High capacity image steganography method by using Particle Swarm Optimization

Shirin Shokranipour, Maryam Hasanzadeh*

Abstract— **The art of steganography is used to hide the relationship and secret messages between sender and receiver for the sake of information security in communication networks. Capacity, imperceptibility and robustness are three important pillars of steganography requirements. Increasing each of these factors in steganography may result in decreasing the other factors. Optimization methods with respect to an acceptable value for one factor can be used to increase the other factors. In this paper by specifying the scope of PSNR as a measure of imperceptibility and in order to increase capacity, steganography is conducted using PSO algorithm. In the proposed method considering the order of each bit-plane of cover image, steganography is run with a matrix encoding method. In the present study the capacity of stego images for famous cover images is examined. The results show that the proposed method in comparison with some recent ones provides better PSNR in addition to increasing capacity.**

Index Terms— **Steganography; Capacity;bit-plane; matrix encoding; PSO; PSNR.**

I. INTRODUCTION

Nowadays, in order to send secure data to different communication networks like internet, the steganography methods are required. Capacity, imperceptibility and robustness are major factors in steganography. Robustness in steganography means that after steganography, embedding messages in the cover image cannot be identified. Imperceptibility is the rate of changes in the appearance of a stego image, and capacity is the maximum amount of information that can be inserted in cover media without being identified [1]. Image steganography is done in spatial and transform domains. In the spatial domain-based techniques, secret data directly change the values of pixels of cover image. In this regard, the use of least significant bits is very common because it provides higher payload and low computational complexity [2]. LSB substitution [3], LSB matching-based [4], and Pixel Value Differencing (PVD) [5] are among the steganographic methods that use LSB of cover image in different ways in spatial domain. In order to send a stego image with better quality, various optimization methods have been proposed based on LSB bits. Of such methods, some are mentioned in what follows. Khodaei and Faez [6] used Genetic algorithm to find an optimal replacement for LSB in order to increase the imperceptibility. Wang RZ et al.[7] used GA for finding the desired matrices for transferring the blocks and value pixel of secret image to a new secret image based on comparing cover image with secret image. Bajaj R et al.[8].used Particle Swarm Optimization to obtain the best pixel positions in a gray level image imperceptibly where secret image pixel data can be hidden in without degrading the quality of the resultant image. In Bajaj R et al. data hiding scheme, PSO was used to embed a message in an image using LSB and to hide an image within another image using LSB technique. Punam Bedi et al.[9, 10] used PSO to find the best pixel locations in a gray cover image where the secret data can be embedded in such a way that both quality and robustness of the stego image are acceptable. These PSO-based techniques yielded better results than the techniques based on genetic algorithms and dynamic programming [10].

In the above optimization methods which are based on LSB bits, only a limited number of secret data bits can be hidden in cover image and as a result the stego image capacity is low. Therefore, by increasing the capacity of the stego image, it is possible to avoid sending several stego images. So the purpose of this paper is to propose an optimization-based method in steganography for increasing capacity by using most significant bits as well as the least significant bits of the cover image. In the proposed method, PSO algorithm is used by specifying an acceptable range of PSNR and for maximizing the capacity the secret data is hidden in bit-planes of cover image using matrix encoding method.

In this paper, steganography is conducted with the aim of increasing capacity and making minimum changes in cover image using matrix encoding method and PSO algorithm. This paper is organized as follows: The Matrix encoding method and PSO algorithm will be presented In Section 2 and 3. The proposed method for embedding and extracting will be described In Section 4, 5 and 6. The experimental results and analyses are illustrated in Section 7. The conclusion is presented in Section 8.

II. MATRIX ENCODING METHOD

Matrix coding was proposed by Crandall [11] to improve the embedding efficiency by decreasing the number of required bit changes. Of different steganographic methods, F5 algorithm, which was proposed by Westfeld [12], is the well-known and efficient one which implemented the matrix encoding. According to this method, n bits of the secret data are embedded in $k = 2^{\mathrm{n}} - 1$ bits of cover image with at most one change. In matrix encoding method, between bits of cover image, the XOR operation is done to hide the secret data in. The block C of cover

Manuscript received April 13, 2016; accepted October 18, 2016

Department of Computer Engineering, Shahed University, Tehran, Iran.

image with size $2^n - 1$ are in the form of vector C = c_1 c_2 …….c_k with the length of k. N is the bit length of secret data S = s_1 s_2 $_s$_ _ _s_n to be embedded into one block. The function f is defined as formula (1) to map k bits cover data c into n bits binary string. ⊕ is the sign for XOR operation.

