Imperceptible Steganography Scheme with High Payload Capacity using Genetic Algorithm and Particle Swarm Optimization

Pratik D. Shah, Rajankumar S. Bichkar

Abstract*: Security is the most significant parameter in all type of confidential data transfers. Steganography is used to enhance the security of such confidential communications. Steganography is a method of covert communication in which the existence of secrecy is concealed. In image steganography, achieving high data embedding capacity and simultaneously retaining good visual quality is a very tricky and difficult objective. In this paper, a reversible, secure, extremely imperceptible and high payload capacity steganography technique in the spatial domain is proposed. The proposed method employs evolutionary computation techniques to identify the most optimum locations and arrangements for secret data embedding. The proposed technique uses Particle Swarm Optimization to find the best possible order of data hiding whereas Genetic algorithm is used to identify the best possible arrangements to modify secret data to produce least amount of change in cover-image. The result of the proposed scheme is compared with many steganography techniques and the proposed scheme outperforms the existing schemes in terms of imperceptibility. The proposed technique produces an average PSNR value of 46.40 dB at 2 bit per pixel data embedding rate.

***Keywords*: Data Hiding, Genetic Algorithm (GA), Image Steganography, Particle Swarm Optimization (PSO)**

I. INTRODUCTION

Steganography is a technique used for concealed communication. In steganography, the presence of covert communication is camouflaged by using dummy cover media [1]. The secret message is hidden in cover media without causing noticeable visual artifacts. In image steganography, the cover media is called as cover- image and the final image obtained after hiding secret information is known as stego-image. The core objective of steganography is to decrease the dissimilarity between stego-image and cover-image. Imperceptibility, payload capacity and security are the three factors used to assess the efficiency of image steganography [2].

In last couple of decades, a vast amount of work has been carried out on steganography. Most of this work is focused on improving imperceptibility. Few studies have focused on imperceptibility and data hiding capacity simultaneously, but most of them have low imperceptibility as data hiding capacity is increased. Since satisfying both imperceptibility and data hiding capacity is a difficult task many researchers have used evolutionary computation approaches to solve this dicey issue. Wang et al. [3] proposed an imperceptible steganography scheme with low data hiding capacity. This technique hides data in firstLSB and uses the second LSB to modify statistical parameters; as a result, the security of the scheme is improved against RS steganalysis. The resultant stego-image produced has a PSNR value of 41.2 dB, which is quite low as compared to LSB replacement steganography at one bit per pixel (bpp) data embedding capacity. Bedi et al. [4] proposed a PSO based image steganography scheme. In theirscheme, PSO is used to identify best positions in the cover-image to embed the secret message. Enhancingimperceptibility and data embedding capacitywas thecore emphasis of this scheme. Kanan and Nazeri [5] presented a high payload spatial domain image steganography method using GA. Their focus was on identifying the best positions and arrangements to implanted secret data. They used GAto discover several possibilities of hiding secret information in cover-image. These possibilities comprise of exploring numerous initial points, exploring various embedding orders, exploring new pixel arrangements, etc. Maheswari and Hemanth [6] presented a transform domain steganography method using GA and PSO to improve imperceptibility.Evolutionary computation techniques are utilized to identifythe best coefficients in the transform domain to hide the secret message.

In this paper, an imperceptible and high data hiding capacity steganography technique in the spatial domain is proposed. The proposed technique allows data embedding at 2 bit per pixel which results in 25% data hiding capacity. The proposed scheme is inspired by steganography method developed by Kanan and Nazeri [5]. The proposed technique uses PSO and GA to improve the imperceptibility of stego-image by reducing the difference between cover and stego image. The secret data is hidden in two LSB's of all pixels in cover-image. To hide the secret message in LSB's of cover image, simple LSB replacement technique is used. PSO is employed for searching the sequence in which binary secret data array is embedded. After PSO finds the best possible sequence for data embedding, GA is used to modify the binary secret data array. The secret data array is modified based on two different factors viz. data polarity and data direction. After insertingthe modified secret information in cover-image,optimal pixel

Revised Manuscript Received on October 15, 2019
***** Correspondence Author
Pratik D. Shah*,Research Scholor Dept. of E&TC, G. H. Raisoni College of Engineering and Management, Assistant Professor Dr. D. Y. Patil School of Engineering, Savitribai Phule Pune University, Pune 412207, India. Email: shahpratik219@gmail.com
Rajankumar S. Bichkar,Principal, Vidya Pratishthan's Kamalnayan Bajaj Institute of Engineering and Technology, Baramati, Savitribai Phule Pune University, Pune, India. Email: bichkar@yahoo.com

adjustment process (OPAP) is applied to enhance the quality of stego-image.

