Minimizing Turtle-Shell Matrix Based Stego Image Distortion Using Particle Swarm Optimization

Qiang Jin[1], Zhihong Li[1], Chin-Chen Chang[2], Anhong Wang[1], and Li Liu[1]
(Corresponding author: Chin-Chen Chang)

Institute of Digital Multimedia and Communication[1]
Taiyuan University of Science and Technology, Taiyuan 030024, China
Department of Information Engineering and Computer Science[2]
Feng Chia University, Taichung 40724, Taiwan
No. 100, Wenhwa Rd., Seatwen, Taichung 40724, Taiwan
(Email: alan3c@gmail.com)
(Received Nov. 19, 2015; revised and accepted Feb. 16 & Mar 6, 2016)

Abstract

The purpose of data hiding is to embed secret messages into cover media so that the receiver will not be aware of the existence of these important messages. Recently, Chang et al. proposed a novel data hiding scheme based on turtle shells, which provided a good visual quality and an acceptable embedding capacity. However, the matrix that is used in the scheme based on turtle shells is not an optimal matrix. This means that the distortion of the image can be reduced further. In this paper, we propose a method that uses particle swarm optimization (PSO) to further reduce the distortion of the cover image based on turtle shells. Our experimental results confirmed that our scheme was efficient in finding an optimal replacement table and achieved a better visual quality of the cover image. Also, our scheme enhanced the security of scheme based on the turtle-shell matrix.

Keywords: Data hiding, GA, PSO, turtle shell

1 Introduction

With the rapid development and extensive application of the Internet, information security has been recognized as an important issue by many users. As a result, data encryption and data hiding, the two key techniques in the field of information security, also have received greater interest as topics of concern and research. Data encryption [3, 13] can alter a piece of clear text, or unencrypted information, into cipher text, or encrypted information. However, assailants are attracted easily by multimedia information after encrypting, and its content is totally transparent once the cipher-text is decrypted. Different from data encryption, data hiding is a technique that hides secret information in an image to make the secret data inaudible or invisible for assailants. Due to this advantage, it is used extensively in many fields, such as electronic commerce, copyright protection, and image authentication.

Generally, date hiding techniques are conducted mainly in three domains, i.e., the compression domain, the frequency domain and the spatial domain. In the compression domain, vector quantization (VQ) [5, 8, 9, 12] is often used in data hiding schemes to obtain more space for embedding secret information. In 2003, Du and Hsu [9] proposed an adaptive algorithm to embed secret data into VQ compressed images. Later, Hu [12] and Chang et al. [8] improved the embedding capacity of VQ compressed images to embed secret data. Although these compression-domain data hiding schemes achieved higher hiding capability, the reduction in the quality of the images was a serious concern.

In the frequency domain, first, cover images are transformed by using a frequency-oriented mechanism such as the discrete cosine transformation (DCT) [6, 16], or the discrete wavelet transformation (DWT) [1, 17]. Then the secret data are embedded into the transformed coefficients. Although these data hiding schemes can obtain the stego images that have better visual quality, the embedding capacity of secret data still is not satisfactory because only a small number of coefficients are used to embed the secret data.

In the spatial domain, secret messages are embedded directly into the pixels to ensure a high embedding capacity of secret data. The most common method is least significant bit (LSB) replacement. In the traditional LSB method [2], the LSB of the cover image is replaced directly by a secret bit. In 2001, Wang [22] proposed a data hiding

method by using optimal LSB substitution and genetic algorithm to improve the visual quality of stego images. To further improve the hiding capacity and visual quality of stego images, Mielikainen [20] proposed an LSB matching scheme that uses a pair of pixels as a unit to embed two secret bits. However, the direction of modification on the cover pixels is exploited incompletely for data hiding in this scheme. In 2006, Zhang and Wang [23] proposed a data hiding scheme that fully exploits the modification directions (EMD) to overcome this problem. In 2008, hang et al. [4] proposed Sudoku solutions to guide the modification of pixels for data hiding to make the data more secure than [23] and [4]. Later, in order to improve the visual quality of Chang et al.'s method, Hong et al. [10] proposed the Sudoku-S scheme based on a search algorithm. In 2010, Kim et al. [15] proposed two new schemes, named EMD-2 and 2-EMD to enhance the embedding capacity of the original EMD scheme. However, these schemes offered an unacceptable trade-off between low image quality and restricted embedding capacity.