$$f: f(c) = \oplus_{i=1}^{k} c_i \,.\, i \tag{1}$$

Where c_i denotes the i-th position of block c. i is the corresponding number of the bit position and is in binary form during the operation. The bit length of binary i is selected as the same size as secret data s. Subsequently, implement XOR operation on the value of function $f(c)$ and secret data S as formula (2) to obtain a decimal numberα.

$$\alpha = f(c) \oplus S \tag{2}$$

Finally, an embedded stego data C' is generated by logically flipping the α -th bit position of block c as formula (3). In special cases, the carrier block c will be left intact when α = 0.

$$C' = \begin{Bmatrix} c & if\ \alpha = 0 \\ c_1\, c_2 c_3 \neg c_\alpha \ldots \ldots \ldots c_k & if\ \alpha \neq 0 \end{Bmatrix} \tag{3}$$

In example1, 2 message bits s_1 s_2 are hidden in 3 bits c_1c_2 c_3 of cover image.

$s_1 = c_1 \oplus c_3$, $s_2 = c_2 \oplus c_3 \Rightarrow$ change no

$s_1 = c_1 \oplus c_3$, $s_2 \neq c_2 \oplus c_3 \Rightarrow$ change c_2

$s_1 \neq c_1 \oplus c_3$, $s_2 = c_2 \oplus c_3 \Rightarrow$ change c_1

$s_1 \neq c_1 \oplus c_3$, $s_2 \neq c_2 \oplus c_3 \Rightarrow$ change c_3

Example1. Embedding 2 bits of secret data in 3 bits of cover image

One of the important performance measure of steganographic algorithm is called the embedding rate R defined as formula (4):

$$R\ (n) = \frac{n}{2^n - 1}\ . \tag{4}$$

III. Particle swarm optimization algorithm

The PSO algorithm was introduced by Kennedy and Eberhart in 1995[13] which had drawn a significant amount of research interests because of its high speed of convergence and global search ability. Particle Swarm Optimization (PSO) is a population-based stochastic optimization technique inspired by social behavior of bird flocking or fish schooling. In bird flocking the flight of each individual is influenced by its own experience and its companion. The particles express the ability of fast convergence to local and/or global optimal positions over a small number of generations. All of the particles iteratively discover the probable solution. A swarm in PSO consists of a number of particles. Population has m particles. Each particle is a potential solution in d-dimensional space. Each particle has a position and velocity according to (5),(6).

$$X_i = (X_{i1}, X_{i2}, ...,X_{id}) \tag{5}$$

$$V_i = (V_{i1}, V_{i2}, ...,V_{id}) \tag{6}$$

i= 1, 2, . . , m.

Each particle generates a position according to the new velocity and the previous positions of the cell, and is compared with the best position which is generated by previous particles in the cost function and keeps the best one as Gbest. Each particle accelerates the directions of not only the local best solution but also the global best position. If a particle discovers a new probable solution, other particles will move closer to it so as to explore the region more completely in the process. In general, there are three attributes, current position x, current velocity v, and local best position Pbest, for particles in the search space to present their features. At first, $Pbest_i$ is equal to X_i as Formula (7).

$$Pbest_{\,i} = (Pbest_{\,i1}, Pbest_{\,i2}, ..., Pbest_{\,id}) \tag{7}$$

Each particle in the swarm is iteratively updated according to the aforementioned attributes. The new velocity of every particle is updated by (8).

$$v = w \times v + c_1\, r_1\, (Pbest - X) + c_2\, r_2\, (Gbest - X) \tag{8}$$

Where w is the inertia weight of velocity, $c_1$and c_2 denote the acceleration coefficients, r_1 and r_2 are elements from two uniform random sequences in the range (0, 1), and the new position of the particle is calculated as follows:

$$x = x + v \tag{9}$$

The algorithm is terminated when one of the following occurs:
1. Maximum number of iterations has been reached.
2. An acceptable solution has been found.
3. No improvement is observed over a number of iterations.

IV. Embedding secret data

The embedding secret data in the proposed method consists of the following three steps and will be described in part 4.1, 4.2, and 4.3:

1. Blocking bit-planes
2. Embedding secret bits in blocks
3. Modifying the pixel value.

A. Blocking bit plane

In this paper for 8-bit pixel cover image, 8 bit-planes are considered. The eighth bit-plane of a cover image includes the most significant bits and the first bit-plane includes the least significant bits.