The key contribution of this paper is the use of PSO for finding the best sequence for data embedding instead of using traditional sequential method.For a grey scale image of size 512×512, there are 262144! possibilities of sequences, which can be used for data hiding. 262144! is a huge number and it is impossible to explore each and every possible sequence, even with the today's fastest supercomputers. Hence we employ PSO to find the near optimal solution.

The remaining paper is organized as follows: Section 2 describes the proposed technique along with other important preliminary concepts like PSO, GA, LCG and OPAP. Experimental setup, results and discussions are presented in section 3. In section 4, the conclusions inferred from the research are presented.

II. PROPOSED TECHNIQUE

The proposed technique is comprised of two parts. First part consists of employing PSO to determine the sequence used for embedding secret data. Second part consists of manipulating secret data to increase the similarity between LSB of cover-image and secret data. To explain the working of proposed scheme, the understanding of few preliminary concepts is necessary. These concepts include linear congruential generator (LCG), PSO, effect of data polarity and data direction, GA and OPAP. These important concepts are presented in the following sub-sections along with the process of embedding and extracting the secret data.

A. Linear Congruential Generator

A pseudo-random sequence of length *M* can be generated by using LCG with the help of (1). To produce a sequence of *M* numbers, LCG needs a seed value, a multiplying factor and an offset value [7].

$$X_{n+1}=(a \times X_n + c)\ mod\ M \qquad (1)$$

In (1),$X_{n+1}$is the subsequent number in the sequence, X_n is the currentvalue, a is the constant multiplying factor, c is an offset value and M is the sequence length. In initial state $X_n=X_0$, where X_0 is the seed value. LCG doesn't always produce non-repetitive sequences hence we use modified LCG algorithm presented in [8].

B. Particle Swarm Optimization

Particle swarm optimization is a clever, meteheuristic optimization algorithm derivedfrom paradigm of swarm intelligence. PSO is motivated by communal behavior of fish and birds [9]. It works on two basic principles i.e. communication and learning. PSO consists of population of candidate solutions called swarm and each candidate in the population is called as a particle. Each particle in the swarm has a position vector $x_i(t) \in X$, where $X$is the multidimensional search space, $i$is the index of particle and t represents the iteration number. Along with position vector, each particle also has a velocity vector $v_i(t)$, this velocity describes the movement of the particle i in terms of direction and distance i.e. step size. In addition to position and velocity vector, each particle also has a memory associated with it which stores the best position of the particle; it is denoted by $p_i(t)$.Similar to $p_i(t)$,PSO also has a global best $g(t)$, which is the common best position among the members of the swarm. The position of each particle is updated to $x_i(t+1)$in subsequent iteration by adding it to $v_i(t+1)$i.e.

$$x_i(t+1) = x_i(t) + v_i(t+1) \qquad (2)$$

To obtain the position vector of next iteration by using (2), we need the velocity vector $v_i(t+1)$which is given by (3).

$$v_i(t+1) = w \cdot v_i(t) + r_1 \cdot c_1 \times \big(p_i(t) - x_i(t)\big) + r_2 \cdot c_2 \times (g(t) - x_i(t)) \qquad (3)$$

In (3), r_1 and r_2 are random numbers, c_1 and c_2 are acceleration coefficients. The term $w \cdot v_i(t)$ representsinertia, $r_1 \cdot c_1 \times \big(p_i(t) - x_i(t)\big)$ is the cognitive component and $r_2 \cdot c_2 \times (g(t) - x_i(t))$ is the social component. The initial population is generated randomly so that they are scattered over the entire multidimensional search space. All particles flow throughout the entire search space to find the best solution to the optimization problem. Each solution in the swarm is judged by its fitness which is calculated by fitness function. The position of all particle is updated in each iteration based on its previous experimental knowledge and socially swapped information amongst the particles.

For the proposed technique, each particle is a three dimensional entity, which is comprised of the LCG parameters as shown in figure 1. The parameters used for PSO are shown in Table 1. The objective function used to evaluate the performance of the particle is PSNR value of stego-image.