Recently, Chang et al. [7] proposed a novel data hiding scheme to overcome this problem by using turtle shells for guiding the modification of cover pixels. This scheme uses a hexagon, which is called a turtle shell, to contain eight different digits, ranging from 0 to 7. A matrix composed of a large number of turtle shells is used to alter the values of pairs of cover pixels for the purpose of hiding secret bits. This scheme achieves a hiding capacity of 1.5 bits per pixel and a peak signal-to-noise ratio (PSNR) value of about 49 dB. Later, Liu et al. [18] enhanced the embedding capacity by using the location table to develop the turtle shell-based scheme. However, the turtle-shell matrix in their schemes is not optimal, meaning that the distortion of the image can be reduced further. There are some well-known evolutionary algorithms that can be used to solve the optimization problem, such as genetic algorithm (GA) [11] and particle swarm optimization (PSO) [19]. Compared with GA, the main advantage of PSO is its high computational efficiency. In this paper, we propose a method that involves using PSO to minimize the distortion of the cover image based on turtle shells. In our method, first, an original matrix is generated according to Chang et al.'s scheme [7]. Then PSO is used to search the optimal replaceable table of the eight integers in the matrix. After a near optimal table is developed, a new matrix is generated by replacing all of the eight integers with integers from the near optimal table. Finally, the data hiding procedure is conducted according to the new matrix in the same way as was done in Chang et al.'s scheme. The experimental results confirmed that our scheme achieved smaller image distortion than the scheme in [7].

The rest of this paper is organized as follows. The two important techniques are reviewed in Section 2, and the proposed scheme is presented in detail in Section 3. The experimental results are discussed in Section 4, and our conclusions are presented in Section 5.

2 Related Work

2.1 A Novel Scheme for Data Hiding Based on A Turtle-shell Matrix

A matrix based on turtle shells is constructed before the data hiding process. The matrix only contains 8 integers ranging from 0 to 7. The first row of the matrix is set from 0 to 7 in a circular permutation, and the other rows obey two rules. The first rule is that the difference between two adjacent values in the same row is set to 1. The second rule is that the difference between two adjacent values in the same column is set alternately to 2 and 3. The matrix is shown in Figure 1.

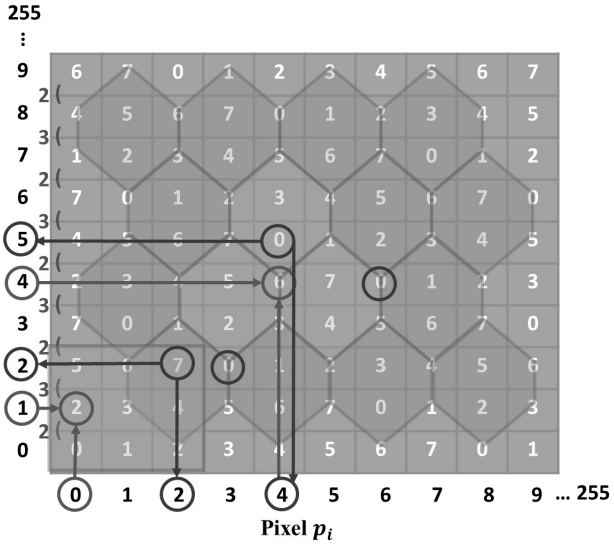

Figure 1: An example of the matrix based on turtle shells

The matrix here is used to hide secret data in order to obtain the minimum distortion between the original image and the stego image. The details of embedding procedure are described as follows.

Step 1. Divide the original image into non-overlapping pixel pairs, each of which contains two adjacent pixels (p_i,p_{i+1}).

Step 2. Generate a random binary bit stream as the secret data and convert each three bits into an octal digit, ranging from 0 to 7.

Step 3. Choose a pixel pair (p_i,p_{i+1}) as the abscissa and ordinate value of the matrix, respectively. Find a corresponding extraction value in the matrix.

Step 4. If the extraction value is equal to the secret digit that is to be embedded, the value of this pixel pair will not be changed. Otherwise, the next step will be implemented.

Step 5. If the extraction value is not equal to the secret digit that is to be embedded, the following two dif-

ferent cases are considered according to the position of the extraction value in the turtle shell.

Case 1: If the extraction value is on the back of the turtle shell, the value that is equal to the secret digit that is to be embedded can be searched within this turtle shell. And the corresponding abscissa and ordinate of this value are used to substitute for the cover pixel pair (p_i,p_{i+1}). However, if the extraction value is on the edge of the turtle shell, the searching range of the value that is equal to the secret digit will be the collection of the turtle shell that contains it.

Case 2: If the extraction value does not involve in any turtle shell, the searching range is a 3×3 sub-block on which this pixel pair is located. Any 3×3 sub-block also contains 8 different values from 0 to 7 because of the architectural property of the matrix.

Step 6. Repeat Steps 3 through 5 until all of the pixel pairs have been processed to embed the secret digit. In this way, a stego image is generated.