B. Embedding secret bits in blocks

In the present paper, the matrix encoding method in blocks of each bit-plane is used. According to this method, n bits are embedded in $2^n - 1$ bits and only a single bit is changed. According to the considered n for a bit-plane, the bit-plane splits to blocks with the size of $2^n - 1$ and secret bits embed in each block separately with one bit changing in each block.
Selecting the value of n as the number of secret data bits in this steganography method is very important and has a considerable effect on the capacity and PSNR of the stego image. Accordingly, for low-order bit-planes for hiding more secret data bits, a smaller n is selected and for high-order bit-planes, a

larger n is selected. For 8-bit pixel cover image, 8 “n”s are put next to each other to form a code which represents the number of bits that can be hidden in blocks of each bit-plane of cover image. It should be noted that for creating bits plane, we consider the image pixels in a column. In this study, steganography is done on 8-bit pixels cover image with size 512 × 512. For example, Code [12, 11, 8, 5, 4, 2, 2, 2] represents the values of n for 8 bit-planes. According to this code, bit-plane eight splits to 64 blocks by calculating $2^{12} - 1$ and 12 bits of secret data embedded in each block.

C. Modifying the pixel value

In this paper, steganography is started from bit-plane 8 to 1 consecutively because changes in most significant bits further reduces the imperceptibility of stego image, so it is necessary that after embedding secret data in most significant bits further modification be done. After embedding secret data in each block, the pixel which has changed in each block is detected and the pixel value is modified in this step. In order to reduce the difference between pixel values before and after steganography, the pixel value is modified. If the pixel value increases, the value of bits having a lower order than the changed bit must be flipped to zero and if the pixel value decreases, the bits with lower order must be flipped to one. In this case the difference between pixel values before and after steganography becomes least. Forasmuch as changes in most significant bits further reduce the imperceptibility of stego image, it is necessary to modify more. The example in Figure 3 shows that the bit with position 5 has changed from zero to one after steganography; so for modifying, the bits with position 0-4 are changed to zero.

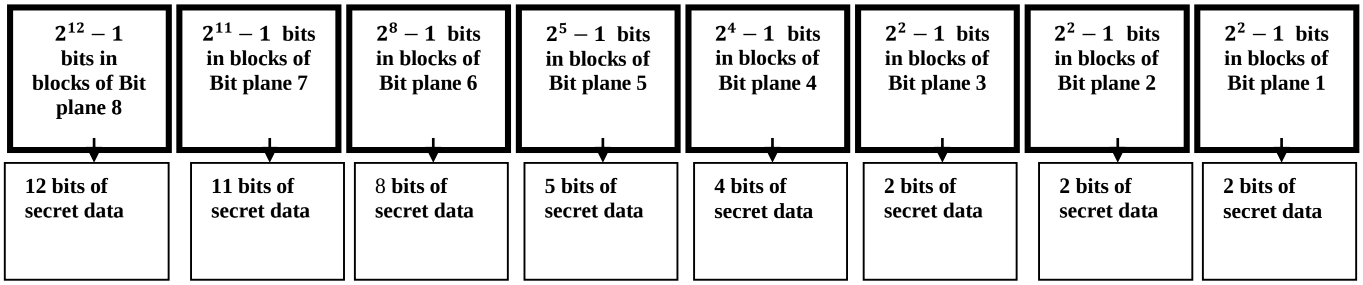
$2^{12} - 1$ bits in blocks of Bit plane 8
$2^{11} - 1$ bits in blocks of Bit plane 7
$2^{8} - 1$ bits in blocks of Bit plane 6
$2^{5} - 1$ bits in blocks of Bit plane 5
$2^{4} - 1$ bits in blocks of Bit plane 4
$2^{2} - 1$ bits in blocks of Bit plane 3
$2^{2} - 1$ bits in blocks of Bit plane 2
$2^{2} - 1$ bits in blocks of Bit plane 1
12 bits of secret data
11 bits of secret data
8 bits of secret data
5 bits of secret data
4 bits of secret data
2 bits of secret data
2 bits of secret data
2 bits of secret data

Fig 1- Steganography in step 2 for Code [12, 11, 8, 5, 4, 2, 2, 2]

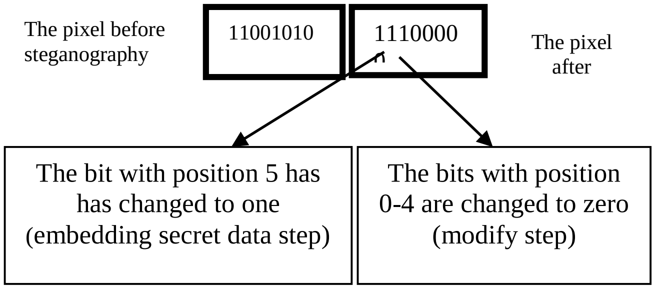
The pixel before steganography
11001010
1110000
The pixel after
The bit with position 5 has has changed to one (embedding secret data step)
The bits with position 0-4 are changed to zero (modify step)