X_o	a	C

Fig. 1.Particle structure for PSO

Table- I: PSO parameters

Parameters	Value
Inertia weight (w)	1
Damping factor (w_{damp})	0.99
c_1 and c_2 (acceleration coefficients)	2
r_1 and r_2	0 to 1
Iterations	500

C. Data Direction and Data Polarity

To embed the secret data in cover image, it is first converted into two binary data arrays named Sm_1 and Sm_2. To insert data in cover image pixels, one bit is selected from each binary data array Sm_1 and Sm_2. With the help of PSO and LCG the best sequence for hiding the secret data is found, but still there are various possibilities to hide the two bits of secret data in LSB of cover image. These possibilities include swapping the two bits of secret data or modifying the secret data before embedding. Data direction (dd) parameter controls the way secret data is embedded in LSB of cover-image. Let us assume binary value '10' as the two bits of secret data to be hidden in LSB of cover-imagepixel.Let the value of cover image pixel be 97 (01100001), the two

possible ways to embed the data are depicted in figure 2.

Data Direction(dd)	**0 - (Normal)**	**1 - (Swapping)**
Secret data	1 0	1 0
Cover-image pixel	0 1 1 0 0 0 **0 1**	0 1 1 0 0 0 **0 1**
Stego-image pixel	0 1 1 0 0 0 **1 0**	0 1 1 0 0 0 **0 1**

Fig. 2.Effect of data direction (dd) parameter on the process of secret data embedding

The decision of embedding bits of $Sm_1$and Sm_2 either in 7th or 8th bit of cover image pixel is taken by data direction parameter. As shown in figure 2 if data direction $dd = 0$, we use traditional approach of data embedding. In this case both the LSB of cover image are changed, whereas if $dd = 1$ we swap the order of data embedding, resulting in no change in the pixels value of cover-image. Hence it's an important parameter and needs to be considered while hiding data.

Another important parameter which can significantly impact the data hiding process is data polarity. In this parameter the secret data is modified before inserting it in LSB of cover image pixel. The data polarity parameter (dp) modifies the secret data by complementing its bits. Since we are inserting two bits of secret data in each pixel of cover image, the data polarity parameter decides which bits to complement. The parameter dp is of two bits hence it has four possible values i.e. 00, 01, 10 and 11. The first bit of dpcorresponds to the modifications to be performed on secret data inserted in 7th bit of cover image while second bit corresponds to the modifications to be performed on secret data inserted in 8th bit. If the value of $dp$is 0 the secret data bit is complemented else if the value of $dp$is 1 it is kept unchanged. The effect of dp is illustrated in figure 3 with an example. Let us assume the value of secret data bits as '10' and the value of cover image pixel as 128 i.e. 1000 0000. The effect of all possible values of dpon data embedding process can be seen in figure 3.

Secret data	10			
Data Polarity *dp*	00	01	10	11
Modified data to be embedded	01	00	11	10
Cover image pixel	10000**00**	10000**00**	10000**00**	10000**00**
Stego-image pixel	10000**01**	10000**00**	10000**11**	10000**10**

Fig. 3.Effect of data polarity (dp) parameter on the process of secret data embedding

The importance of dpcan be easily seen from the example illustrated in figure 3, if we use traditional LSB replacement method for data embedding the cover image value of 128 would have been replaced by 130 in stego-image. However instead of traditional LSB replacement technique if we modify secret data before embedding with the help of dpwe get four different possibilities of data embedding and one of these four possibilities will definitely produce less error. In the above example if we select $dp = 01$, then there is no change between stego-image pixel and cover-image pixel.

The combined effect of $dd$and dp,on the process of data hiding is illustrated in Table 2. Let S_7 and S_8 be 7th and 8th LSB of stego-image pixel and let Sm_1andSm_2 be the secret data is to be hidden in it. All possible combinations and effective data embedding arrangements are shown in Table 2.

