Engineering Science and Technology, an International Journal 24 (2021) 782–794

HOSTED BY

Contents lists available at ScienceDirect

Engineering Science and Technology, an International Journal

journal homepage: www.elsevier.com/locate/jestch

Secret data modification based image steganography technique using genetic algorithm having a flexible chromosome structure

Pratik D. Shah [a,*], Rajankumar S. Bichkar [b]

[a] Department of E&TC, G H Raisoni College of Engineering and Management, Savitribai Phule Pune University, Pune, Maharashtra, India
[b] Vidya Pratishthan's Kamalnayan Bajaj Institute of Engineering and Technology, Baramati, Savitribai Phule Pune University, Pune, Maharashtra, India

ARTICLE INFO

Article history:
Received 28 May 2020
Revised 9 November 2020
Accepted 26 November 2020
Available online 26 January 2021

Keywords:
Image steganography
Genetic algorithm
Information hiding
LSB steganography

ABSTRACT

In all forms of confidential communication, the most significant element is security. Cryptography can be used to secure the information, but it discloses the presence of covert communication. Hence, steganography was invented; steganography is an art of concealed communication. Steganography is performed by inserting the secret information in a cover media to camouflage the presence of covert communication. In image steganography, an image is used as the cover media for secure communication. There are various image steganography techniques invented by many researchers, but a small number of them focus on improving visual quality and increasing payload capacity simultaneously. This paper presents a secret data modification based high capacity image steganography technique using genetic algorithm (GA). This new technique uses the least significant bit (LSB) replacement steganography, to embed the secret data. However, the secret data is rearranged and modified before embedding it in the LSBs of the cover image. The parameters used to rearrange and modify the secret data are controlled by GA. A unique concept called flexible chromosome is introduced which allows GA to interpret the chromosome value in different ways. GA attempts to find the best possible parameter value that yields high visual quality of the stego images. The stego images produced by the proposed technique have an average PSNR value of 46.41 dB and 40.83 dB for 2 bit per pixel (bpp) and 3 bpp data hiding capacity respectively.

1. Introduction

The goal of steganography is to hide the presence of secret communication [1]. It is accomplished by inserting the secret data in the cover image without incurring significant changes to it [2]. The insertion of the secret data in the cover image yields the stego image [3]. Reducing the difference between the stego and cover image is the primary objective of image steganography [4]. The major parameters used to evaluate steganography techniques are imperceptibility, payload capacity and security [5]. Optimizing all the parameters at once is a very challenging task. For example, if we try to increase the payload capacity, the imperceptibility and security will be degraded [6]. Similarly, if we try to improve any other parameter it leads to degradation of the remaining two parameters.

LSB replacement steganography is the most conventional digital steganography technique [7]. Most of the spatial domain image steganography techniques are inspired and modified versions of LSB steganography. In LSB steganography, the LSBs of the cover image pixels are replaced by the secret data bits. The number of LSBs to be replaced by secret data is dependent on the amount of secret data to be embedded. The secret data can be either embedded in all the pixels of the cover image or it can be embedded in a few selected pixels. The information of these selected pixels is given through a secret stego-key [8].

The advent of LSB steganography gave rise to various versions of this traditional steganography technique. Few authors have used image features like edge, texture, brightness, intensity levels, etc. to decide the amount of data to be hidden in the cover image [9]. Few researchers have even used the approach of inserting additional redundant random bits along with the secret data bits, to make the histogram of the stego image similar to the histogram of the cover image. LSB+ technique is one such example, but this method decreases the data embedding capacity. An improved version of the LSB+ technique was presented by K. Qazanfari et al.; their technique identified the sensitive pixels and avoided them from LSB data insertion in higher-order bits [10]. This approach improved the imperceptibility and data hiding capacity.

* Corresponding author.
E-mail address: shahpratik219@gmail.com (P.D. Shah).
Peer review under responsibility of Karabuk University.

https://doi.org/10.1016/j.jestch.2020.11.008

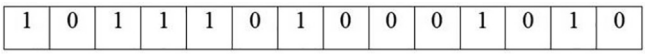

Fig. 1. Secret data bit-array with 14 bits.

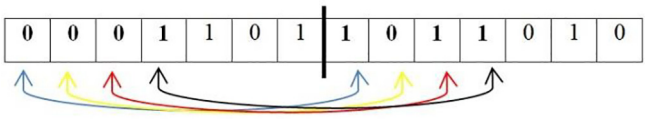

Fig. 2. Effect of $ns = 4$ on the secret data bit-array shown in Fig. 1.

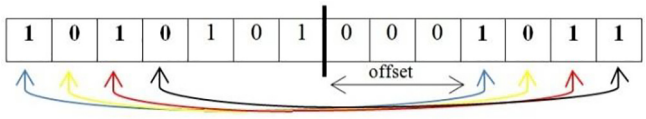

Fig. 3. Effect of the parameter $off = 3$ and $ns = 4$ on the secret data bit-array shown in Fig. 1.

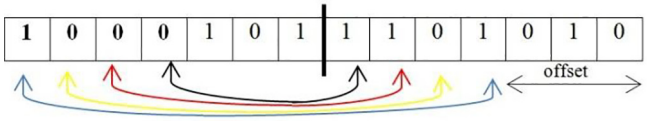

Fig. 4. Effect of the parameter $dd = 1$, $off = 3$ and $ns = 4$ on the secret data bit-array shown in Fig. 1.

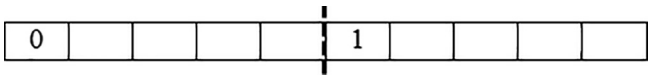

Fig. 5a. Secret data bit-array.

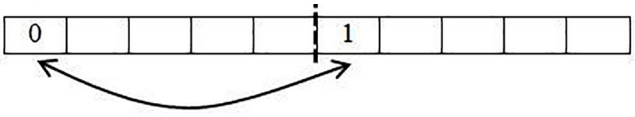

Fig. 5b. Effect of $dp = 00$ on the secret data bit-array shown in Fig. 5(a).

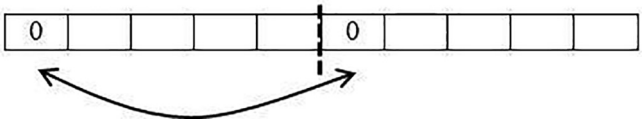

Fig. 5c. Effect of $dp = 01$ on the secret data bit-array shown in Fig. 5(a).

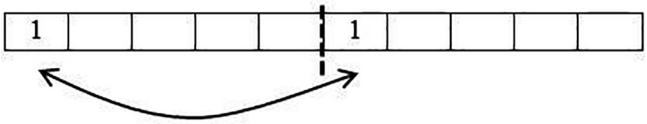

Fig. 5d. Effect of $dp = 10$ on the secret data bit-array shown in Fig. 5(a).

Several steganographic approaches have used encryption techniques, to encrypt the secret data before inserting them in the cover image [11–16]; this approach adds one more level of security. Few techniques have used adaptive approaches to select the number of bits to be embedded in different areas of the cover image [16,17]. These techniques use image characteristics like edges, noisy regions, etc. to hide more volume of data while restricting the amount of data for smooth areas.

In recent years, an immense amount of research has been carried out on steganography, but maximum amount of it is dedicated to the improvement of imperceptibility. Limited studies have concentrated on increasing imperceptibility and payload capacity concurrently. Few researchers have experimented using evolutionary computation techniques to achieve the tricky task of increasing the payload capacity and the imperceptibility at the same time.

A steganography technique having good visual quality and variable data embedding capacity using GA was proposed in [18]. This technique was developed in the spatial domain with a payload capacity of up to 4 bpp. The main focus of this research was to use GA for searching the best order and arrangements to insert the secret message in the image. Ranyiah Wazirali et al. [19] presented a high payload steganographic scheme that focuses on increasing the imperceptibility by using various operations like optimized pixel scanning order, circular shifting, flipping secret bits and transposing secret data. Their technique used GA to find optimum solutions. A GA and Particle swarm optimization (PSO) based approach to increase the imperceptibility with a good payload was presented in [20]. Similarly few other approaches [21–23] have explored PSO for improvement of visual quality and payload capacity. Hemanth et al. developed a GA based technique in Fresnelet transform and Discrete ripplet transform [24]. The authors used traditional GA and modified GA approaches to enhance the performance of the technique. Few other techniques have also explored the use of GA in transform domain for improvement in the imperceptibility with a huge payload capacity [25–27]. These techniques used Wavelet transform and Discrete cosine transform.