Fig 2- The pixel before steganography and after modifying

V. THE PROPOSED PSO ALGORITHM

The details of proposed PSO algorithm are described in 6 steps as below:

Step 1: It is required to initialize the particle swarm and the parameters of the PSO algorithm (such as population size, generations, etc.). Initializing the population is the most important step in PSO algorithm. In this paper, cover image with eight-bit pixels, which includes eight bit-planes, is considered for steganography. According to the number of secret data bits, any number of bit-planes can be used in steganography. In this paper all 8 bit-planes are considered. Each member is a code which is based on matrix encoding method and consists of 8 “n”s related to each bit-plane. As a result, each dimension of code is a decimal number that corresponds to each bit-plane and is determined according to its order. Moreover, 20 members are intended and selection of the members are based on their PSNR that they have minimum PSNR of 30. The PSNR value of more than or equal 30 dB is acceptable because of not being understood by humans and it is indicative of the imperceptibility of stego image [14]. The velocities are selected based on the order of each bit-plane. The amount of the initial parameters C and W are c1, 2=1.49, w=1.79.

[12, 11, 8, 5, 4, 2, 2, 2] is an example of a member of population. First, the pbest is initialized equal to position of the members. Afterwards there is an update.

Step 2: Embedding secret data is performed with 3 steps according to part 2.

Step 3: In this step the fitness value of each particle is calculated according to (10). After embedding secret data operations for individual members, the capacity of each member is calculated. The member who has the greatest capacity is gbest.

$$\text{Fitness (capacity)} = \sum_{i=1}^{8} \text{Number of blocks}(i) \times \text{n}(i) \qquad (10)$$

where i denotes the i-th bit-plane of cover image.

Step 4: The velocity and position of each particle needs updating according to (8) and (9) in part 1.2. Embedding secret data again are performed as in step 2 for the new position and the fitness value of each particle is calculated according to (10).

Step 5: The fitness of each new member is compared with the fitness of pbest; if the new fitness value is greater, then the Pbest value changes for the new members.

Step 6: The personal best position (Pbest) and the global best position (Gbest) for the population should be updated. Here the fitness value of each member ought to be updated as well.

Step 7: If termination condition is satisfied, Gbest=X and the algorithm stops. Otherwise, go to step 4. The process of algorithm in different stages is shown in Figure 3.

Some important points in the implementation of the proposed algorithm are as follows:

- Selection of the velocity matrix is not random. Since in steganography most significant bits should not be changed much, velocity values are deemed to be less. Nevertheless, as the value of the bits in the pixel decreases, velocity values can increase.
- The initial population values are discrete.
- The inertial weight (w) and other parameters' values of the PSO algorithm are based on standard algorithm. According to the discrete values of the initial population, the velocity values are rounded after calculating the velocity.
- The best fitness has the maximum capacity for the member which has PSNR>= 30.