For example, suppose that two secret digits, i.e., 0 and 7, are to be embedded into two pixel pairs, i.e., (4, 4) and (0, 1), respectively. Figure 1 shows an example of the matrix based on turtle shells. The corresponding extraction value of pixel pair (4, 4) is 6, which, obviously, it is not equal to the secret digit 0 that is to be embedded. Because this extraction value is on the edge of three turtle shells, the closest 0 is searched within the three turtle shells with the red background in Figure 1. Then the pixel pair (4, 5) is taken as the new pixel pair to substitute for the original pixel pair. In the same way, the extraction value of the pixel pair (0, 1) does not involve in any turtle shell, so the searching range is a 3×3 sub-block with the red background in Figure 1. For this pair, the secret digit is 7 which is located in (2, 2). This pixel pair is replaced by (2, 2).

In the process of extracting secret data, taking each pixel pair of the stego image as a coordinate of the matrix, a corresponding extraction value in the matrix can be determined. It is the embedded secret digit. By converting each extraction value into binary bits in order, the original secret data can be recovered.

2.2 Particle Swarm Optimization

PSO, which was proposed by Kennedy and Eberhart in 1995 [19], is a global search algorithm. It was developed based on the social behavior of flocks of birds and schools of fish when searching for food. This technique has been used successfully in many fields as an optimization tool, such as estimating the distribution state [21] and dispatching reactive power [24].

In PSO, each individual of the population is called a particle, and it flies around in a multi-dimensional search space in order to find the optimal solution. The position of a particle represents a candidate solution to the optimization problem at hand. Each particle searches for a better position in the search space by changing its velocity according to the rules that were inspired originally by behavioral models of flocks of birds. The position of the particle is updated as follows [19]:

$$x_i(t+1) = x_i(t) + v_i(t+1), \tag{1}$$

where $x_i(t+1)$ and $x_i(t)$ are two parameters that represent the updated position and the current position, respectively. The velocity of this particle is updated according to the following equation:

$$\begin{aligned} v_i(t+1) \quad = \quad & wv_i(t) + c_1r_1(p_{best} - x_i(t)) \\ & + c_2r_2(g_{best} - x_i(t)), \end{aligned} \tag{2}$$

where c_1 and c_2 are two acceleration coefficients, w is an inertia factor, r_1 and r_2 are two independent random numbers that are distributed uniformly in the range of [0,1], and p_{best} and g_{best} are the local best particle and the global best particle, respectively.

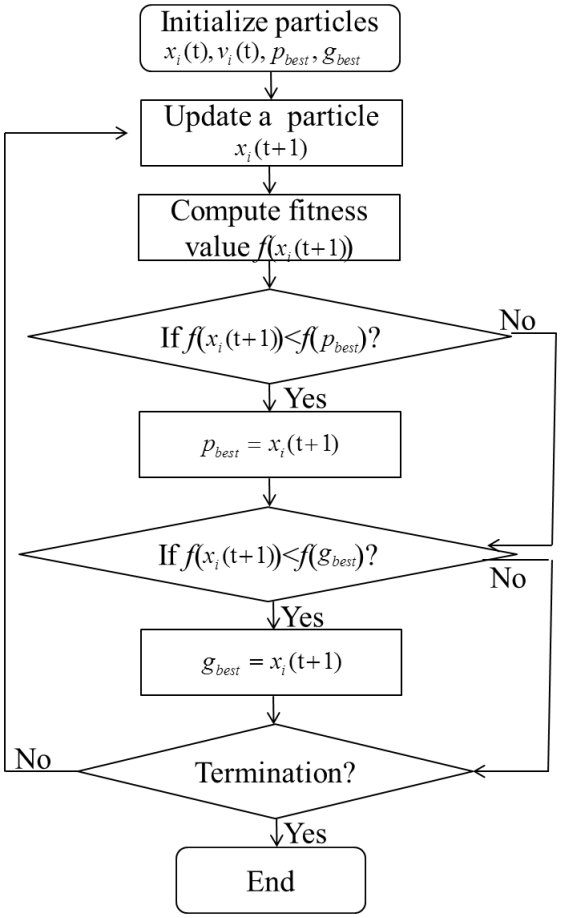

Figure 2: The flowchart of PSO

Figure 2 shows the basic flowchart of PSO, and the process of PSO is summarized as follows:

Step 1. Initialize a population of particles with random positions $x_i(t)$ and velocities $v_i(t)$ and compute the corresponding fitness values $f(x_i(t))$. Each initial particle is denoted as the local best particle p_{best}, and the particle that has the best fitness value is denoted as the global best particle g_{best}.

Step 2. Update the position of a particle $x_i(t+1)$ according to its updated velocity $v_i(t+1)$, which are determined by using Equations (1) and (2), respectively.

Step 3. Compute the fitness value of the updated particle $f(x_i(t+1))$.