In this paper, we propose a GA based high payload capacity secret data modification steganography technique called SDMS. The SDMS technique modifies and rearranges the secret data to reduce the embedding error. LSB substitution steganography is used to embed this modified data in the cover image. The modification and rearrangement of the secret data are done based on four parameters namely; number of swaps, offset, data direction and data polarity. Simultaneous exploration for the best possible combination of these parameters is a very tedious task as there is enormous number of possibilities. To make this task simple we use GA. The chromosome structures used in GA comprises of all these parameters, and GA explores the best value of each parameter. Initially, this technique was developed for a payload capacity of 1 bpp. However, we modified it to increase the embedding capacity until 3 bpp.

The decision of choosing GA for the proposed technique is based on many factors, but the most prominent factor being the robustness of GA with respect to local minima and maxima. Many optimization algorithms have a steady transition in the search space and they get struck in local minima and maxima. In GA, we start the search at various different locations in the search space and the transition is carried out by using probabilistic rules instead of deterministic rules [28]. The GA works on the chromosomes and not on the parameters directly. Hence, it is the best choice to explore all parameters simultaneously. GA does not require any auxiliary information to operate this is one of the best things about GA. The choice of using GA over PSO was mainly due to the parameters involved in the search and optimization problem. In PSO each particle in the swarm has a position vector and a velocity vector [21]. Since, our technique involves many parameters having different length PSO was not a suitable choice for exploring the solution.

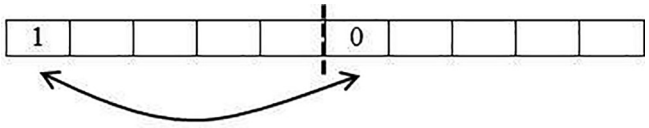

Fig. 5e. Effect of $dp = 11$ on the secret data bit-array shown in Fig. 5(a).

Furthermore, PSO gets trapped in local minima and maxima while solving complex problems [29]. GA also has one drawback of being computationally heavy as compared to other simpler optimization methods. However, in the proposed technique its implementation is kept simple and that makes it less time consuming. The use of mathematical models like Lagrange multiplier method and other such approaches can also be utilized to solve the optimization problems. However, they require much auxiliary information like gradient, derivatives, etc. and are computationally taxing [30].

The main contribution of this paper is modifying and rearranging the secret data to improve the imperceptibility at a high payload capacity. Most of the researchers have worked on finding various possibilities to insert the secret data in the cover image, but very few have explored the possibility of modifying the secret message before embedding it. In this paper, a new concept called flexible chromosome is proposed, which allows GA to interpret the same chromosome value in different ways.

The remaining paper is structured as follows: In Section 2, we explain the proposed technique Secret data modification based image steganography for 1, 2 and 3 bpp data embedding capacity. In the same section, the detailed explanation of all the parameters used for secret data modification, GA implementation details and the concept of the flexible chromosome is given. Section 3 illustrates the results of the proposed technique for 1, 2 and 3 bpp data hiding capacity. The results of the proposed technique SDMS are compared with several state of the art steganography techniques for the same test images. The paper concludes in Section 4, here the findings are summarized and future directions are highlighted.

2. Proposed technique: Secret data modification based image steganography (SDMS)

As discussed earlier, we use LSB substitution steganography to insert the data in the cover image. But, before that, we modify and rearrange the secret data. The prerequisite for explaining the proposed technique SDMS is the knowledge of the parameters used for modification and rearrangement of the secret data. In this section, we discuss all these parameters in detail. Along with it the implementation of GA its operators and chromosome structure is elaborated. The unique concept of the flexible chromosome is also described in this section. Finally, the algorithm for data insertion and extraction is presented in separate sub-sections.

2.1. Parameters used to modify and rearrange the secret data

To modify and rearrange the secret data, it is initially converted into a one-dimensional bit-array. The quantity of data which may be inserted in the cover image depends on two factors. The first factor is the cover image size and the second factor is the total bit planes used for data insertion. Initially, we will explain only 1 bpp data embedding, later we will discuss the process for increasing the payload capacity by hiding data in 2nd and 3rd bit planes. Consider a cover image that has m rows and n columns, the maximum amount of the secret data that can be inserted at 1 bpp data embedding capacity is $m \times n$ bits. The proposed technique can also be used to insert fewer amounts of data. However, $m \times n$ bits will be the maximum capacity at 1 bpp steganography. The size of the bit-array is dependent on the size of the secret messages. The secret message is converted to its binary form and it is stored in one-dimensional bit-array for further processing. Later, each bit in this array is inserted in the LSB of the cover image pixel sequentially from topmost left corner pixel. However, before performing LSB replacement steganography we modify and rearrange the secret data using four parameters viz. number of swaps, offset, data direction and data polarity. These parameters are explained in the next sub-sections.

Number of swaps (*ns*)	Offset (*off*)	Data direction (*dd*)	Data polarity (*dp*)

Fig. 6. Chromosome structure.

Flexible chromosome (*fc*)	Parameter 1	Parameter 2	Parameter 3	Parameter 4

Fig. 7. New chromosome structure with the *fc* gene.

Flexible chromosome value (*fc*)	Effective changes in interpretation of chromosome			
00	Number of swaps (*ns*)	Data direction (*dd*)	Data polarity (*dp*)	Offset (*off*)
01	Number of swaps (*ns*)	Data polarity (*dp*)	Data direction (*dd*)	Offset (*off*)
10	Offset (*off*)	Data direction (*dd*)	Data polarity (*dp*)	Number of swaps (*ns*)
11	Offset (*off*)	Data polarity (*dp*)	Data direction (*dd*)	Number of swaps (*ns*)

Fig. 8. Effect of *fc* gene on the interpretation of the GA chromosome.

2.1.1. Number of swaps (*ns*)

This parameter is employed to rearrange the bit-array. The bit-array is divided into two equal parts and *ns* bits from the first half in the bit-array are swapped with the second part of bit-array. For illustration purpose a secret data bit array of 14 bits is considered and shown in Fig. 1. The data in bit-array is selected randomly for illustration purpose. The effect of number of swaps parameter on the bit array for the value of $ns = 4$ is illustrated in Fig. 2. Let the size of the bit-array be $len = m \times n$. Hence, the maximum value of this *ns* parameter will be $len/2$.

2.1.2. Offset (*off*)

The offset parameter is used in conjunction with the *ns* parameter. The *off* parameter specifies the starting location for swapping the elements from the second part of the bit-array. Fig. 3 depicts the effect of the offset parameter with the value of $off = 3$ and $ns = 4$. The initial data in the bit-array is the same as the one in Fig. 1. In Fig. 3, it can be noticed that the swapping starts from the beginning of the first part of the bit-array whereas in the second part the swapping starts from the offset value. So, the effect of the offset value is only for the second part of the bit-array and it does not affect the first part. The maximum value of this parameter will also be $len/2$. It can be observed that in some cases the value of $ns + off$ may exceed $len/2$. In that case, we will keep swapping the bits till we reach the last element in the second part of the bit-array, after that we will roll over and start from the beginning of the second part of the bit-array

2.1.3. Data direction (*dd*)

The order of swapping the elements in bit array was from left to right, whereas we may even swap them in opposite direction