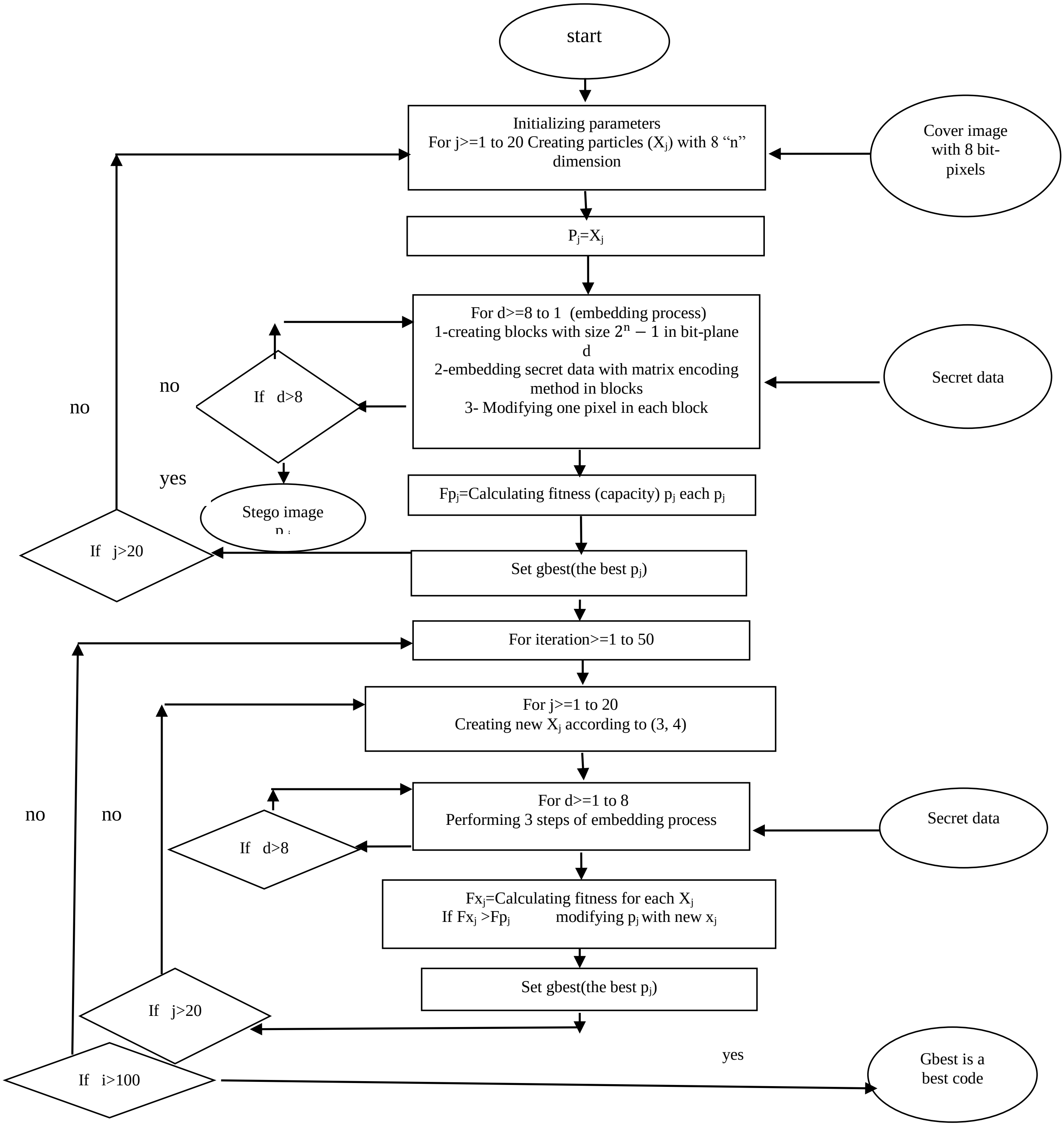
start
Initializing parameters
For j>=1 to 20 Creating particles (X_j) with 8 "n" dimension
Cover image with 8 bit-pixels
P_j=X_j
For d>=8 to 1 (embedding process)
1-creating blocks with size 2ⁿ − 1 in bit-plane d
2-embedding secret data with matrix encoding method in blocks
3- Modifying one pixel in each block
Secret data
no
If d>8
no
yes
Stego image
Fp_j=Calculating fitness (capacity) p_j each p_j
If j>20
Set gbest(the best p_j)
For iteration>=1 to 50
For j>=1 to 20
Creating new X_j according to (3, 4)
For d>=1 to 8
Performing 3 steps of embedding process
Secret data
no
no
If d>8
Fx_j=Calculating fitness for each X_j
If Fx_j >Fp_j modifying p_j with new x_j
Set gbest(the best p_j)
If j>20
yes
If i>100
Gbest is a best code

Fig 3-The process of algorithm in different stages

VI. EXTRACTING SECRET DATA

Extracting secret data is based on the code. The code is obtained from 8 “n”s of each bit-plane being next to each other. According to embedding secret data part, which is based on matrix encoding method, n bits of secret data are hidden in each block of 8 bit planes. For the 8-bit image pixels, the code is 8 decimal digits. This code add to the number of bits of secret data can be sent as a small encrypted file with stego image or can be embedded as a binary code in a specified part of stego image. The code specifies the receiver as to how to extract the secret message. Extracting the block S= s_1 s_2 $_s$_ _ _s_n of secret data is done in blocks of each bit-plane separately according to the matrix encoding method as formula (9). The results of XOR operations in blocks of 8 bit-planes specify the secret data bits.