Table-II: Combined effect of (*dd*) and (*dp*) on the process of secret data embedding

dd **(Data direction)**	***Dp*** **(Data Polarity)**	**Effect on data insertion**	
		S_7	S_8
0	00	$\overline{Sm_1}$	$\overline{Sm_2}$
0	01	$\overline{Sm_1}$	Sm_2
0	10	Sm_1	$\overline{Sm_2}$
0	11	Sm_1	Sm_2
1	00	$\overline{Sm_2}$	$\overline{Sm_1}$
1	01	$\overline{Sm_2}$	Sm_1
1	10	Sm_2	$\overline{Sm_1}$
1	11	Sm_2	Sm_1

D. Genetic Algorithm

GA is motivated from Darwin's theory of evolution and follows the principle, "survival of the fittest". In GA, an initial population is randomly generated and each individual in the population is a probable solution to the problem being solved.All the probable solutions will contest with each other for their existence and seeding next generation. Every individual in the population will be assessed depending on its fitness. Various GA operators such as crossover, reproduction, mutation, etc. are utilized to get next generation of solutions [10]. This process is iterated till predefined number of iteration is reached or the optimum result is obtained or there is no improvement in the fitness of the population over a prescribed amount of iterations.

In the proposed steganography scheme, the secret information is modified before inserting it in cover-image pixels using the eight combinations presented in the Table 2. GA is employed to decide which combination should be used for a particular set of cover image pixels. Let the size of cover image be $M \times N$. The GA chromosome consists of eight genes; each gene represents a value from 0 to $M \times N - 1$. The sequence used for embedding is generated by modified LCG algorithm and is stored in an array. The values of GA chromosome are representing the index of array in which sequence of data embedding is stored. The values in chromosome will be arranged in ascending order and all the combination of dd and dp will be used in the intervals prescribed by the chromosome. For the sake of convenience the pixels in cover image are numbered from left to right direction and from first pixel in the top till the last pixel in the bottom. Figure 4 displays the genetic algorithm chromosome structure; figure 5 depicts an illustrative example of chromosome with random values and Table 3 illustrates how the chromosome values are used to decide the combination of dd and dp.

G1	G2	G3	G4	G5	G6	G7	G8

Fig. 4.Genetic algorithm chromosome structure

75	1256	25966	65842	99125	109475	178536	237512

Fig. 5.Illustrative example of GA chromosome structure

Table-III: Interpretation of GA chromosome values

Chromosome value (sequence array index)		*dd* (Data direction)	*dp* (Data Polarity)
from	to		
75	1256	0	00
1256	25966	0	01
25966	65842	0	10
65842	99125	0	11
99125	109475	1	00
109475	178536	1	01
178536	237512	1	10
237512	75	1	11

To implement genetic algorithm for the proposed technique the population size is set to 50. The initial population is randomly generated. Each individual in the population competes with each other for its survival and to seed the next generation. To evaluate fitness of each individual PSNR value of stego-image is used as the fitness function.To obtain population of next generation tournament selection is used, in which five chromosomesare arbitrarily selected and out of it two chromosomes with highest fitness are used to seed next generation. Elitism is used to send the best solution out of five selected chromosomes in the next generation. Crossover is used to obtain one new chromosome by combining the selected two individuals. Mutation is performed on these selected chromosomes to generate three new probable solutions. These GA operators are used until all the individuals in the population are not screened. This process is repeatedtill five hundred iterations. Table 4 shows the parameters used in genetic algorithm.

Table- IV: Genetic algorithm parameters

Parameters	Value
Total Iterations	500
Population Size	50
Total genes in chromosome	8
Selection probability	0.2
Crossover probability	0.2
Mutation probability	0.6
Fitness function	PSNR

E. Optimal Pixel Adjustment Process

Chan and Cheng [11] proposed the concept of OPAP to enrich the quality of stego-image after LSB steganography. The fundamental concept of OPAP is as follows:

Let x_i, $x_i'$and x_i''be the corresponding pixel values of the i^{th} pixel in the cover-image I, the stego-image I' obtained by proposed technique,I'' the reformed stego-image achieved after the optimal pixel adjustment process.

Let $\Delta_i = x_i' - x_i$ be the embedding error betweenx_i and x_i' obtained after inserting n bits of secret data in each pixel of cover-image.

Thus-$2^n < \Delta_i < 2^n$

The embedding error Δ_i is sub-divided in three intervals:

Interval 1: $2^{n-1} < \Delta_i < 2^n$,

Interval 2: $-2^{n-1} \le \Delta_i \le 2^{n-1}$,

Interval 3:$-2^n < \Delta_i < -2^{n-1}$

OPAP modifies x_i' to x_i'' based on the above three intervals.