Step 4. Compare this new fitness value $f(x_i(t+1))$ with $f(p_{best})$. If $f(x_i(t+1)) < f(p_{best})$, p_{best} is replaced by $x_i(t+1)$. Otherwise, p_{best} remains unchanged.

Step 5. Compare the current fitness value $f(x_i(t+1))$ with $f(g_{best})$. If $f(x_i(t+1)) < f(g_{best})$, g_{best} is replaced by $x_i(t+1)$. Otherwise, g_{best} remains unchanged.

Step 6. If the terminal condition is met, then the best particle g_{best} and its fitness value are output. Otherwise, go back to Step 2.

3 Proposed Scheme

In the turtle-shell matrix, there are eight integers from 0 to 7 in a turtle shell or a 3×3 sub-block due to the architectural property of the matrix. In order to minimize the distortion of the cover image, it is important to determine the best table to replace the original eight integers. An example of substitution tables is shown in Figure 3. There is a total of 40,320 ways to sort these eight integers in total. So, it would take a lot of time to search all of the tables to find the best one. Thus, in this paper, we used PSO to search for the best table of the matrix.

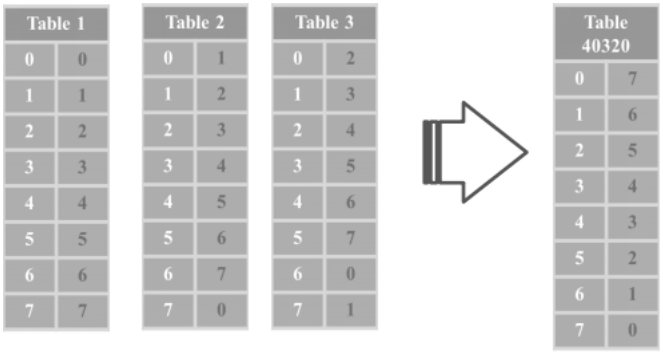

Figure 3: Substitution tables

3.1 Searching for The Best Replaceable Table Based on PSO

Input: A matrix M based on turtle shells, a cover image CI, secret messages S;

Output: A near optimal replaceable table T;

Step 1. The initial phase.

First, a population of particles is initialized randomly. Each particle represents a table that has 8 different values that range from 0 to 7. Table 1 shows an example of an initial particle, $P_i(t)$=($x_{i1}(t)$,$x_{i2}(t)$,...,$x_{i8}(t)$), where $x_{ij}(t)$ represents the initial position of this particle and each position has a different value. The initial particle denotes its local best particle as p_{best}, and the particle that has the best fitness value is denoted as the global best particle g_{best}. The velocity of each position of a particle, $v_{ij}(t)$, is chosen randomly from 0 through 7.

Step 2. Update one position of a particle.

In order to keep a particle that has 8 different values, each position should be saved before this position is updated. The parameter $temp_{ij}$ is used to save the original values of these particles and it is shown in Equation (3).

$$temp_{ij} = temp_{ij}(t), i = 1, 2, \cdots, n; \quad j = 1, 2, \cdots, 8. \tag{3}$$

The velocity of a particle is updated mainly according to three values, i.e., the local best particle, the global best particle and its particle value, which is shown in Equation (4).

$$v_{ij}(t+1) = wv_{ij}(t) + c_1 r_1 (p_{best(ij)} - x_{ij}(t)) + c_2 r_2 (g_{best(j)} - x_i(t)), \tag{4}$$

where i is the i^th particle, j is the j^th position of this particle, c_1 and c_2 are two acceleration coefficients, both of which are set to 1, and w is an inertia factor that is set to 0.4.

Then the particle's position is updated as $x_{ij}(t+1)$ according to its corresponding velocity. When the position of this particle is updated, the corresponding value should be limited in the range of 0 through 7. A modular function is used to achieve this purpose, as shown in Equation (5).

$$v_{ij}(t+1) = (x_{ij}(t)) + v_{ij}(t+1)) \bmod 8. \tag{5}$$

If the new value has already existed in this particle, the old value will be replaced by the saved value $temp_{ij}$ that is shown in Equation (6). In other words, the two values of this particle in different positions are exchanged. In each circulation, only one position

Table 1: An example of an initial particle $P_i(t)$

position	$x_{i1}(t)$	$x_{i2}(t)$	$x_{i3}(t)$	$x_{i4}(t)$	$x_{i5}(t)$	$x_{i6}(t)$	$x_{i7}(t)$	$x_{i8}(t)$
value	2	5	7	1	6	3	4	0

of the particle is updated. In this way, this particle is updated as $P_{ij}(t+1)$=($x_{i1}(t)$,$x_{i2}(t)$,...,$x_{ij}(t+1)$,...,,$x_{i8}(t)$).

$$x_{ik}(t+1) = \begin{cases} temp_{ij} & \text{if } x_{ij}(t+1) = x_{ik}(t) \\ & k = 1, 2, \ldots, 8, k \neq j, \\ x_{ik}(t) & \text{otherwise} \end{cases} \quad (6)$$

Step 3. Compute the corresponding fitness value.