$$S = \bigoplus_{i=1}^{k} C'_i \,.\, i \qquad (9)$$

Where C'_i denotes the i-th position of block
$C'= C'_1 C'_2 \ldots \ldots \ldots \ldots C'_k$ of stego image.
The following extracting secret message is described by an example. According to [12, 11, 8, 5, 4, 2, 2, 2] code, the receiver should extract 12 bits of secret data from each block with size $2^{12} - 1$ of bit-planes 8. Assuming an image with size 512×512 for steganography, 64 blocks with size 4095 bits are created from each of which 12 bits of secret data are extracted. In the above code, the blocks of least significant bit-planes 3, 2 and 1 consist of 3 bits from each of which 2 bits of the secret data can be extracted. Because steganography was started from the 8th bit-plane, the recipient must start extracting from blocks of bit-plane 8 and in order to extract the whole data it must continue to bit-plane 1 sequentially. Extracting the secret data from stego image is illustrated in Figure 6. Example 2 shows extracting two bits $s_1 s_2$ of secret data from three bits $C'_1 C'_2 C'_3$ of stego image.
$s_1 = C'_1 \oplus C'_3$, $s_2 = C'_2 \oplus C'_3$. Example2- extracting two bits

s_1 s_2

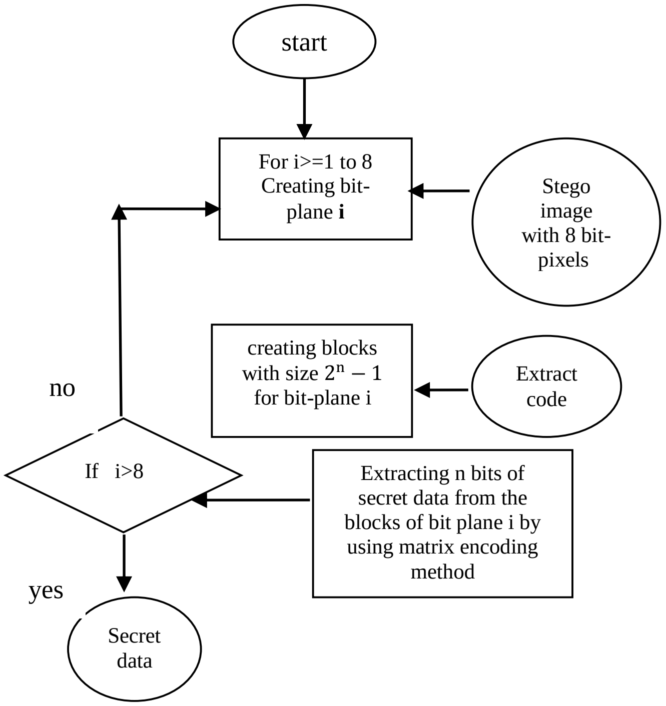
start
For i>=1 to 8
Creating bit-plane **i**
Stego image with 8 bit-pixels
creating blocks with size $2^n - 1$ for bit-plane i
Extract code
no
If i>8
Extracting n bits of secret data from the blocks of bit plane i by using matrix encoding method
yes
Secret data

Fig 4-The process of Extracting in different stages

VII. EXPERIMENTAL RESULTS

The algorithm was implemented as a MATLAB program running on a computer with 2.00 GHz CPU and 2.00 Gb RAM. The experiments were tested on 512×512 gray images as shown in Figure 5, and the cover image and secret data were generated randomly. The capacity was calculated to obtain the amount of bits embedded in the cover image. The peak signal-to-noise ratio (PSNR) was used to measure the visual image quality. For an H × W grayscale image, the PSNR value is defined as follows:
PSNR = 10log10 [2552/MSE] (5)

$$\text{MSE} = \frac{1}{\text{H}\times\text{W}} \sum_{\text{i}=1}^{\text{H}} \sum_{\text{j}=1}^{\text{W}} (\text{I}_{\text{ij}} - \text{I}'_{\text{ij}})^2 \qquad (6)$$

Where I_{ij} and I'_{ij} denote the pixel values in row i and column j of the cover image and the stego image, respectively.

Fig 5- cover images Baboon, jet, Lena, Elaine with size 512×512

Table 1 shows the best capacity after performing optimization algorithm and the Gbest code whit PSNR> = 30 for the four cover images shown in Figure 5. The results show the images, jet, and lena have the maximum capacity; however, the lena in imperceptibility has higher PSNR. Baboon image also has the highest PSNR value. The four selected cover images in other methods are used as the cover image in this paper. Converging results are obtained in this table after 30 to 50 runs with 20 members’ as initial population. Figure 6 shows converging results after 30 runs for lena cover image.

