Journal of Ambient Intelligence and Humanized Computing (2020) 11:3771–3786
https://doi.org/10.1007/s12652-019-01580-z

ORIGINAL RESEARCH

Differential evolution optimization of intertwining logistic map-DNA based image encryption technique

Mohit Dua[1] · Aishwarya Wesanekar[1] · Vishwas Gupta[1] · Mayank Bhola[1] · Shelza Dua[2]

Received: 14 January 2019 / Accepted: 12 November 2019 / Published online: 19 November 2019

Abstract

Differential evolution (DE) is a powerful evolutionary algorithms, widely applied in different fields of science and engineering for solving the problem of optimization. Since image encryption has been viewed as an interesting research topic by many experts and innumerable methods to encrypt images have emerged, currently, the focus is on obtaining optimized images. The paper presents a novel image encryption scheme that uses intertwining logistic map (ILM), DNA encoding and DE optimization. The proposed approach is based on three phases: permutation involving ILM, diffusion engaging DNA and optimization using DE. Parameters like entropy, key sensitivity, secret key space, unified average change in intensity (UACI), correlation coefficient —vertical, horizontal and diagonal, and number of pixel change rate have been evaluated to test the efficiency of the proposed method. The paper also compares this performance with that of the genetic algorithms (GA), used previously for optimization. The significance of this approach is enhancing entropy, the essential characteristic of randomness, resisting against numerous statistical and differential attacks and generating good experimental results. The main contribution of this paper is to present the efficiency of DE in image optimization and exhibit how DE is better than GA.

Keywords ILM · DE · DNA · Image encryption

1 Introduction

Due to the fast development of network-related technologies in the last few decades, there has been a heavy usage of digital data for the purpose of information exchange (Chen et al. 2018; Suneja et al. 2019). Digital image, being an important integral component of this data, has different security concerns associated with it. For this, a number of image encryption algorithms have been proposed, the conventional being—Rivest–Shamir–Adleman (RSA), international data encryption algorithm (IDEA) and data encryption standard (DES). However, the intrinsic properties of images such as strong redundancy, high correlations among pixels, and bulky data capacity have led to the outmoding of the aforementioned standard encryption schemes (Enayatifar et al. 2017; Li et al. 2007; Solak et al. 2010; Solak and Çokal 2011; Wang et al. 2015a, b, c).

A colossal interest has emerged in the study of chaotic systems for image encryption as these types of systems work on a phase space of real numbers, are sensitive to initial conditions, and are also, stochastic or random in nature (Bisht et al. 2019a, b; Sneha et al. 2019). The similarity of chaos and cryptography has further catalysed the engagement of chaos theory in encryption. While cryptography involves rounds and secret key, chaos makes use of iterations and control parameters. Image encryption, generally, can be divided into two sections—diffusion and permutation. The need of diffusion arises in order to render the statistics of encrypted data independent of the original data. Permutation is also a prerequisite for augmenting the complexity between the key and image

✉ Mohit Dua
er.mohitdua@nitkkr.ac.in

Aishwarya Wesanekar
aishwes.1996@gmail.com

Vishwas Gupta
vishwas294@gmail.com

Mayank Bhola
mayankbhola71@gmail.com

Shelza Dua
shelza_ecn@yahoo.com

[1] Department of Computer Engineering, National Institute of Technology, Kurukshetra, India

[2] Department of Electronics and Communication Engineering, National Institute of Technology, Kurukshetra, India

pixels, and this randomized complexity can be obtained using chaotic systems. In 1990, an approach for controlling chaotic system was proposed by Ott et al.(1990). Later, Fridrich (1998) also devised chaos based encryption and since then, various researches have been carried out to use chaos theory for reducing the redundancies of the encrypted image (Chen et al. 2004; Masuda et al. 2006). Yet, some common-attacks are not resisted by chaos-based algorithms (Zhang et al. 2012; Rhouma and Safya 2008). Spatial bit-level permutation explored in (Liu and Wang 2011) by dividing the color images into grayscale matrices has improved encryption algorithm and a more recent algorithm (Zhang and Wang 2015) exhibits spatiotemporal chaos bringing about higher efficiency. The proposed algorithm engages ILM, a high dimension chaotic system to obtain bit-level permutations of the individual R, G, B matrices, leading to a higher security and giving good encryption results.