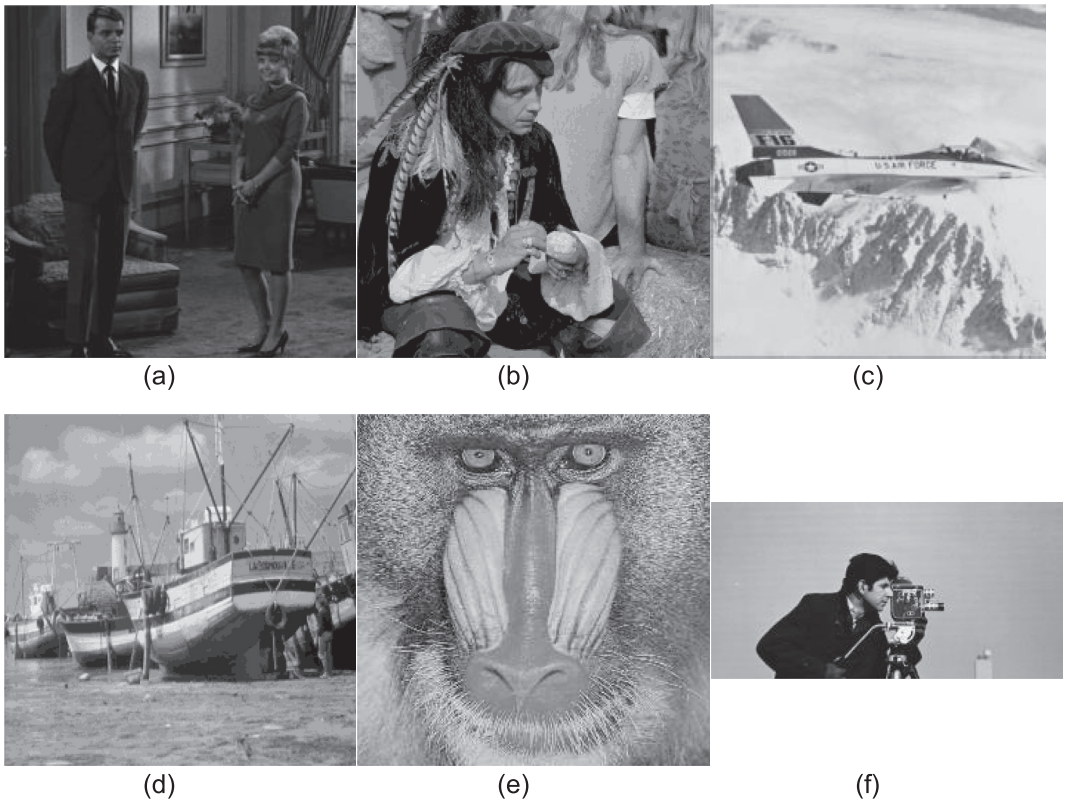

Fig. 9. Test images (a)–(e) Cover images (512 × 512) (Living room, Pirate, Airplane, Boat and Mandrill) (f) - Secret message image (128 × 256) (Cameraman).

Table 1
Stego image SSIM and PSNR value of the SDMS techniques and the LSB replacement steganography at 1 bpp payload capacity.

Parameter	PSNR			SSIM		
Cover Image	LSB	SDMS-SGA	SDMS-FC	LSB	SDMS-SGA	SDMS-FC
Living room	51.12	52.05	52.17	0.9982	0.9988	0.9989
Pirate	51.14	52.35	52.41	0.9980	0.9985	0.9986
Airplane	51.14	52.24	52.21	0.9971	0.9981	0.9979
Boat	51.12	52.28	52.35	0.9977	0.9986	0.9986
Mandrill	51.13	52.04	52.13	0.9986	0.9993	0.9994

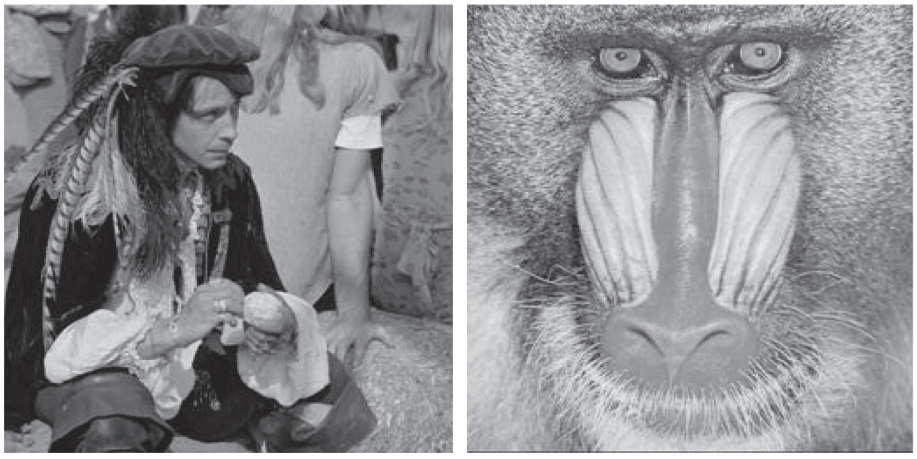

Fig. 10. Stego images Pirate and Mandrill.

i.e. from right to left. This possibility for rearranging the secret data is explored by using the data direction (*dd*) parameter. There are only two possible values for this parameter i.e. $dd = 0$ which implies swapping in left to right direction as used in previous examples and $dd = 1$ where the data is swapped from right to left direction. The effect of *dd* parameter is only applicable for second part of bit array. The effect of *dd* parameter is shown in Fig. 4. For illustration purpose, we consider $ns = 4$, $off = 3$ and $dd = 1$.

2.1.4. Data polarity (*dp*)

All the parameters discussed until now were used to rearrange the secret data in the bit-array. However, the data polarity parameter results in the modification of the secret data. The modification of the data in the bit-array is done by selectively complementing the bits. While swapping the data from the first and second part of the bit-array, we can modify it by selectively complementing the secret data bits. In some cases, we may not complement the swapped data bits and keep them unchanged, this decision of whether to complement the bits or not is taken by the *dp* parameter. There are four possibilities for data modification using *dp* hence this parameter also has four values viz. 00, 01, 10 and 11. The effect of all four values is illustrated with examples in Figs. 5a–5e. For illustration, we consider the length of the bit-array to be 10 and for simplicity of explanation, we have shown only one bit in both parts. In Fig. 5b, it can be seen that for the value of $dd = 00$ both the bits are complemented while swapping. Similarly for the value of $dd = 01$, the bit moved to the left part of bit-array is complemented whereas the bit moved to the right part is kept unchanged. The exact opposite action is performed on the bit-array for the value of$dd = 10$. Finally, if the value of $dd = 11$, regular swapping of bits from both the parts is performed without complementing. All the four parameters work together for rearrangement and modification of the secret data (Fig 5e).

2.2. Genetic algorithm

GA is inspired by Darwin's theory of evolution and it obeys the principle, "survival of the fittest". GA is a population-based technique and each individual in the population is a possible solution for the problem being solved. These individuals contest one another for their survival and to produce new solutions for the next generation. The population evolves as the generation progress. To evaluate the efficiency of each individual in the population a fitness function is defined and it is used to assess their performance [28].

In the proposed technique, we use GA to find the best values for the parameters *ns*, *off*, *dd* and *dp*. As a result of this exploration, these parameter values should result in very few changes in the cover image. The chromosome structure of every individual in the population consists of these four parameters and it's shown in Fig. 6. The size of *dd* gene is 1 bit; the size of *dp* gene is 2 bits. But, the size of the *ns* and *off* gene is not fixed as it is dependent on the size of the bit-array. If we choose a cover image of size 512×512, then the maximum data that can be inserted at 1 bpp will be 262,144 bits and the same will be the size of the bit-array. This bit-array will be divided into two parts for swapping the data, so the maximum value for *ns* and *off* will be $2^{17} = 131072$. Hence, we have allotted 17 bits for *ns* and *off* gene. The total size of the chromosome for 1 bpp steganography will be 37 bits. Hence, the total possible combinations of all these four parameters will be $2^{37} = 1.37 \times 10^{11}$. Exploring all the possible values is a very difficult and time-consuming task, so we employ GA to help us find near optimum solutions.

The implementation of the genetic algorithm is carried out for 1000 iterations. The initial population will be produced randomly. In each iteration, 100 chromosomes will compete with one another to produce the population for the next generation. To assess the competence of all the individuals in the population, the PSNR value of the stego image is used as the fitness function. The roulette wheel selection strategy is used to choose the individuals from the population for reproduction. Two-point crossover is performed on these chromosomes with a crossover probability of 0.9 to produce new individuals. Mutation is performed on these new solutions with a probability of 0.1. The critical parameters of the GA are decided after performing the sensitivity analysis. The Elitist strategy is used to retain the best individual in the population. The entire process is executed for 1000 iterations in search of a near-optimal solution.