Case 1: ($2^{n-1} < \Delta_i < 2^n$):If $x_i' \ge 2^n$, then $x_i'' = x_i' - 2^n$; otherwise $x_i'' = x_i'$

Case 2: ($-2^{n-1} \le \Delta_i \le 2^{n-1}$):$x_i'' = x_i'$

Case 3: ($-2^n < \Delta_i < -2^{n-1}$):If $x_i' < 256 - 2^n$, then $x_i'' = x_i' + 2^n$; otherwise $x_i'' = x_i'$

The OPAP operates by modifying the bits other than those affected by LSB steganography. In the proposedtechnique, two bits are embedded in each pixel hence OPAP operates by modifying the third bit of stego-image. The OPAP only changes those pixels in stego-image where the embedding error Δ_i is more than 2^{n-1} or smaller than -2^{n-1}. By changing the third LSB OPAP reduces the error Δ_i and subsequently improves the PSNR value of modified stego-image.

F. Secret Data Embedding Procedure

In the proposed technique we use traditional LSB replacement steganography to insert secret data. The proposed technique works on grey scale images of size 512×512 and 256×256 for cover-image and secret data image respectively. Initially the secret data image is converted into two bit arrays. The first array contains four MSB of secret data image pixels whereas the second array consists of four LSB. As we embed two bits of secret data in every pixel of cover image, one bit is selected from each array. The sequence of inserting secret data bits in cover-image is not sequential; it is defined through the sequence generated by LCG. The sequence generation is a tricky and important part hence PSO is used to find the parameters of sequence generator resulting in near-optimum sequence for data insertion. The LCG provides only the sequence of data embedding but out of two arrays which bit is to be inserted in which LSB is yet to be decided and also the polarity of secret data is to be modified before data insertion. These decisions are dependent on the chromosome used for genetic algorithm. This process is iterated continuously to obtain an improvement in results. The algorithm for secret message insertion is as follows:

Input: Cover-image $C = \{c_1, c_2, \ldots, c_{m\times n}\}$,
Secret data image $M = \{m_1, m_2, \ldots, m_{(m/2)\times(n/2)}\}$

Output: Stego-image $S = \{s_1, s_2, \ldots, s_{m\times n}\}$

1. Initialize two, 1-D bit arrays $sm_1$and $sm_2$of length $len=m\times n$ as,
   $sm_1 \leftarrow (M)_{8\text{-}5} \forall m_i$
   $sm_2 \leftarrow (M)_{4\text{-}1} \forall m_i$
2. Initialize
   $p \leftarrow$ population size
   $iteration \leftarrow$number of iterations
3. Randomly generate p chromosomes (ch) and particles (pt) for GA and PSO respectively.
4. Produce p sequences (seq)of length len using LCG
5. For every chromosome in the population initialize
   $S \leftarrow C$
   $cnt \leftarrow 1$
   for i=1 to m
   for j=1 to n
   if $seq(cnt) >= ch_1 \&\& seq(cnt) < ch_2$
   $S_7(i,j) = \sim sm_1(seq(cnt))$
   $S_8(i,j) = \sim sm_2(seq(cnt))$
   elseif $seq(cnt) >= ch_2 \&\& seq(cnt) < ch_3$
   $S_7(i,j) = \sim sm_1(seq(cnt))$
   $S_8(i,j) = sm_2(seq(cnt))$
   elseif $seq(cnt) >= ch_3 \&\& seq(cnt) < ch_4$
   $S_7(i,j) = sm_1(seq(cnt))$
   $S_8(i,j) = \sim sm_2(seq(cnt))$
   elseif
   $seq(cnt) >= ch_4 \&\&$
   $seq(cnt) < ch_5$

$S_7(i, j) = sm_1(seq(cnt))$
$S_8(i, j) = sm_2(seq(cnt))$
elseif $seq(cnt) >= ch_5 \&\& seq(cnt) < ch_6$
$S_7(i, j) = \sim sm_2(seq(cnt))$
$S_8(i, j) = \sim sm_1(seq(cnt))$
elseif $seq(cnt) >= ch_6 \&\& seq(cnt) < ch_7$
$S_7(i, j) = \sim sm_2(seq(cnt))$
$S_8(i, j) = sm_1(seq(cnt))$
elseif $seq(cnt) >= ch_7 \&\& seq(cnt) < ch_8$
$S_7(i, j) = sm_2(seq(cnt))$
$S_8(i, j) = \sim sm_1(seq(cnt))$
else
$S_7(i, j) = sm_2(seq(cnt))$
$S_8(i, j) = sm_1(seq(cnt))$
$cnt = cnt + 1;$