Adleman (1994) proposed a new DNA-based method for image encryption, which consistently improved the security as an effective biological tool (Liu et al. 2012a, b; Zhang et al. 2010a, b, 2013). The fundamental nature of DNA to act as a carrier of information has led to researchers proposing varied encryption algorithms using DNA over chaos (Xiao et al. 2006; Zhang and Fu 2012). Moreover, because of huge storage, massive parallelism, and super-low power consumption, DNA encoding proves to be secure and efficacious (Head et al. 2000; Zheng et al. 2009). Combining DNA with high dimension chaotic system such as ILM results in a robust encryption system. DNA XOR, DNA XNOR and DNA addition are few of the operations that can be applied alongside chaos (Liu et al. 2012a, b). The only limitation of DNA encoding is the independency of the key-stream creation from the cipher text (Zhang 2015). This inadequacy of DNA paves the way for optimization of images.

Inspired by the evolution of natural species, EA has been variedly applied in order to solve the optimization problems in diverse areas of engineering. Storn (1995) introduced a stochastic population-based search technique, DE. Today, DE is being fruitfully applied in numerous fields such as communication (Storn 1996), pattern recognition (Ilonen et al. 2003), and mechanical engineering (Joshi and Sanderson 1999). To ensure operative functioning of the algorithm, appropriate evolutionary operators and effective encoding schemes need to be determined (Qin et al. 2009; Tuson and Ross 1998; Gómez et al. 2003; Julstrom 1995). In the proposed algorithm, the candidates can replace the parents, or the initial population, depending upon their fitness values. An optimized sequence is successfully produced and it can be further utilized to achieve an efficacious encryption scheme. In this paper, entropy has been employed for the fitness function. Generally, images which show high entropy are considered to be efficiently encrypted and DE operates on this basis.

In 2010, a novel technique for secret key generation through 128-bit hash function using MD5 of mouse-positions is introduced (Liu and Wang 2010). The paper exhibits greater security through larger key space produced by the one-time keys. In the proposed algorithm, to improve upon the security of encryption, SHA-256 is applied (Guesmi et al. 2016). The SHA-256 function produces a 256-bit hash from a 120-bit input, which expands the key-space to 2^{256}. A small variation in the input bit can result to very large variation in the output, leading to Avalanche effect, which reduces the probability of the brute-force attack. There have been several new advances in the field of chaos based encryption system. Chaos systems based on mathematical models such as perceptron exhibit input parameters or weights being altered dynamically (Wang et al. 2010). A more recent algorithm (Wang et al. 2019) shows the power of fast encryption in real-time system. Incorporating parallel computation in this algorithm will improve the execution speed of the algorithm.

The main background and motivation behind the proposed work is the techniques proposed in (Abdullah et al. 2012; Suri and Vijay 2017; Enayatifar et al. 2014). In the year 2012, Abdullah et al. first time combined logistic map (LM) with genetic algorithm (GA) to introduce an optimized and more secure image encryption approach. Later in the year 2014, Enayatifar et al. (2014) extended the work by combining DNA with LM and GA to make the algorithm more secure and optimized. Recently, Suri et al. extended these two works by using weighted GA (a bi-objective approach of GA) with LM and GA to address the issue of objective selection while doing optimization using GA. The implemented approach in this paper, targets weakness of LM by using a better and efficient chaos map ILM, combines it with DNA and uses Differential optimization to get faster and optimized results. The evaluated parameters show that the conflation of ILM-based permutation, DNA diffusion and DE based optimization produces an optimized encrypted image, secure for transmission with high dynamicity due to the utilization of evolutionary algorithms.

The remaining paper is divided in the following manner. Section 2 enumerates the fundamentals that form the crucial elements of the algorithm. Section 3 discusses the proposed algorithm. Analysis is included in Sect. 4, and the final conclusion is presented in Sect. 5.

2 Preliminaries

This section elucidates the fundamental techniques used in our proposed algorithm.

2.1 Intertwining logistic map (ILM)

The classic chaotic system, LM is the simplest of all and is popular to its dynamicity (Fridrich 1998; Zhang et al. 2010a, b). The mathematical expression for LM function is defined as:

$$p_k = \mu \times p_k(1 - p_{k+1}) \tag{1}$$

where p_k denotes the kth position value of the sequence and lies in the range (0,1]. The control parameter μ is kept in the range of 3.57 and 4 to have a completely chaotic sequence.