2.3. Flexible chromosome

In this paper, a new concept for the implementation of GA called the flexible chromosome is proposed. This concept deals with the interpretation of the chromosome. In the previous subsection, we have discussed the chromosome structure for the proposed technique. But, the interpretation of each part of the chromosome was fixed. For example, the first 17 bits of the chromosome will be used for the value of the *ns* parameter; simi-

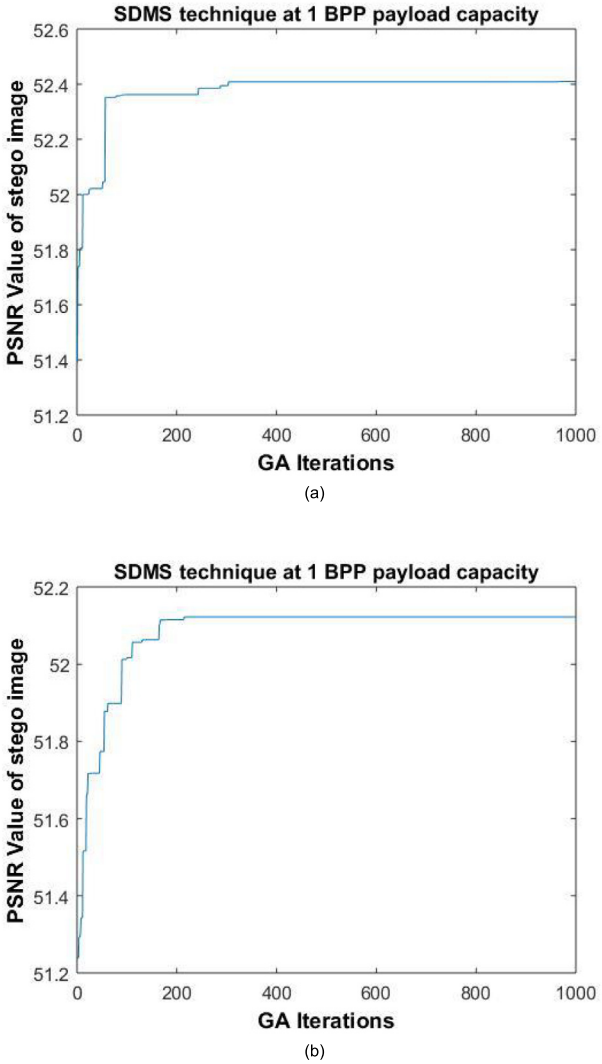

Fig. 11. PSNR value of stego images (Pirate and Mandrill) for each GA iteration.

Fig. 12. Natural images used for experimentation (Flower, Beach, Car and Building).

Table 2
PSNR value of SDMS and LSB steganography at different payload capacities.

Payload	0.25 BPP		0.5 BPP		1 BPP	
Cover Image	LSB	SDMS-FC	LSB	SDMS-FC	LSB	SDMS-FC
Flower	57.16	57.42	54.20	54.71	51.11	52.06
Beach	57.12	57.53	54.10	54.78	51.15	52.48
Car	57.14	57.45	54.13	54.83	51.12	52.40
Building	57.14	57.41	54.15	54.69	51.14	52.27

larly, the next 17 bits will be used for the *off* parameter and so on. However, this new concept of flexible chromosome will enable us to interpret the same chromosome differently. The new chromosome structure for the proposed technique will have one additional gene called the flexible chromosome (*fc*), and the value of *fc* will decide the interpretation of the chromosome. The gene *fc* allows us to interpret the chromosome as the permutation of the parameters. Fig. 7, depicts the new chromosome structure with *fc* gene. Here, we don't have a fixed location for any specific parameter and the interpretation of these values depends on *fc*. Fig. 8, illustrates the interpretation of the chromosome depending on the value of *fc*. Four different interpretations of the same chromosome based on the value of *fc* can be noted from Fig. 8. The actual size of *fc* is 5 bits, but for the explanation of the concept, we have assumed the size of *fc* as two bits. The total possible combinations with four parameters are 24 hence we require at least 5 bits to represent all possible combinations. The GA is considered to be significantly robust, and is known for not getting stuck in local minima and maxima [28]. However, as the search space is enormous and to explore maximum possibilities in it we have used flexible chromosome. The flexible chromosome is very effective as it provides variation in the population and helps to get out of fixed search space.

To get the secret data back from the stego image, the receiver requires the chromosome value used to modify the secret data. Hence, in the proposed technique we hide the chromosome value as the secret key in the stego image. Two approaches can be employed for inserting the secret key in the stego image. The first approach is used in the case when the size of the bit-array is greater than $(len - 42)$ bits. In this case, the secret key is concealed in the 2nd LSB of the border pixels of the image. The second approach is used when the size of the bit-array is less than $(len - 42)$ bits. In this case, we start hiding the secret key in LSB of the last pixel in the bottom rightmost corner of the stego image and continue towards left.

2.4. Algorithm for secret message insertion (1 bpp)

Algorithm 1. Secret message insertion (1 bit per pixel)

Input: Cover image$I = \{i_1, i_2, \cdots, i_{(m\times n)}\}$,
Secret data (image)$D = \{d_1, d_2, \cdots, d_{(m/4)\times(n/2)}\}$
Output: Stego image $O = \{o_1, o_2, \cdots, o_{(m\times n)}\}$

1. Initialize a one-dimensional bit-array *sm* of length $len = m \times n$ as,$sm \leftarrow (D_8 toD_1)\forall d_i$
2. Initialize
   popmax ← *population size*
   imax ← *numberofiterations*
3. Randomly create *popmax* number of chromosomes for GA.
4. *for* $i = 1$ *to i max*
   for $p = 1$ *to popmax*
   Interpret the chromosome based on the value of *fc* as discussed in section 2.3.
   Modify and rearrange the bit-array *sm* using the chromosome value as discussed in section 2.1.
   Insert the data from the bit-array *sm* in the LSB of the cover image producing the stego image*O*.
   Calculate the PSNR value of this stego image*O*.
   Store the fitness value of the chromosome in the array *fitness*(p) ← PSNR.
   end
   Select two individuals from the population using roulette wheel perform two-point crossover on these chromosomes with cross selection strategy.
   over probability of 0.9 to produce two new individuals.
   Perform mutation on these chromosomes with the mutation probability of 0.1.
   Repeat the above steps of selecting two individual and performing the crossover and mutation operators till we don't get 100 new chromosomes.
   Apply elitism.
   end
5. Hide the chromosome value as the secret key

2.5. Algorithm for extracting secret data (1 bit per pixel)

Algorithm 2. Extracting secret data (1 bit per pixel)
Input: Stego image $O = \{o_1, o_2, \cdots, o_{(m\times n)}\}$
Output: Secret data (image) $D = \{d_1, d_2, \cdots, d_{(m/4)\times(n/2)}\}$
Obtain the secret key from the predetermined positions of the stego image.

1. Get all the LSB from the stego image and store them in the bit-array *se*.
2. From the extracted secret key, interpret the value of GA chromosome.
3. Modify and rearrange the bit array *se* using the interpreted chromosome value.
4. Recreate the secret message image *D* using *se*.

2.6. Modification of the proposed technique SDMS to perform 2 bpp & 3 bpp steganography

The algorithm used for embedding 1 bpp can be extended to accommodate 2 and 3 bpp with minor modifications. These modifications include using an extra set of genes and the secret data bit-array for each bit plane of the cover image. The strategy of inserting the data sequentially from the bit-array into the cover image

Table 3
Sensitivity analysis to decide the crossover and mutation rate of GA.

Probability of crossover	Probability of mutation	Objective function value
0.9	0.1	52.34
	0.2	52.31
	0.3	52.27
	0.4	52.28
0.8	0.1	52.32
	0.2	52.29
	0.3	52.25
	0.4	52.22
0.7	0.1	52.29
	0.2	52.26
	0.3	52.23
	0.4	52.19

LSBs is the same. But as we increase the embedding capacity from 1 bpp to 2 bpp and 3 bpp, we need extra bit-arrays to accommodate the secrete data. These secret data bit-array will be embedded in the higher bit planes. For example, if we want to perform 2 bpp steganography, then we need to convert the secret data into two separate bit-arrays. The size of these bit-arrays depends on the quantity of secret data to be inserted. The chromosome structure will contain five genes as shown in Fig. 7 for each bit-array. Similarly, we can also perform 3 bpp steganography. However, that will require three bit-arrays to store the secret data and for modification of these bit-arrays, we will also require three sets of genes as shown in Fig. 7. The same concept can be extended further to 4 bpp steganography, but it causes visual artifacts and reduces the fidelity of the stego image. Hence, we have restricted only till 3 bpp steganography.

The Optimal pixel adjustment process (OPAP) developed by Chan and Cheng [31] is used in the proposed technique for the reduction of the error caused due to LSB replacement steganography. OPAP is only applicable to steganographic techniques having data embedding capacity of 2 bpp or more. Hence, it is not used in SDMS technique having payload capacity less than 2 bpp. The OPAP reduces the embedding error by modifying the higher bit plane that is not used for inserting secret data. For example, in the case of 2 bpp steganography, OPAP modifies the 3rd LSB to reduce the embedding error. To understand the effect of OPAP in reducing the embedding error, let us assume 11 as the binary data to be inserted in the cover image pixel with intensity 68 (0100 0100). The resultant stego image pixel value will be 71 (0100 0111) which results in an embedding error of 3. To reduce this embedding error OPAP modifies the 3rd LSB and changes it to 0, hence the resultant stego image pixel after OPAP will be 67 (0100 0011). This reduces the embedding error from 3 to −1. In our technique SDMS, we have used the OPAP in 2 bpp and 3 bpp variant and the results of our technique with OPAP and without OPAP are compared in Section 3.