After getting the new updated particle, $P_i(t+1)$, it is used to replace the eight integers in the matrix that is from 0 through 7. Then, a new matrix is generated, and it is used to embed secret data into cover images. In order to judge whether the new matrix is better than the original matrix, the fitness function is used to measure the distortion between the cover image and the stego-image as shown in Equation (7).

$$f(P_i(t+1)) = \sum_{i=1}^{H} \sum_{j=1}^{W} (CI_{hw} - SI_{hw})^2, \quad (7)$$

where CI_{hw} and SI_{hw} are the pixel values of the original image and the cover image, respectively, and $H \times W$ is the size of the image.

For example, if we use the particle, $P_i(t)$ = (2, 5, 7, 1, 6, 3, 4, 0), in Table 1 to replace the original eight integers, $P_0(t) = (0, 1, 2, 3, 4, 5, 6, 7)$, one by one at corresponding positions, a new matrix will be obtained, as shown in Figure 4. The main architectural property of the matrix is not changed, which means that any 3×3 sub-block or a turtle shell still contains 8 different values from 0 through 7. Suppose that two secret digits, e.g., 0 and 7, must be embedded into two pixel pairs, i.e., (4, 4) and (0, 1), respectively. According to the theory mentioned in Subsection 2.1, the new pixel pairs will be (5, 4) and (0, 1) after embedding the secret digit based on the new matrix. The fitness value between the original two pixel pairs and the new pixel pairs is computed as $f(P_i(t))=(4-4)^2+(5-4)^2+(4-4)^2+(0-0)^2+(1-1)^2$ based on the new matrix. If the original matrix is used to hide the two secret digits by using the same pixel pairs, as shown in Figure 1, the new pixel pairs will be (4, 5) and (2, 2). The corresponding fitness value is computed as $f(P_0(t))=(4-4)^2+(5-4)^2+(4-4)^2+(2-0)^2+(2-1)^2$. Since $f(P_i(t))¡f(P_0(t))$, the distortion associated with embedding the secret digits using the new matrix is less than that by using the original matrix.

Step 4. Update p_{best} and g_{best}.

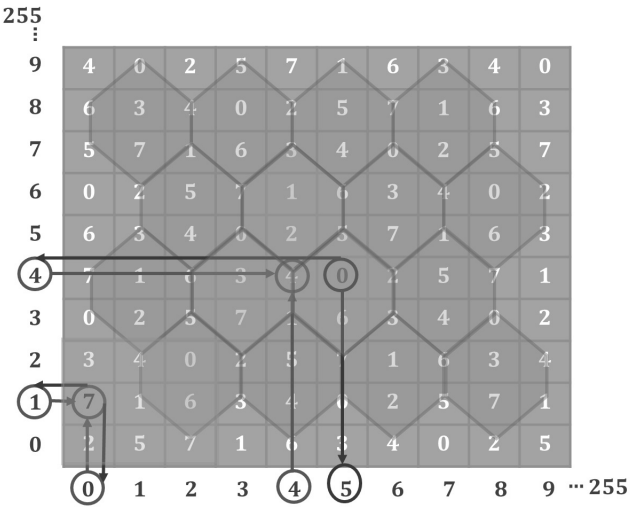

Figure 4: An example of the new matrix based on turtle shells

If $f(P_i(t+1))¡f(p_{best})$, p_{best} will be updated by this new particle. Otherwise, p_{best} will remain unchanged. If $f(P_i(t+1))¡f(g_{best})$, g_{best} will be updated by this new particle. Otherwise, g_{best} will remain unchanged.

Step 5. Termination.

If the terminal condition is encountered, then the best particle, g_{best}, is output. Otherwise, go back to Step 2. Here, the terminal condition is the maximum iteration time. After the near optimal table T is obtained, it is saved as a secret key for hiding and extracting the secret data.

3.2 Data Hiding Procedure

In this procedure, first, a new matrix is generated by replacing the original eight integers ranging from 0 through 7 in ascending order with the integers in the near optimal table. And then, each three bits of the secret messages are converted into a decimal digit ranging from 0 to 7. Selecting a pair of pixels from a cover image CI as a pair of coordinates of the new matrix, the secret digit is embedded in the same way that is used in the turtle shell scheme. After all of the pixel pairs are used for embedding secret digits, a stego-image SI is generated, and it is delivered, along with the near optimal table T, to a participant.