Table I
THE GBEST CODE FOR PSNR>=30

Cover images	**PSNR (db)**	**Capacity (bits)**	**Gbest code**
Elaine	30.31	853711	[18,18,5,3,2,2,2,2]
Jet	30.01	878463	[18,18,9,2,2,2,2,2]
Lena	30.09	878463	[18,18,9,2,2,2,2,2]
Baboon	30.55	836397	[18,18,6,3,2,2,2,2]

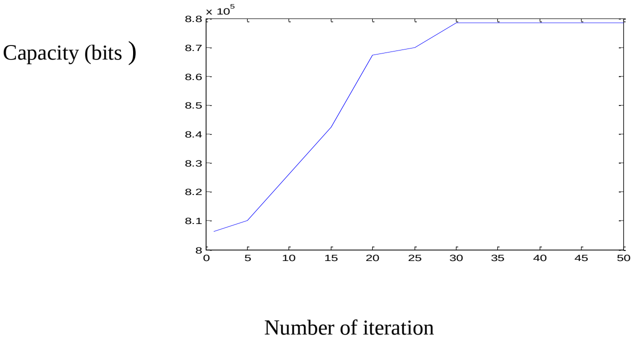

Fig 6- converging results after 30 runs for lena cover image

Fig7, which is based on Table 1, show cover images and stego images for jet, lena with the maximum capacity 878463 and Baboon image with the highest PSNR 30.55.

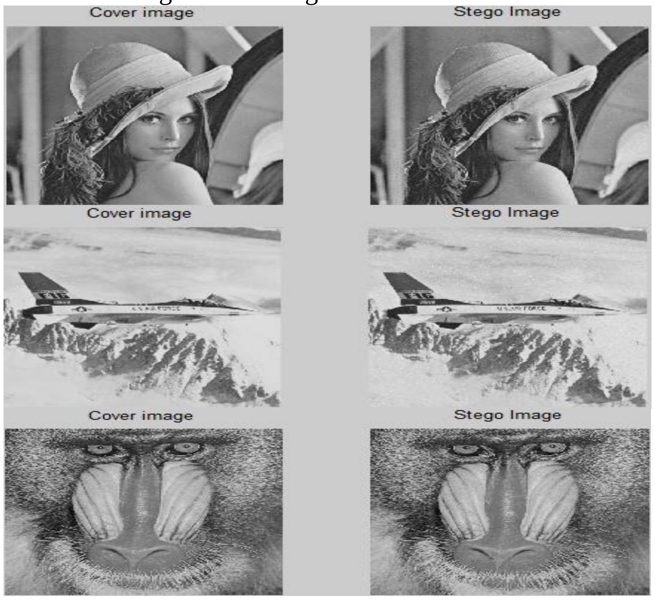

Fig 7-cover images and stego images for lena, jet and baboon

In addition to comparing the proposed method with the matrix encoding method, the performance of the proposed method has been evaluated and compared with other spatial domain data hiding techniques which used optimization algorithm like Khodai and Faez's technique [6] based on genetic algorithms, Chang et al.'s technique [15] based on dynamic programming and Punam Bedi et al.'s Method [10] based on PSO. In Punam Bedi et al.'s method the cover image is divided into a number of blocks with size 8 * 8. The embedding technique finds optimum locations of blocks, using PSO. The fitness function used by the PSO module is based on the quality and distortion tolerance of the stego image. The comparisons of the results in capacity and PSNR of the proposed method and above mentioned methods are shown in Table 2.

TABLEII
THE COMPARISONS OF THE RESULTS OF PROPOSED METHOD WITH OTHER METHODS

Cover images	Lena		Jet		Baboon		Elaine	
Methods	PSNR	capacity	PSNR	Capacity	PSNR	Capacity	PSNR	Capacity
1-R. Crandall's Method[13]	52.09	112722	52.07	112722	52.10	112722	52.10	112722
2-Khodaei and Faez [6] method	42.87	131072	42.92	131072	42.71	131072	42.79	131072
3-Chang et al. [9] Method	43.24	131072	43.20	131072	42.75	131072	43.19	131072
4-Punam Bedi et al. [12] Method	45.19	131072	45.28	131072	44.31	131072	45.93	131072
Proposed method	54.49	141558	53.90	141558	53.98	141558	54.40	141558
	53.78	174762	53.21	174762	53.26	174762	53.73	174762
	47.52	349524	46.95	349524	46.98	349524	47.42	349524
	43.74	461373	43.21	461373	43.20	461373	43.68	461373

Reviewing Table 2, it can be conducted that the proposed method, unlike other methods, don't have the limitations in capacity and although it embeds 3 to 4 times more secret data bits in cover image, and PSNR is still better than other methods. The methods in Table 2 have done steganography in least significant bits of cover image while in the proposed method steganography is started with low order bit-planes but by increasing the number of secret data bits, higher-order bit-planes are used. Of different cover images in Table 2, lena image has the best results in PSNR. Figure 8 shows the result of the proposed method in comparison with other methods for lena image. As it can be seen in Figure 8, the proposed method in comparison with the resent methods is better in terms of both capacity and PSNR and this trend is true for other cover images as well. The lena image is selected as an instance because of the better results in PSNR value than others.