The secret data extraction process for 2 and 3 bpp steganography remains mostly identical to the process explained in Section 2.5 for 1 bpp steganography, however it has few minor modifications. The steps involved for extracting secret data from the stego image with 2 and 3 bpp is as follows:

1. Obtain the secret key from the predetermined positions of the stego image.
2. Convert the secret key into separate GA chromosomes, which will be used for each bit array.
3. Extract the LSBs from the stego image and store them in the separate bit-array as per the bpp capacity of the technique.
4. Use the respective GA chromosome to modify and rearrange the secret data in the bit array.
5. Recreate the secret message image using the data in the bit arrays.

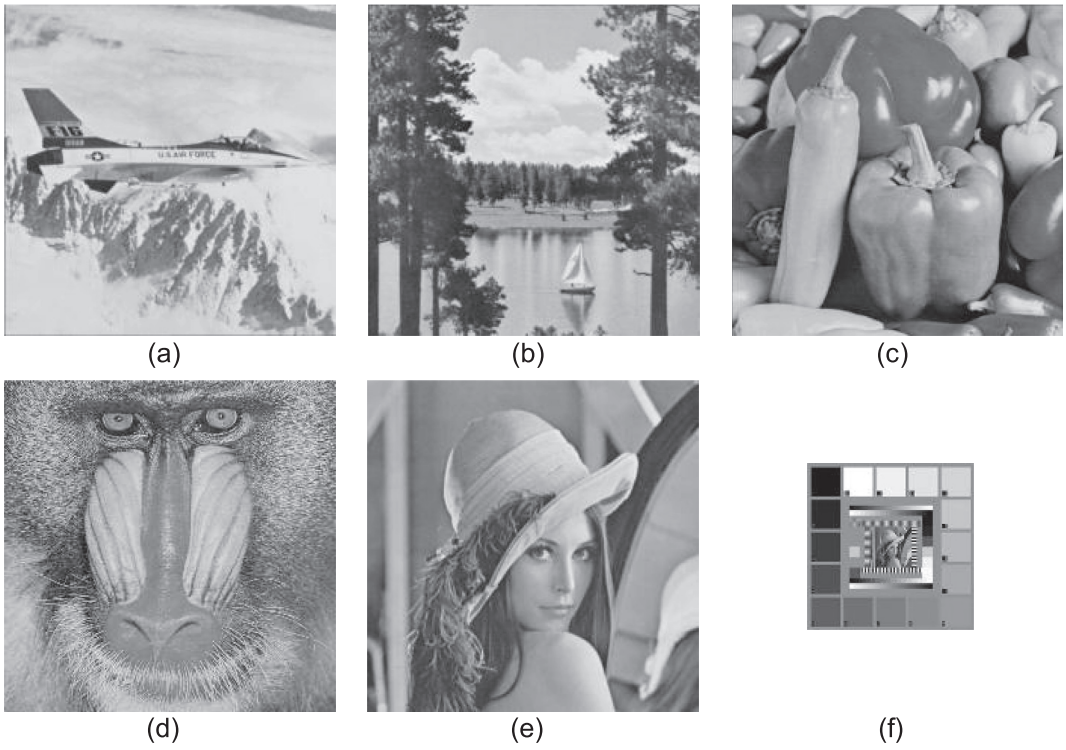

Fig. 13. Test images (a)–(e) Cover images (512 × 512) (Airplane, Lake, Pepper, Mandrill and Lena) (f) - Secret data (256 × 256) (Test pattern).

Table 4
Stego image PSNR value of the SDMS technique and other steganography algorithms at 2 bpp payload capacity.

Cover image	Lin's technique [33]	Yang's technique [34]	Chang's technique [35]	Wu's technique [36]	Kanan's technique [18]	SDMS technique	SDMS with OPAP technique
Airplane	39.25	41.66	40.73	43.53	45.18	45.23	46.37
Lake	39.18	41.51	38.86	43.55	45.10	45.22	46.42
Pepper	39.17	41.56	39.30	43.56	45.13	45.34	46.39
Mandrill	39.18	41.55	39.94	43.54	45.12	45.19	46.42
Lena	39.20	41.60	40.37	43.54	45.12	45.26	46.43

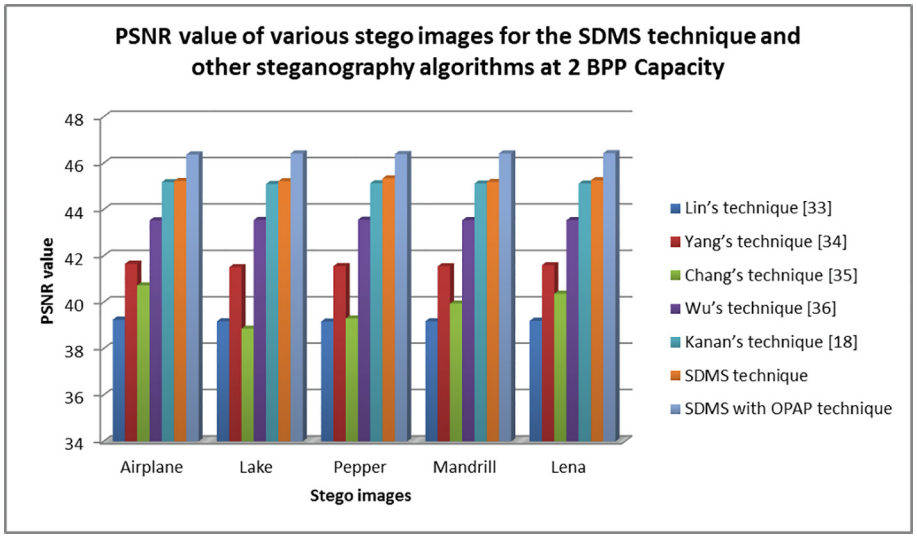

Fig. 14. PSNR value of various stego images for the SDMS technique and other steganography algorithms at 2 BPP capacity.

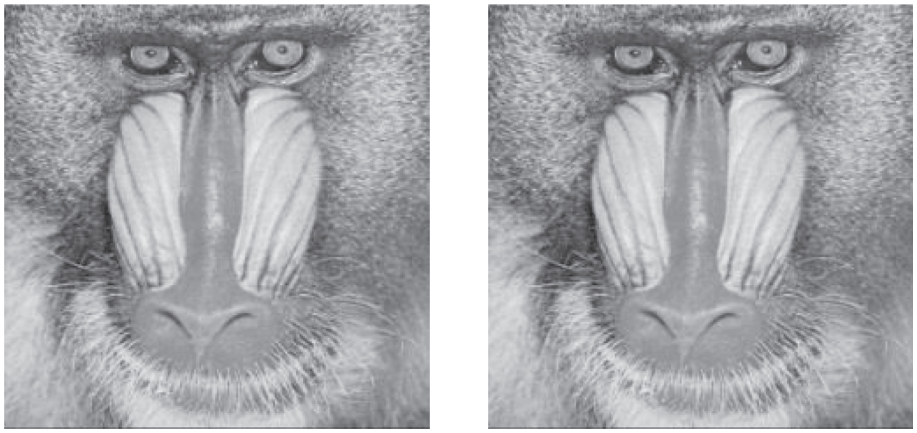

Fig. 15. Mandrill stego image obtained from (a) SDMS technique (b) SDMS technique with OPAP.