3.3 Data Extracting Procedure

In this procedure, first, the original matrix is generated using the original rule described in related work. By using the saved near optimal table T, a new matrix can be generated by replacing the original eight integers ranging from 0 to 7 in an ascending order with the integers in the best table. Then select a pair of pixels from the stego-image SI as a coordinates of the new matrix for extracting the secret digits. After all of the stego pixel pairs have been mapped into the new matrix, the original binary secret messages S can be restored from the extracted digits.

4 Experiments and Discussion

This section describes the experiments that were conducted on a group of 512×512 gray-level images that are shown in Figure 5. The simulation environment of our experiments was a PC with a 2.1 GHz CPU and 4 GB RAM. All schemes were implemented by using MATLAB 2013a. The population of particles was set to 8, and the maximum iteration times were set to 20 in our algorithm.

Figure 5: Six 515×512 gray images

The $PSNR$ was used to evaluate the visual quality of the image after embedding the secret data, which is shown in Equation (8). O_{hw} and C_{hw} denote the pixel values of the original image and the cover image, respectively, and $H \times W$ denotes the size of the image.

$$PSNR = 10log_{10}(\frac{255^2}{MSE}), \tag{8}$$

where

$$MSE = \frac{1}{H \times W}\sum_{h=1}^{H}\sum_{w=1}^{W}(O_{hw} - C_{hw})^2. \tag{9}$$

In order to prove the efficiency of our algorithm, we used the exhaustive search and a genetic algorithm (GA) to search the best table. The exhaustive search was intended to identify the best table among the 40,320 tables. For the GA, a chromosome is presented as a table that contains 8 integers ranging from 0 to 7 and each integer represents a gene of this chromosome. A chromosome was selected according to its probability, which depended on the corresponding $PSNR$ value. The population size was set to 8. The crossover operator was implemented by exchanging four genes of two chromosomes. The probability of crossover was controlled by a parameter that was set to 0.8. The mutation operation was to exchange two genes of one chromosome and the probability of mutation was set to 0.001.

We used the scheme based on the turtle-shell matrix to hide the secret digits. There are 393,216 bits embedded into the cover image and the embedding capacity (EC) is 18.75% of the size of the cover image. Table 2 provides a comparison of various schemes. Three different methods were used to search the best table to obtain the minimum distortion of the turtle-shell matrix, i.e., the exhaustive search, GA, and PSO. From Table 2, it is apparent that exhaustive search had the best $PSNR$ values among the three schemes. Both GA and our scheme achieved near optimal results, and the average improvements of $PSNR$ values for the two schemes were 0.02 and 0.03dB, respectively, compared with the original scheme based on the turtle-shell matrix.

The main purpose of our scheme is to find a best table to obtain the minimum distortion of the turtle-shell matrix. Recently, Liu et al. [18] enhanced the embedding capacity by using the location table to develop the turtle shell-based scheme. However, the turtle-shell matrix used in Liu et al.'s scheme is also not optimal, meaning that our scheme can be used in [18] to obtain the minimum distortion of the turtle-shell matrix. The experimental results are shown in Table 3. There are 524,288 bits embedded into the cover image and the embedding capacity (EC) is up to 25% of the size of the cover image. From Table 3, it is obvious that our scheme achieves higher $PSNR$ values.

Table 4 shows the encoding time requited of different schemes. Although [7] and [18] cost less time than other schemes, the turtle-shell matrix in their schemes is not optimal. The exhaustive search took the most time to find the best table among these methods, because all of the tables were tested. The GA took less time than the exhaustive search, but the $PSNR$ values were lower than those for the exhaustive search and PSO, as shown in

Table 2: Comparison of various schemes

Images	Ref.[18]		Exhaustive search		GA		PSO	
	EC	PSNR	EC	PSNR	EC	PSNR	EC	PSNR
Baboon	18.75%	49.45	18.75%	49.49	18.75%	49.48	18.75%	49.49
Boat	18.75%	49.46	18.75%	49.48	18.75%	49.48	18.75%	49.48
Peppers	18.75%	49.44	18.75%	49.49	18.75%	49.48	18.75%	49.49
Barb	18.75%	49.45	18.75%	49.49	18.75%	49.48	18.75%	49.49
Lena	18.75%	49.48	18.75%	49.51	18.75%	49.49	18.75%	49.50
Goldhill	18.75%	49.46	18.75%	49.49	18.75%	49.47	18.75%	49.48
Average	18.75%	49.46	18.75%	49.49	18.75%	49.48	18.75%	49.49

Table 3: Comparisons of the proposed scheme and [18]

Images	Ref.[19]		PSO	
	EC	PSNR	EC	PSNR
Baboon	25%	45.55	25%	45.57
Boat	25%	45.55	25%	45.58
Peppers	25%	45.54	25%	45.56
Barb	25%	45.56	25%	45.58
Lena	25%	45.55	25%	45.57
Goldhill	25%	45.49	25%	45.52
Average	25%	45.54	25%	45.56

Table 2. Compared with other methods, our PSO algorithm achieved near optimal results and more efficiently identified the near optimal table.

