S. Ghosh, and S. Nandy. A spatial domain image authentication technique using genetic algorithm. In J. K. Mandal, P. Dutta, and S. Mukhopadhyay, editors, *Computational Intelligence, Communications, and Business Analytics*, pages 577–584. Springer, Singapore, 2017.

[45] S. M. Douiri and S. Elbernoussi. An ant colony optimisation for data hiding in greyscale images. *Int. J. Oper. Res.*, 29(1):101, 2017.

[46] A. Banharnsakun. Artificial bee colony approach for enhancing lsb based image steganography. *Multimed. Tools Appl.*, 77(20):27491–27504, oct 2018.

[47] A. H. Mohsin, A. I. Al-Maqoshi, R. H. Al-Samarraie, L. A. Jawad, A. M. Ibrahim, and M. A. Al-Maamari. New method of image steganography based on particle swarm optimization algorithm in spatial domain for high embedding capacity.

[48] M. Boryczka and G. Kazana. Hiding information in digital images using ant algorithms. *Entropy*, 25(7):963, jun 2023.

[49] M. M. Fadel, W. Said, E. A. A. Hagras, and R. Arnous. A fast and low distortion image steganography framework based on nature-inspired optimizers. *IEEE Access*, 11:125768–125789, 2023.

[50] M. A. Hameed, O. A. Abdel-Aleem, and M. Hassaballah. A secure data hiding approach based on least-significant-bit and nature-inspired optimization techniques. *J. Ambient Intell. Humaniz. Comput.*, 14(5):4639–4657, may 2023.

[51] V. Sabeti and S. Faiazi. MOLSBM: A multiobjective lsb matching steganography method. In *2024 11th International Symposium on Telecommunications (IST)*, pages 738–743, oct 2024.

[52] A. D. Ker. Steganalysis of lsb matching in grayscale images. *IEEE Signal Process. Lett.*, 12(6):441–444, 2005.

[53] G. Cancelli, I. J. Cox, and G. Doërr. Improved lsb matching steganalysis based on the amplitude of local extrema. In *IEEE International Conference on Image Processing*, 2008.

[54] M. Ferrara, S. Rasouli, M. Khademi, and M. Salimi. A robust optimization model for a decision-making problem: An application for stock market. *Oper. Res. Perspect.*, 4:136–141, 2017.

[55] K. Deb. An investigation of niche and species formation in genetic function optimization. In *ICGA'89*, pages 42–50, 1989.

[56] R. Huang, C. Lian, Z. Dai, Z. Li, and Z. Ma. A novel hybrid image synthesis-mapping framework for steganography without embedding. *IEEE Access*, 11:113176–113188, 2023.

[57] Z. Zhou, Q. Wang, X. Du, X. Liang, C. Sun, Y. Huang, M. Yu, C. Liu, and L. Hu. Latent vector optimization-based generative image steganography for consumer electronic applications. *IEEE Trans. Consum. Electron.*, 70(1):4357–4366, feb 2024.

[58] M. Qasaimeh, A. A. Qtaish, and S. Aljawarneh. Robust steganographic approach using generative adversarial network and compressive autoencoder. *Multimed. Tools Appl.*, nov 2024.

[59] F. Chen, Q. Xing, and F. Liu. Technology of hiding and protecting the secret image based on two-channel deep hiding network. *IEEE Access*, 8:21966–21979, 2020.

[60] X. Duan, J. Song, R. Yan, H. Wang, and Z. Zhang. High-capacity image steganography based on improved fc-densenet. *IEEE Access*, 8:170174–170182, 2020.

[61] S. Baluja. Hiding images in plain sight: deep steganography. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, pages 2066–2076, Red Hook, NY, USA, dec 2017. Curran Associates Inc.

[62] A. Khalifa and A. Guzman. Imperceptible image steganography using symmetry-adapted deep learning techniques. *Symmetry*, 14(7):1325, jul 2022.

[63] Q. Li, S. Cao, Y. Zhao, S. Wang, Q. Guo, and Q. Fu. A novel grayscale image steganography scheme based on chaos encryption and generative adversarial networks. *IEEE Access*, 8:168166–168176, 2020.

[64] L. Liu, L. Meng, Y. Peng, and X. Wang. A data hiding scheme based on u-net and wavelet transform. *Knowl.-Based Syst.*, 223:107022, jul 2021.

[65] Z. Gan and Y. Zhong. A novel grayscale image steganography via generative adversarial network. In *Web Information Systems and Applications*, pages 405–417. Springer, Cham, 2021.

[66] F. Huang, B. Li, and J. Huang. Attack lsb matching steganography by counting alteration rate of the number of neighbourhood gray levels. In *2007 IEEE International Conference on Image Processing*, pages I–401–I–404, sep 2007.

[67] T. Denemark, V. Sedighi, V. Holub, R. Cogranne, and J. Fridrich. Selection-channel-aware rich model for steganalysis of digital images. In *2014 IEEE International Workshop on Information Forensics and Security (WIFS)*, pages 48–53, dec 2014.

Vajiheh Sabeti is an Assistant Professor in the Department of Engineering and Technology at Alzahra University. She received her B.Sc. degree in Software Engineering in 2004, her M.Sc. degree in Computer Architecture in 2007, and her Ph.D. degree in Computer Engineering in 2012 from Isfahan University of Technology (IUT), Isfahan, Iran. Her research interests are Soft Computing, Image Processing and Information Hiding.