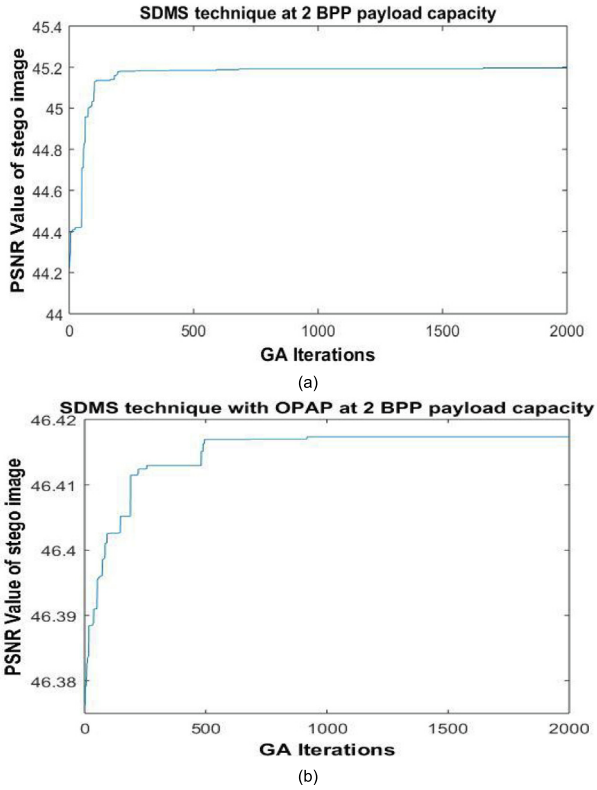

Fig. 16. PSNR value of the Mandrill stego image for each GA iteration using (a) SDMS technique (b) SDMS technique with OPAP.

3. Results and discussions

In this section, the results of the proposed scheme are compared with a number of popular and state of the art techniques. We have implemented the SDMS technique on Matlab 2018b version. The system having an Intel Core i5 processor with an 8 GB RAM was used for the experimentation. The test images were used from the USC-SIPI Image Database [32]. Different sets of images were used for 1 bpp, 2 bpp and 3 bpp payload capacity of the proposed technique. The images used for each case and subsequent results obtained from the experimentation are shown in the following subsections. To compare the efficiency of the proposed technique and to evaluate the imperceptibility, we use the PSNR parameter and Structural similarity index (SSIM). The comparison of the proposed technique with several other algorithms is performed by using same dataset. The PSNR is the ratio of the maximum power of a signal to the power of its corrupting noise. To obtain the PSNR value of any stego image, we have to first compute the MSE value. MSE of any stego image can be obtained by Eq. (1) and PSNR by Eq. (2). In Eq. (1), M and N denote the rows and columns of the image respectively. X_{ij} and Y_{ij} are pixel intensities of ij^{th} location of the cover image and the stego image respectively. This section also discusses the results of the extraction process. The extracted secret data is compared with the embedded secret data using the Bit error ratio (BER) evaluation parameter.

Table 5
SSIM value of stego images at 2 bpp capacity for SDMS and LSB technique.

Cover Image	LSB	SDMS without OPAP	SDMS with OPAP
Airplane	0.9935	0.9963	0.9968
Lake	0.9957	0.9971	0.9980
Pepper	0.9946	0.9958	0.9975
Mandrill	0.9981	0.9987	0.9992
Lena	0.9946	0.9955	0.9974

$$MSE = \frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N}\left(X_{ij} - Y_{ij}\right)^2 \quad (1)$$

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

3.1. Results of the SDMS technique for 1 BPP Data Embedding Capacity

The gray scale test images used for performing 1 bpp steganography are shown in Fig. 9. Living room, Pirate, Airplane, Boat and Mandrill are used as cover images and their size is 512×512. The cropped Cameraman image of size 128×256 is used as the secret data. In Table 1, the SSIM and PSNR value of the SDMS technique with flexible chromosome and traditional simple genetic algorithm (SGA) is compared with LSB steganography at 1 bpp payload capacity. The average PSNR value of the stego images obtained from the SDMS-FC technique is more than 52.2 dB which is slightly more than SDMS-SGA technique. When compared to the LSB steganography, both the SDMS techniques provide an average 1 dB improvement in PSNR value of the stego images. This difference of 1 dB is quite significant and hence we can conclude that the SDMS technique is more imperceptible. This improve-

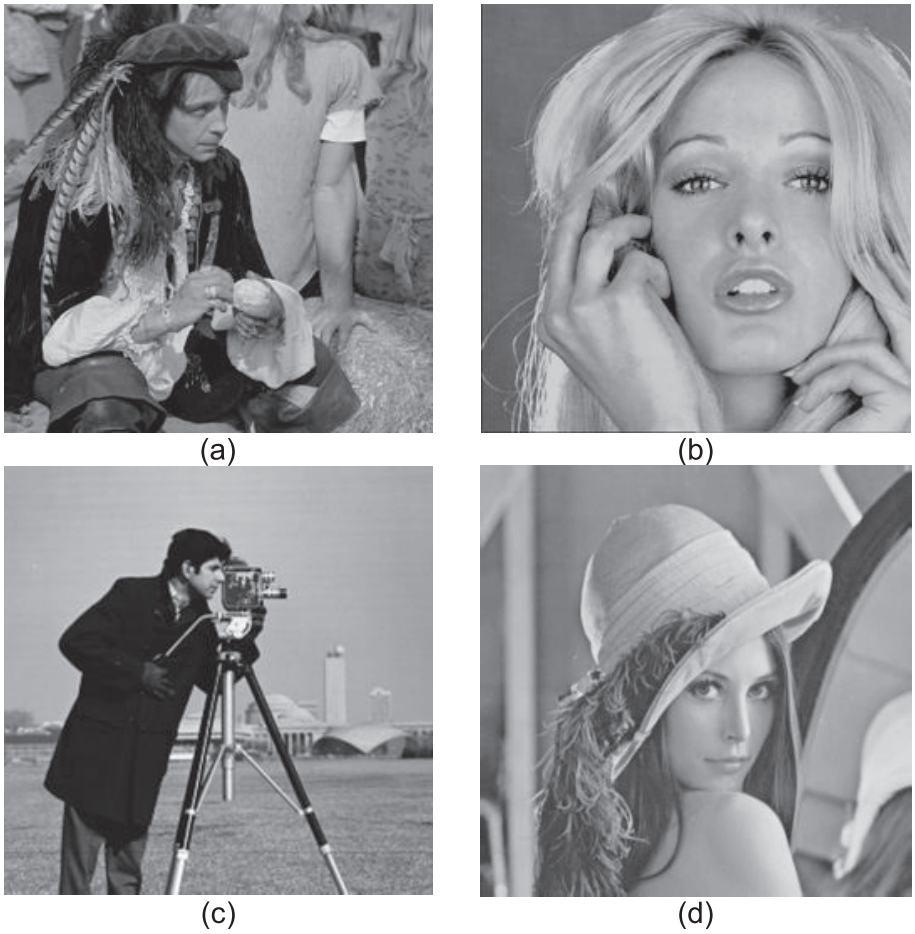

Fig. 17. Test images (Pirate, Blonde, Cameraman and Lena).

Table 6
Stego image PSNR value of the SDMS technique and other steganography algorithms at 3 bpp payload capacity.

Cover image	Eslami's technique [38]	Kanan's technique [18]	Wu's technique [39]	Yang's technique [40]	Yadav's technique [37]	SDMS technique	SDMS with OPAP technique
Pirate	37.86	37.21	37.06	38.11	39.10	39.78	40.82
Blonde	37.28	37.82	37.44	38.32	39.15	39.84	40.91
Cameraman	37.92	38.18	37.40	38.31	39.27	39.82	40.82
Lena	37.54	37.81	37.33	38.25	39.38	39.88	40.75

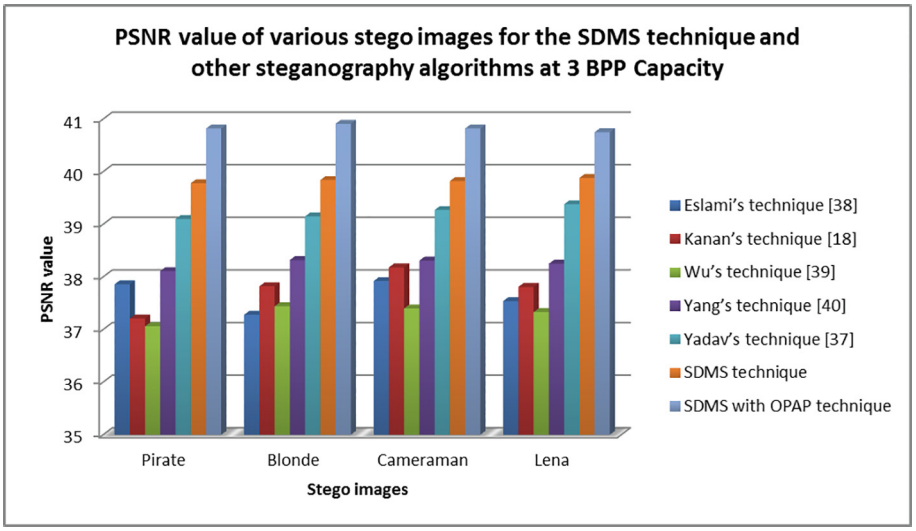

Fig. 18. PSNR value of various stego images for the SDMS technique and other steganography algorithms at 3 BPP capacity.