In addition, the near optimal table can be used as a secret key for the extraction of secret data. There were 40,320 tables to generate a matrix for the extraction of data extraction. If a receiver were to use a fake table to generate the matrix of turtle shells, the false secret data would be extracted by using the pixel pairs of the stego-image. Therefore, the security of the scheme based on the turtle-shell matrix was enhanced by using our method.

5 Conclusions

In this paper, we proposed a novel scheme to optimize data hiding based on turtle shells. First, we generated an original matrix based on the turtle shell scheme. Then PSO was used to search the near optimal table of the matrix. The integers in the matrix of the turtle shells are replaced by the near optimal table in order to minimize the distortion of the image. Two pixels of the cover image were used to embed secret digits according to this near optimal matrix. Our scheme has the same embedding capacity as the original data hiding scheme based on turtle shells, but it provides higher PSNR values. Also, the near optimal table can be used as a secret key for extracting data, which enhances the security of the scheme based on the turtle-shell matrix. The experimental results showed that the proposed scheme had better visual quality than the original scheme based on the turtle-shell matrix and was efficient in finding the near optimal table.

Acknowledgments

This work has been supported in part by National Natural Science Foundation of China (No. 61272262 and No. 61210006), The Program of "One hundred Talented People" of Shanxi Province, Research Project Supported by Shanxi Scholarship Council of China (2014-056), Program for New Century Excellent Talent in Universities (NCET-12-1037), International Cooperative Program of Shanxi Province (No. 2015081015), and Scientific and Technological project of Shanxi Province (20150310032). Innovative Projects for Graduate Students of TYUST (No.20145018).

References

[1] A. A. Abdelwahab and L. A. Hassan, "A discrete wavelet transform based technique for image data hiding," in *Proceedings of 25th National Radio Science Conference*, pp. 1–9, Egypt, 2008.

[2] W. Bender, D. Gruhl, N. Morimoto, and A. Lu, "Techniques for data hiding," *IBM Systems Journal*, vol. 35, no. 3&4, pp. 313–336, 1996.

[3] N. Bourbakis and C. Alexopoulos, "Picture data encryption using scan patterns," *Pattern Recognition*, vol. 25, no. 6, pp. 567–581, 1992.

[4] C. C. Chang, Y. C. Chou, and T. D. Kieu, "An Information Hiding Scheme Using Sudoku," in *Proceedings of the Third International Conference on Innovative Computing, Information and Control*, pp. 17–22, 2008.

[5] C. C Chang, T. D. Kieu, and W. C. Wu, "A lossless data embedding technique by joint neighboring coding," *Patter Recognition*, vol. 42, no. 7, pp. 1597–1603, 2009.

[6] C. C. Chang, C. C. Lin, C. S. Tseng, and W. L. Tai, "Reversible hiding in DCT-based compressed images," *Information Sciences*, vol. 177, no. 13, pp. 2768–2786, 2007.

Table 4: Encoding time requited of different schemes (s)

Images	Ref.[18]	Ref.[19]	Exhaustive search	GA	PSO
Baboon	12.35	25.47	54149.21	1284.19	175.21
Boat	13.14	24.52	59712.17	1230.43	171.74
Peppers	11.86	23.76	54038.28	1238.56	173.36
Barb	11.57	24.15	52824.63	1142.42	173.15
Lena	12.63	25.24	54856.31	1190.71	172.97
Goldhill	13.72	24.68	59942.15	1302.48	173.63
Average	12.55	24.64	55920.46	1231.47	173.34

[7] C. C. Chang, Y. Liu and T. S. Nguyen, "A novel turtle shell based scheme for data hiding," in *Proceedings of 2014 Tenth International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP)*, pp. 89–93, 2014

[8] C. C. Chang and W. C. Wu, "Hiding secret data adaptively in vector quantisation index tables," *IEE Proceedings-Vision, Image and Signal Processing*, vol. 153, no. 5, pp. 589–597, 2006.

[9] W. C. Du, and W. J. Hsu, "Adaptive data hiding based on VQ compressed images," *IEE Proceedings-Vision, Image and Signal Processing*, vol. 150, no. 4, pp. 233–238, 2003.

[10] W. Hong, T. S. Chen, and C. W. Shiu, "A Minimal Euclidean Distance Searching Technique for Sudoku Steganography," in *Proceedings of International Symposium on Information Science and Engineering*, vol. 1, pp. 515–518, 2008.