6. Use OPAP on the received stego-image
7. Evaluate each individual using fitness function i.e. PSNR.
8. Randomly choose five individuals from population.
9. Use tournament selection to select the best in the batch and use GA operators like crossover, reproduction and mutation to generate new probable solutions.
10. Repeat step 8 and 9 till all the individuals from the populations have been evaluated.
11. Use PSO to produce new solutions.
12. Repeat the entire process from step 4 to11 for all iterations.
13. Conceal the secret key in predefined pixel locations

G. Secret Data Extraction Process

To retrieve the concealed information from the stego-image the exact reverse process needs to be performed. The first step in getting secret message starts with obtaining the secret key from the predefined positions of stego-image. Next step is to get the PSO particle and GA chromosome from the secret key and obtain the sequence used for hiding secret data. All the LSB's from each pixel of stego-image are obtained and arranged in two bit arrays as per the sequence generated. Later, the GA chromosome is used to modify the two bit arrays as per chromosome value and its interpretation as shown in Table 3. Secret message is then constructed back by using these two bit arrays. The algorithm to extract secret message is as follows:

Input: Stego-image $S = \{s_1, s_2, \ldots, s_{m \times n}\}$
Output: Secret data $M = \{m_1, m_2, \ldots, m_{(m/2)\times(n/2)}\}$

1. Obtain the key from predefined positions of stego-image.
2. Extract PSO particle and GA chromosome from the key.
3. Obtain the sequence used for data insertion using PSO particle.
4. Obtain both the LSB's from all pixels of stego-image and arrange them in two bit arrays se_1 and se_2as per the sequence
5. Modify the bit arrays as per the values of GA chromosome with reference to its implication in Table 3
6. Recreate secret message by usingse_1 and se_2

III. RESULTS AND DISCUSSIONS

The proposed scheme was implemented on Matlab 8.1 running on intel core i3 processor with a RAM of 4 GB. The experimentation was performed on standard grey scale test images depicted in figure 6.We have compared the results of proposed technique with various other techniques for same test images. To measure the imperceptibility we have used PSNR parameter. The PSNR is the ratio of maximum power of a signal topower of its corrupting noise.To calculate PSNR, first MSE needs to be calculated, MSE is calculated by (4) and PSNR is calculated by (5).In (4), Y and Z denote the rows and columns in the image respectively. A_{ij} and B_{ij} are pixel values of ij^{th}location of cover-image and stego-image respectively.

$$MSE = \frac{1}{YZ}\sum_{i=1}^{Y}\sum_{j=1}^{Z}(A_{ij} - B_{ij})^2 \quad (4)$$

$$PSNR = 10 \cdot log_{10}\frac{(255)^2}{MSE} \quad (5)$$

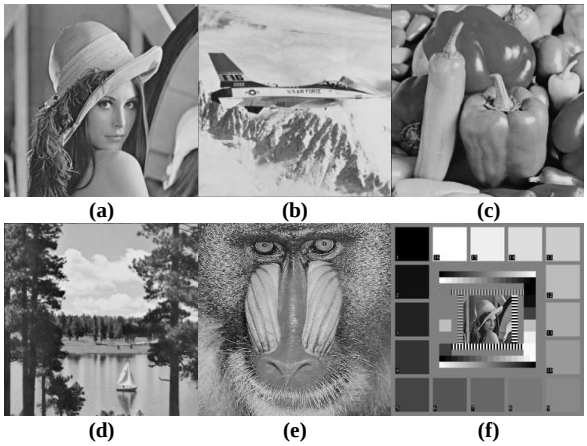
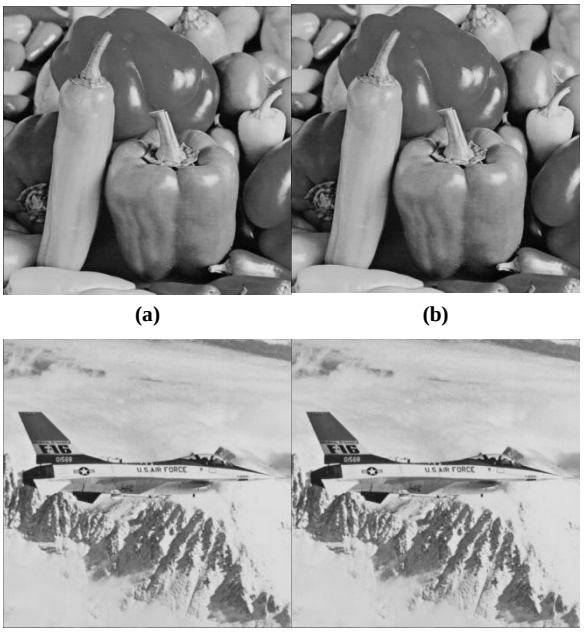