Despite exhibiting randomness, the one-dimensional LM is sensitive to only one control parameter and has a smaller key-space and. The authors of (Alvarez and Li 2006), extended this one-dimensional LM function to a two-dimensional chaotic function that is mathematically expressed as:

$$p_{k+1} = \lambda_1 \times p_k\left(1 - p_k\right) + \delta_1 \times p_k^2 \tag{2}$$

$$q_{k+1} = \lambda_2 \times p_k\left(1 - q_k\right) + \delta_2\left(p_k^2 + p_k q_k\right) \tag{3}$$

where p and q are the two chaotic sequences, lying in the range of (0,1], are generated using above two-dimensional chaotic function. The variables and δ are taken as $2.75 < \lambda_1 \leq 3.4$, $2.75 < \lambda_2 \leq 3.45$, $0.15 < \delta_1 \leq 0.21$, $0.13 < \delta_2 \leq 0.15$ to have a complete chaotic sequence. In Khade and Narnaware (2012), a three-dimensional LM has been proposed by extending the function of the two-dimensional LM, which is mathematically expressed as:

$$p_{k+1} = \lambda \times p_k\left(1 - p_k\right) + \delta q_k^2 p_k + \mu r_k^3 \tag{4}$$

$$q_{k+1} = \lambda \times q_k\left(1 - q_k\right) + \delta r_k^2 q_k + \mu p_k^3 \tag{5}$$

$$r_{k+1} = \lambda \times r_k\left(1 - r_k\right) + \delta p_k^2 r_k + \mu q_k^3 \tag{6}$$

The variables , δ and μ are taken as 0.53 < < 3.81, 0< < 0.022, and 0< < 0.015, respectively. The p_0, q_0 and r_0 are kept in range the (0,1] to represent a non-linear chaotic system.

Wang and Xu (2014) designed an intertwining relation, where the value of one sequence depends upon the other two sequences, between three different LM sequences (Khade and Narnaware 2012; Kumar et al. 2016):

$$p_{k+1} = \left[\lambda \times \alpha \times q_k \times \left(1 - p_k\right) + r_k\right] Mod1 \tag{7}$$

$$q_{k+1} = \left[\lambda \times \beta v q_k + r_k \times \left(1 + p_{k+1}^2\right)\right] Mod1 \tag{8}$$

$$r_{k+1} = \left[\lambda \times \left(q_{k+1} + p_{k+1} + \gamma\right) \times \sin\left(r_k\right)\right] Mod1. \tag{9}$$

where λ have values between 0 and 3.9999, $\alpha > 33.5$, $\beta > 37.9$, $\gamma > 35.7$.

ILM chaotic sequences showcase uniform distribution as compared to LM sequence (Suri and Vijay 2019). Hence, the disadvantages of one-dimensional LM like blank windows, stable windows, and irregular distributions of iterated sequences are overcome by ILM (Chen et al. 2011). Figure from Wang and Xu (2014), compares the Lyapunov exponents of LM with that of ILM. It can clearly be seen that the Lyapunov exponents of ILM are all above zero, reinforcing the dynamical nature of ILM. Consequently, ILM is used in the approach for scrambling of image pixels (Fig. 1).