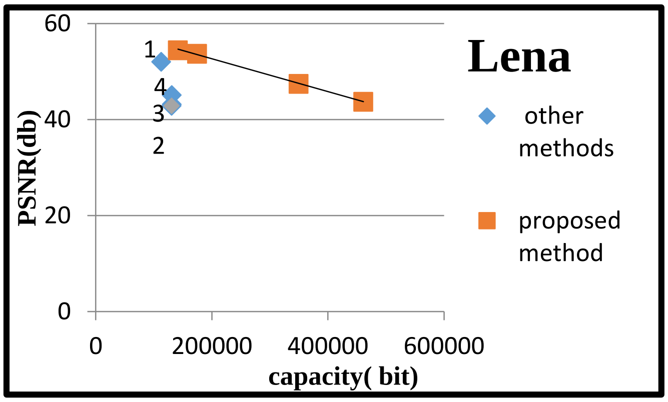

Fig 8- The comparison chart of proposed method with other methods for Lena image

VIII. Conclusions

In the proposed method by considering the order of each bit-plane, the entire bit-planes of cover image can be used with small changes. In the proposed optimization algorithm, to determine the scope of appropriate PSNR we can embed intended secret data bits in the bit-planes of cover image. Obtained result indicates higher PSNR and capacity of the proposed method in comparison with other methods. Because the selection of the block size for bit-planes are wider, it is necessary to use the proposed method for cover images with 16- or 32-bit pixel. The proposed method can be used in gray scale or color cover images in spatial and transform domains. Different steganographic methods, which are used the PSO algorithm, result in different implementations and different selection of initial population and the velocities. In future in the area of cover selection, the framework of the proposed method can be used and by selection of different cover images as members of the population, it is possible to examine the increase of the capacity or PSNR.

References

[1] Sajedi, H., "high capacity image steganography methods", Thesis in computer engineering, sanati sharif., pp. 156 (2010).

[2] Tang, M., J. Hu, and W. Song, "A high capacity image steganography using multi-layer embedding", Optik-International Journal for Light and Electron Optics., 125(15), pp. 3972-3976 (2014).

[3] Chang, C.-C., M.-H. Lin, and Y.-C. Hu, "A fast and secure image hiding scheme based on LSB substitution", International Journal of Pattern Recognition and Artificial Intelligence., 16(04), pp. 399-416 (2002).

[4] Mielikainen, J., "LSB matching revisited", Signal Processing Letters., IEEE, 13(5), pp. 285-287 (2006).

[5] Wu, D.-C. and W.-H. Tsai," A steganographic method for images by pixel-value differencing", Pattern Recognition Letters., 24(9), pp. 1613-1626 (2003).

[6] Khodaei, M. and K. Faez, "Image hiding by using genetic algorithm and LSB substitution", in Image and Signal Processing, Springer., pp. 404-411 (2010).

[7] Wang, R.-Z., C.-F. Lin, and J.-C. Lin, "Image hiding by optimal LSB substitution and genetic algorithm", Pattern recognition., 34(3), pp. 671-683 (2001).

[8] Bajaj, R., P. Bedi, and S. Pal, "Best hiding capacity scheme for variable length messages using particle swarm optimization", in Swarm, Evolutionary, and Memetic Computing, Springer., pp. 230-237 (2010).

[9] Bedi, P., R. Bansal, and P. Sehgal, "Using PSO in image hiding scheme based on LSB substitution", in Advances in Computing and Communications Springer., pp. 259-268 (2011).

[10] Bedi, P., R. Bansal, and P. Sehgal, "Using PSO in a spatial domain based image hiding scheme with distortion tolerance". Computers & Electrical Engineering., 39(2), pp. 640-654 (2013).

[11] Crandall, R., Some notes on steganography. Posted on steganography mailing list, 1998.

[12] Andreas, W., "F5—A Steganographic Algorithm", Pro-ceedings of the 4th International Workshop on Information Hiding, Pittsburgh., April pp. 289-302 (2001).

[13] Eberhart, R.C. and J. Kennedy. "A new optimizer using particle swarm theory", in Proceedings of the sixth international symposium on micro machine and human science., New York, NY (1995).

[14] Yu, Y.-H., C.-C. Chang, and I.-C. Lin, A new steganographic method for color and grayscale image hiding. Computer Vision and Image Understanding, 2007. 107(3): p. 183-194.

[15] Chang, C.-C., C.-S. Chan, and Y.-H. Fan, "Image hiding scheme with modulus function and dynamic programming strategy on partitioned pixels", Pattern Recognition., 39(6), pp. 1155-1167 (2006).