Fig. 6. Test images (a) - (e) Cover-images. (f) - Secret data image (Test Pattern)

In Table 5 PSNR values of proposed technique and many other popular steganography schemes is compared. The results exhibit the superiority of proposed technique in terms of imperceptibility. For any steganography technique more the PSNR value better the visual quality of stego-image and consequently lower the risk of its detection through visual analysis. In figure 7, the stego-image output of proposed technique is compared with original images.

Table- V: PSNR values of various stego-images

Sr. No.	Cover image	Chang et al. [5]	Yang et al. [5]	Wu et al. [5]	Kanan et al. [5]	Proposed technique
1	Lena	40.37	41.60	43.54	45.12	46.40
2	Jet	40.73	41.66	43.53	45.18	46.42
3	Pepper	39.30	41.56	43.56	45.13	46.41
4	Sailboat	38.86	41.51	43.55	45.10	46.40
5	Baboon	39.94	41.55	43.54	45.12	46.42



(a) (b)

(a) (b)

Fig. 7.(a) Cover-images Pepper and Jet, (b) Stego-images

The results of proposed technique i.e. PSNR values shown in Table 5 are average values, the experiment was performed ten times and the mean value of PSNR is presented in the result. The main reason of executing the program ten times was to analyze the performance of evolutionary computation techniques, to make sure the evolutionary computation techniques produce good results consistently. The detailed statistical analysis of all ten explorations in terms of mean, variance and standard deviation is shown in Table 6. It is observed that for all images the value of standard deviation is very small, so we can conclude that the performance of proposed technique is consistent. Figure 8 shows the plot of PSNR value of stego-image for all the iterations of the proposed technique, the test image considered was Jet and total 500 iterations were executed for GA and PSO.

Table-VI: Statistical parameters of PSNR value of stego-image

Sr. No.	Image	Statistical parameters		
		Mean	Variance	Standard deviation
1	Lena	46.4027	0.000135	0.0116
2	Jet	46.4211	0.000042	0.0065
3	Pepper	46.4124	0.000193	0.0138
4	Sailboat	46.4010	0.000145	0.0120
5	Baboon	46.4189	0.000021	0.0045

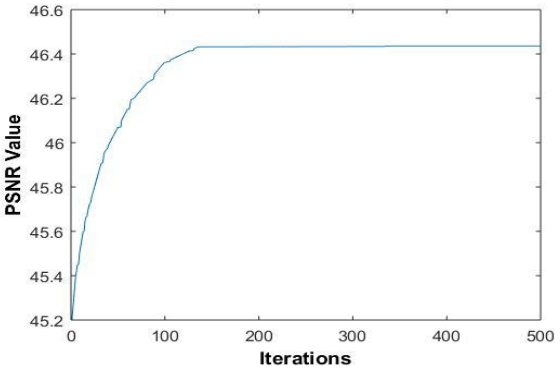


Fig. 8.PSNR value of stego-image jet vs. iterations

To evaluate the stego-image obtained from proposed technique Structural Similarity (SSIM) Index is also used.SSIM quantifies the degradation in stego-image due to the process of data embedding. SSIM is a product of three terms namely luminance term, contrast term and structural term.

$$SSIM(x,y)=[l(x,y)]^{\alpha}\times[c(x,y)]^{\beta}\times[s(x,y)]^{\gamma} \quad (6)$$

In (6) if we put $\alpha=\beta=\gamma=1$and $C_3=C_2/2$ then we get SSIM by (7) as:

$$SSIM(x,y)=\frac{(2\mu_x\mu_y+C_1)\times(2\sigma_{xy}+C_2)}{(\mu_x^2+\mu_y^2+C_1)\times(\sigma_x^2+\sigma_y^2+C_2)} \quad (7)$$

In (7) μ_x, μ_y, σ_x, σ_y, and σ_{xy} are the local means, standard deviations, and cross-covariance for images *x* and *y*.C_3 and C_2 are two variables to stabilize the division with weak denominator**.** In Table 7 SSIM value of various stego-images obtained from proposed technique and LSB steganography is shown. From the results we can conclude that there is no significant degradation in the structural integrity.