Fig. 19. Pirate stego image obtained from (a) SDMS technique (b) SDMS technique with OPAP.

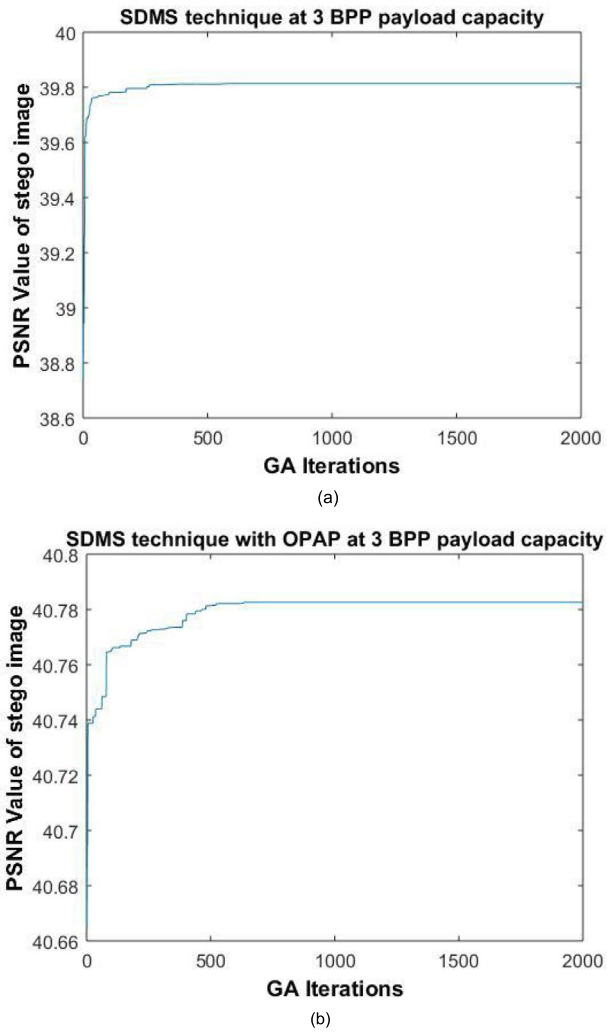

Fig. 20. PSNR value of the Pirate stego image for each GA iteration using (a) SDMS technique (b) SDMS technique with OPAP.

ment in the results is possible because of exploring different possibilities to rearrange and modify the secret data using GA. The stego images (Pirate and Mandrill) obtained from embedding the data by the SDMS technique is presented in Fig. 10. The visual inspection of these stego images will not create any suspicion in an observer's mind as the visual quality of these stego images is very high. The graph of the PSNR value of the stego images (Pirate and Mandrill) vs. GA iteration is shown in Fig. 11. The PSNR of the stego image in the early iterations is approximately 51.2 dB, but as the iterations progress, this value goes beyond 52 dB. This illustrates the impact of GA in the search of near optimum parameter values. The average execution time for the SDMS technique was 8.42 s.

Table 7
SSIM value of stego images at 3 bpp capacity for SDMS and LSB technique.

Cover Image	LSB	SDMS without OPAP	SDMS with OPAP
Pirate	0.9751	0.9765	0.9786
Blonde	0.9685	0.9692	0.9710
Cameraman	0.9615	0.9627	0.9635
Lena	0.9699	0.9705	0.9716

Few natural images captured from Canon 1300D were also used for validating the performance of the SDMS technique. These images were converted to gray scale with a dimension of 512×512. These images are shown in Fig. 12. Different payload capacities were explored on these images such as 0.25 bpp, 0.5 bpp and 1 bpp. The secret data was a randomly generated bit stream as per the payload capacity. The result of these experiments is presented in Table 2. To select the crossover and mutation rate of GA, sensitivity analysis was done in which fitness value of the solution is evaluated for various mutation and crossover rates and the best result is used to decide the parameter values. In Table 3, the average object function value i.e. the PSNR value of stego image for 1 bpp is obtained for various crossover and mutation rate of the GA. The best result in Table 3 corresponds to the crossover rate 0.9 and mutation rate 0.1. Hence, these values are used for implementation of GA in SDMS technique.

3.2. Results of the SDMS technique for 2 BPP Data Embedding Capacity

The gray scale test images used for performing 2 bpp steganography are shown in Fig. 13. Airplane, Lake, Pepper, Mandrill and Lena are used as the cover image, their size is 512×512. The secret message is the test pattern image of size 256×256 shown in Fig. 13 (f). In Table 4, the result of the SDMS technique at 2 bpp payload capacity is compared with several prevalent steganography algorithms. The SDMS technique with OPAP is also included in the result analysis. The graphical representation of these results

Table 8
BER value of extracted secret data at various bpp payload capacities.

1 BPP Payload		2 BPP Payload		3 BPP Payload	
Cover Image	SDMS	Cover Image	SDMS	Cover Image	SDMS
Living room	0	Airplane	0	Pirate	0
Pirate	0	Lake	0	Blonde	0
Airplane	0	Pepper	0	Cameraman	0
Boat	0	Mandrill	0	Lena	0
Mandrill	0	Lena	0	–	–

is shown in Fig. 14. From the result analysis, it can be concluded that the performance of the SDMS with OPAP technique is better than other steganography techniques. The SDMS with OPAP technique produces the stego images that have an average PSNR value greater than 1 dB as compared to the Kanan et al.'s technique [18]. The Kanan et al.'s technique explores starting location and arrangements to insert the secret data, whereas the proposed SDMS technique modifies secret data based on many parameters. The availability of many parameters for GA to explore for best solution and use of OPAP is the probable reason for the improvement in results. The stego image of Mandrill obtained from the SDMS technique and the SDMS technique with OPAP is shown in Fig. 15. The graph of the PSNR value of Mandrill stego image vs. GA iteration is shown in Fig. 16. In Table 5, the SSIM value of stego images is shown for the proposed technique in comparison with LSB steganography at 2 bpp capacity.

3.3. Results of the SDMS for 3 BPP Data Embedding Capacity

The gray scale test images used for performing 3 bpp steganography are shown in Fig. 17. Pirate, Blonde, Cameraman and Lena image of size 512 × 512 are used as the cover image and a binary data stream of 7,86,432 bits is used as the secret data. In Table 6, the result of the SDMS algorithm is compared with several state of the art image steganography algorithms at 3 bpp payload capacity for same dataset. The SDMS technique with OPAP is also included in the result analysis. The graphical representation of these results is shown in Fig. 18. From the result analysis, it can be concluded that the performance of the SDMS technique with OPAP is better than other steganographic algorithms including the latest algorithm by Yadav and Ojha [37]. In the technique proposed in [37] the authors have used Hamilton path generated patterns to insert the secret data in blocks of the cover image. They use the same pattern for all the blocks of the image but the pattern is rotated for each block to reduce the distortion. Since, this approach explores very limited options for data insertion it has lower imperceptibility as compared to our technique. The SDMS technique yields stego images with an average PSNR value of 40.82 dB. These results are quite better results as compared to other steganography algorithms. The stego image produced by the SDMS technique and the SDMS technique with OPAP is shown in Fig. 19. The graph of the PSNR vs. GA iteration shown in Fig. 20 is of Pirate image. In Table 7, the SSIM value of stego images is shown for the proposed technique in comparison with LSB steganography at 3 bpp capacity. The improvement in PSNR value is in line with the improvement in SSIM value. This is true for all payload capacities and can be validated from Tables 1, 5 and 7.

3.4. Results of secret data extraction process

The extracted secret data is compared with the embedded data to confirm the proper working of the SDMS technique. The proposed steganography algorithm is a lossless and completely reversible method. However, to ensure the effectiveness of the extraction process we use the quantitative measure BER. The equation used for the calculation of BER is given below:

$$BER = \sum_{i=0}^{n} (S_i \oplus E_i)/n \tag{3}$$

In Eq. (3), S is the secret data image, E is the extracted data image, n is the total number of bits inserted and $\oplus$ characterizes the exclusive OR operation. To calculate BER, bit by bit ex-or operation is performed between embedded and extracted secret data [22]. The BER value is between the range 0 and 1. The BER value should be zero to ensure cent percent accuracy in the extraction process. The BER values of the extracted secret data from the respective cover image at various bpp payload capacities are shown in Table 8. The BER value for all the images is zero. Hence, we can conclude that the secret data extraction process works perfectly. There is no error in extracting secret data back from the stego image for all payload capacities.