[11] J. Horn, N. Nafpliotis, and D. E. Goldberg, "A niched Pareto genetic algorithm for multiobjective optimization," in *Proceedings of the First IEEE Conference on Evolutionary Computation, IEEE World Congress on Computational Intelligence*, vol. 1, pp. 82–87, 1994.

[12] Y. C. Hu, "High capacity image hiding scheme based on vector quantization," *Pattern Recognition*, vol. 39, no. 9, pp. 1715–1724, 2006.

[13] J. K. Jan, and Y. M. Tseng, "On the security of image encryption method," *Information Processing Letters*, vol. 60, no. 5, pp. 261–265, 1996.

[14] J. Kennedy, and R. C. Eberhart, "Particle swarm optimization," in *Proceedings of ?the 1995 IEEE international conference on neural network*, pp. 1942–1948, 1995.

[15] H. J. Kim, C. Kim, Y. Choi, S. Wang, and X. Zhang, "Improved Modification Direction Schemes," *Computers & Mathematics with Applications*, vol. 60, pp. 319–325, 2010.

[16] C. C Lin, and P. F. Shiu, "High capacity data hiding scheme for DCT-based images," *Journal of Information Hiding and Multimedia Signal Processing*, vol. 1, no. 3, pp. 220–240, 2010.

[17] H. Liu, J. Liu, J. Huang, D. Huang, and Y. Q. Shi, "A robust DWT-based blind data hiding algorithm," in *Proceedings of IEEE International Symposium on Circuits and Systems*, vol. 2, pp. 672–675, Arizona, USA, 2002.

[18] Y. Liu, C. C. Chang, and T. S. Nguyen, "High capacity turtle shell-based data hiding," *IET Image Processing*, vol. 10, no. 2, pp. 130–137, 2016.

[19] B. Liu, L. Wang, and Y. H. Jin, "An effective PSO-Based Memetic Algorithm for Flow Shop Scheduling," *IEEE Transactions on Systems, Man and Cybernetics, Part B: Cybernetics*, vol. 37, no. 1, pp. 18–27, 2007.

[20] J. Mielikainen, "LSB matching revisited," *IEEE Signal Processing Letters*, vol. 13, no. 5, pp. 285–287, 2006.

[21] S. Naka, T. Genji, and T. Yura, "A hybrid particle swarm optimization for distribution state estimation," *IEEE Transactions on Power Systems*, vol. 18, no 1, pp. 60–68, 2003.

[22] R. Z. Wang, C. F. Lin, and J. C. Lin, "Image hiding by optimal LSB substitution and genetic algorithm," *Pattern Recognition*, vol. 34, no. 3, pp. 671–683, 2001.

[23] X. Zhang, and S. Wang, "Efficient Steganographic Embedding by Exploiting Modification Direction," *IEEE Communications Letters*, vol. 10, no. 11, pp. 781–783, 2006.

[24] B. Zhao, C. X. Guo, and Y. J. Cao, "A multiagent-based particle swarm optimization approach for optimal reactive power dispatch," *IEEE Transactions on Power Systems*, vol. 20, no. 2, pp. 1070–1078, 2005.

Qiang Jin was born in Shanxi Province, China, in 1989. He is currently pursuing the M.E. degree in Taiyuan University of Science and Technology. His current research interests include data hiding, and image compression.

Zhihong Li is currently an associate professor in Taiyuan University of Science and Technology, China. He received the B.Eng. in electronic information engineering from Taiyuan University of Science and Technology in 1994. His research interests include compressed sensing, and secret image sharing. He was participated in the projects on distributed video coding and now is leading one research project on image secret from Shanxi Natural Science Foundation.

Chin-Chen Chang received the B.S. degree in applied

mathematics and the M.S. degree in computer and decision sciences from National Tsing Hua University, Hsinchu, Taiwan, R.O.C., in 1977 and 1979, respectively. He received his Ph. D in computer engineering in 1982 from the National Chiao Tung University, Hsinchu, Taiwan. Since February 2005, he has been a Chair Professor of Feng Chia University. In addition, he has served as a consultant to several research institutes and government departments. His current research interests include database design, computer cryptography, image compression and data structures.

Anhong Wang received B.E and M.E. degrees from Taiyuan University of Science and Technology (TYUST) respectively in 1994 and 2002, and Ph. D degree in Institute of Information Science, Beijing Jiaotong University in 2009. She became an associate professor with TYUST in 2005 and became a professor in 2009. She is now the director of Institute of Digital Media and Communication, Taiyuan University of Science and Technology. Her research interest includes image/video coding, compressed sensing, and secret image sharing. Now she is leading two national research projects from National Science Foundation of China.

Li Liu received her B.E. degree in communication engineering in 2002, from Lanzhou Railway University and M.E. degree in communication and information system in 2006, from Lanzhou Jiaotong University. Her current research interests include image compression and secret sharing.