Table- VII: Structural similarity index value of stego-images

Stego-image / Technique	Lena	Jet	Pepper	Sailboat	Baboon
Proposed technique	0.9976	0.9972	0.9976	0.9981	0.9992
LSB Steganography	0.9970	0.9963	0.9969	0.9975	0.9981

IV. CONCLUSION

In this paper, we have presented a steganography scheme with a high data embedding capacity and imperceptibility. The proposed scheme is highly imperceptible as the average PSNR value of all stego-images is more than 46.3 dB. Along with PSNR, another performance metric i.e. SSIM is also used to analyze the quality of stego-image. The average SSIM value of stego-images is more than 0.9964. These results assure the structural integrity of the image. It is almost impossible for anyone to extract the secret data from the stego-image without the secret key. The use of GA and PSO ensures that the secret data is hidden in most suitable locations and the order of embedding will cause least changes. The statistical analysis on the result assures us the similar performance of the proposed technique almost every time as standard deviation is very small.

ACKNOWLEDGMENT

Authors would like to thank the research center G. H. Raisoni College of Engineering and Management, Pune for all the support and technical assistance for carrying the research. Author P. D. Shah would like to thank the management of DYPSOE, Lohegaon for the whole hearted support during the research work. The research is sponsored by Savitribai Phule Pune University under ASPIRE Research Mentorship Grant (18TEC000842).

REFERENCES

1. Kasana, G., Singh, K. and Bhatia, S., "Singular Value Decomposition based Steganography Technique for JPEG2000 Compressed Images", *International Journal of Engineering (IJE), TRANSACTIONS C: Aspects*,Vol. 28, No. 12, (2015), pp. 1720-1727.
2. Yadav, G. and Ojha, A., "Hamiltonian path based image steganography scheme with improved imperceptibility and undetectability", *Applied Soft Computing*, Vol. 73, (2018), pp. 497-507.
3. Wang, S., Yang, B. and Niu, X., "A secure steganography method based on genetic algorithm" *Journal of Information Hiding and Multimedia Signal Processing*, Vol. 1, No. 1, (2010), pp. 28-35.
4. Bedi, P., Bansal, P. and Sehgal, P., "Using PSO in a spatial domain based image hiding scheme with distortion tolerance", *Computers & Electrical Engineering*, Vol. 39, No. 2, (2013), pp. 640-654.
5. Kanan, H. and Bahram, N., "A novel image steganography scheme with high embedding capacity and tunable visual image quality based on a genetic algorithm", *Expert Systems with Applications*, Vol. 41, No. 14, (2014), pp. 6123-6130.
6. Maheswari, S. and Hemanth, D., "Performance enhanced image steganography systems using transforms and optimization techniques", *Multimedia Tools and Applications*, Vol. 76, No. 1, (2017),pp. 415-436.
7. Shah, P. and Bichkar, R., "A Secure Spatial Domain Image Steganography Using Genetic Algorithm and Linear Congruential Generator", In International Conference on Intelligent Computing and Applications, *Advances in Intelligent Systems and Computing, Springer, Singapore,* (2018), pp. 119-129.
8. Shah, P. and Bichkar, R., "Genetic Algorithm Based Imperceptible Spatial Domain Image Steganography Technique with High Payload Capacity", *International Journal of Recent Technology and Engineering*, Vol. 7, No. 5, (2019), pp. 224-229.
9. Nahvi, H. and Mohagheghian, I., "A Particle Swarm Optimization Algorithm for Mixed-Variable Nonlinear Problems", *International Journal of Engineering-Transactions A: Basics*, Vol. 24, No. 1, (2009), pp. 65-78.
10. Bahalke, U., Yolmeh, A., Hajizade, A. and Bahalke, A., "Genetic and tabu search algorithms for the single machine scheduling problem with sequence-dependent set-up times and deteriorating jobs", *International Journal of Engineering-Transactions A: Basics*, Vol. 23, No. 3, (2010), pp. 227-233.
11. Chan, C. and Cheng, L., "Hiding data in images by simple LSB substitution", *Pattern recognition*, Vol. 37, No. 3, (2004), pp. 469-474