4. Conclusion

A secret data modification based image steganography technique with a variable payload capacity using GA is proposed. The presented scheme provides high imperceptibility and payload capacity at the same time. GA is used to search the best value of the parameters to rearrange and modify the secret data. The proposed SDMS technique provides better results as compared to several state of the art steganography techniques. The stego images produced by the SDMS technique have significantly higher PSNR value as compared to other existing steganography algorithms. The stego images produced by the SDMS technique have at least 1 dB higher PSNR value as compared to other techniques for same payload capacities. A new concept of the flexible chromosome is also proposed; the flexible chromosome plays a significant role in improving the imperceptibility of the SDMS technique. The result of the SDMS technique with and without the use of flexible chromosome is also presented. Apart from standard test images few natural images were also used to test the performance of the proposed technique.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] A. Zenati, W. Ouarda, A. Alimi, SSDIS-BEM: a new signature steganography document image system based on beta elliptic modeling, Eng. Sci. Technol. Int. J. 23 (3) (2020) 470–482.
[2] A. Malik, G. Sikka, H.K. Verma, A high capacity text steganography scheme based on LZW compression and color coding, Eng. Sci. Technol. Int. J. 20 (1) (2017) 72–79.
[3] P. Shah, R. Bichkar, 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, 2018, pp. 119–129.

[4] R. Biswas, I. Mukherjee, S.K. Bandyopadhyay, Image feature based high capacity steganographic algorithm, Multim. Tools Appl. 78 (14) (2019) 20019–20036.
[5] M. Subhedar, V. Mankar, Current status and key issues in image steganography: a survey, Comp. Sci. Rev. 13 (2014) 95–113.
[6] M. Hussain, A. Wahab, Y. Idris, A. Ho, K. Jung, Image steganography in spatial domain: a survey, Signal Process. Image Commun. 65 (2018) 46–66.
[7] S. Mukherjee, G. Sanyal, A physical equation based image steganography with electro-magnetic embedding, Multim. Tools Appl. 78 (13) (2019) 18571–18593.
[8] A. Cheddad, J. Condell, K. Curran, P. Mc Kevitt, Digital image steganography: survey and analysis of current methods, Signal Process. 90 (3) (2010) 727–752.
[9] H. Yang, X. Sun, G. Sun, A high-capacity image data hiding scheme using adaptive LSB substitution, Radioengineering 18 (4) (2009) 509–516.
[10] K. Qazanfari, R. Safabakhsh, A new steganography method which preserves histogram: Generalization of LSB++, Inform. Sci. 277 (2014) 90–101.
[11] R. Amirtharajan, J.B. Balaguru Rayappan, An intelligent chaotic embedding approach to enhance stego-image quality, Inform. Sci. 193 (2012) 115–124.
[12] K. Muhammad, J. Ahmad, N. Rehman, Z. Jan, M. Sajjad, CISSKA-LSB: color image steganography using stego key-directed adaptive LSB substitution method, Multim. Tools Appl. 76 (6) (2017) 1–30.
[13] K. Muhammad, M. Sajjad, S. Baik, Dual-level security based cyclic18 steganographic method and its application for secure transmission of keyframes during wireless capsule endoscopy, J. Med. Syst. 40 (5) (2016) 114–129.
[14] K. Muhammad, J. Ahmad, S. Rho, S.W. Baik, Image steganography for authenticity of visual contents in social networks, Multim. Tools Appl. 76 (18) (2017) 18985–19004.
[15] S. Chakraborty, A.S. Jalal, C. Bhatnagar, Secret image sharing using grayscale payload decomposition and irreversible image steganography, J. Inform. Sec. Appl. 18 (4) (2013) 180–192.
[16] H. Tseng, H. Leng, High-payload block-based data hiding scheme using hybrid edge detector with minimal distortion, IET Image Process. 8 (11) (2014) 647–654.
[17] T.D. Nguyen, S. Arch-int, N. Arch-int, An adaptive multi bit-plane image steganography using block data-hiding, Multim. Tools Appl. 75 (14) (2016) 8319–8345.
[18] H.R. Kanan, 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) (2014) 6123–6130.
[19] A. Khan, A. Sarfaraz, Novel high-capacity robust and imperceptible image steganography scheme using multi-flipped permutations and frequency entropy matching method, Soft Comput. 23 (17) (2019) 8045–8056.
[20] S. Uma Maheswari, D. Jude Hemanth, Performance enhanced image steganography systems using transforms and optimization techniques, Multim. Tools Appl. 76 (1) (2017) 415–436.
[21] P. Shah, R. Bichkar, Imperceptible steganography scheme with high payload capacity using genetic algorithm and particle swarm optimization, Int. J. Eng. Adv. Technol. 9 (1) (2019) 917–923.
[22] P. Bedi, R. Bansal, P. Sehgal, Using PSO in a spatial domain based image hiding scheme with distortion tolerance, Comput. Elect. Eng. 39 (2) (2013) 640–654.
[23] A.H. Mohsin, A.N. Jasim, A.H. Shareef, A.A. Zaidan, B.B. Zaidan, O.S. Albahri, A.S. Albahri, M.A. Alsalem, K.I. Mohammed, S. Nidhal, N.S. Jalood, New method of image steganography based on particle swarm optimization algorithm in spatial domain for high embedding capacity, IEEE Access 7 (2019) 168994–169010.
[24] D. Jude Hemanth, J. Anitha, D.E. Popescu, L.H. Son, S. Patnaik, A modified genetic algorithm for performance improvement of transform based image steganography systems, J. Intell. Fuzzy Syst. 35 (1) (2018) 197–209.
[25] S. Pramanik, R.P. Singh, R. Ghosh, Application of bi-orthogonal wavelet transform and genetic algorithm in image steganography, Multim. Tools Appl. 79 (25-26) (2020) 17463–17482.
[26] E. Ghasemi, J. Shanbehzadeh, N. Fassihi, High capacity image steganography based on genetic algorithm and wavelet transform, in: Intelligent Control and Innovative Computing, Springer, (2012), 395–404.
[27] R. Biswas, S.K. Bandyapadhay, Random selection based GA optimization in 2D-DCT domain color image steganography, Multim. Tools Appl. 79 (11-12) (2020) 7101–7120.
[28] D. Goldberg, Genetic algorithms in search Optimizations and Machine Learning, Pearson Education India, 2006.
[29] Z. Abdmouleh, A. Gastli, L. Ben-Brahim, M. Haouari, N.A. Al-Emadi, Review of optimization techniques applied for the integration of distributed generation from renewable energy sources, Renew. Energy 113 (2017) 266–280.
[30] S. Farzin, M. Anaraki, Optimal construction of an open channel by considering different conditions and uncertainty: application of evolutionary methods, Eng. Optim. (2020).
[31] C.-K. Chan, L.M. Cheng, Hiding data in images by simple LSB substitution, Pattern Recog. 37 (3) (2004) 469–474.
[32] USC-SIPI Image Database, http://sipi.usc.edu/database/
[33] C.-C. Lin, W.-H. Tsai, Secret image sharing with steganography and authentication, J. Syst. Softw. 73 (3) (2004) 405–414.
[34] C.-N. Yang, T.-S. Chen, K.H. Yu, C.-C. Wang, Improvements of image sharing with steganography and authentication, J. Syst. Softw. 80 (7) (2007) 1070–1076.
[35] C.-C. Chang, Y.-P. Hsieh, C.-H. Lin, Sharing secrets in stego images with authentication, Pattern Recog. 41 (10) (2008) 3130–3137.
[36] C.-C. Wu, S.-J. Kao, M.-S. Hwang, A high quality image sharing with steganography and adaptive authentication scheme, J. Syst. Softw. 84 (12) (2011) 2196–2207.
[37] G.S. Yadav, A. Ojha, Hamiltonian path based image steganography scheme with improved imperceptibility and undetectability, Appl. Soft Comput. 73 (2018) 497–507.
[38] Z. Eslami, J.Z. Ahmadabadi, Secret image sharing with authentication-chaining and dynamic embedding, J. Syst. Softw. 84 (5) (2011) 803–809.
[39] Q. Wu, C. Zhu, J. Li, C. Chang, Z. Wang, A magic cube based information hiding scheme of large payload, J. Inform. Security Appl. 26 (17) (2016) 1–7.
[40] C.-N. Yang, S.-C. Hsu, C. Kim, Improving stego image quality in image interpolation based data hiding, Comput. Stand. Interfaces 50 (2017) 209–215.