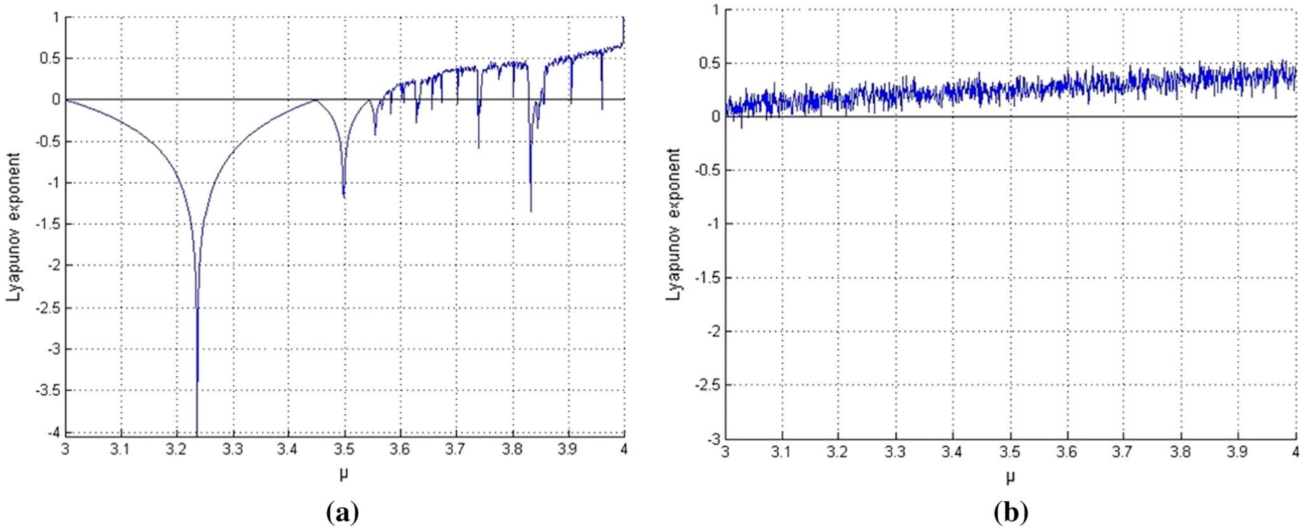


Fig. 1 Lyapunov exponents of LM and ILM (Wang and Xu 2014)

Table 1 DNA encoding–decoding rules

Rules	G (guanine)	A (adenine)	C (cytosine)	T (thymine)
R_1	01	00	10	11
R_2	10	00	01	11
R_3	01	11	10	00
R_4	10	11	01	00
R_5	11	10	00	01
R_6	11	01	00	10
R_7	00	10	11	01
R_8	00	01	11	10

Table 2 DNA XOR operation

XOR	G (guanine)	A (adenine)	C (cytosine)	T (thymine)
A	G	A	C	T
C	T	C	A	G
G	A	G	T	C
T	C	T	G	A

2.2 Deoxyribonucleic acid (DNA)

In 1994, the first analysis of DNA computing was performed by Adleman. A (adenine), C (cytosine), G (guanine) and T (thymine) are the four nucleic acids that comprise a DNA sequence. It can be inferred from the Watson–Crick relationship, that pairing of Adenine nucleic acid is always done with Thymine nucleic acid to represent as complement sand pairing of Guanine is always done with Cytosine to represent as complements. DNA can be applied in encryption using binary system (Wang et al. 2015a, b, c; Enayatifar et al. 2015; Zhang et al. 2016). Tables 1 and 2 show the DNA encoding–decoding rules and DNA XOR operation respectively.

2.2.1 DNA rules

There are eight different ways of assigning two-bit values to all the four nucleic acids. Table 1 defines the assignment based on the rule number.

2.2.2 DNA XOR operation

When two of the nucleic acids undergo XOR operation, it is termed as DNA XOR. Following all the properties of DNA XOR, Table 2 shows the result of performing the operation.

2.3 Differential evolution (DE)

DE is a predominantly used EA in a wide range of scientific applications. Its high speed and low-resource utilization makes it a potential optimization tool for cryptosystems. DE differs from the conventional EA in its greedy approach for the selection of candidate. It aims at transforming the initial population, P to evolve into an optimum solution. Each vector in the initial population is multi-dimensional. The number of dimensions chosen to obtain the optimal solution depends upon the application on which DE is applied. For image encryption, the number of dimensions is taken equivalent to the size of the image. The population size, NP, determines the number of vectors, and is a critical parameter for DE optimization. DE, like other EA, involves three operations—mutation, crossover and selection. Mutation generates the mutant (biologically referred to as offspring) by making some alterations to the parents. Crossover engages the offspring and the parent to undergo a recombination process to produce the candidate vector. The interpolation of the offspring and the parent is determined by the crossover rate, CR. The selection operation then chooses the vector from among the offspring and the parent that will sustain. All the three operations- mutation, crossover and selection, are reiterated over again for the evolution of the optimum solution. Figure 2 exhibits the flow of DE algorithm used in this paper.

2.3.1 Mutation

The genetic operator mutation, is used to produce the offspring O from the parent vector P in the population for each iteration, i and each dimension j. Given below are few strategies through which mutation process is carried out.

1. DE/best/1

$$O_{i,j} = P_{best,j} + F\left(P_{r1(i),j} - P_{r2(i),j}\right) \tag{10}$$

2. DE/rand/1

$$O_{i,j} = P_{r1(i),j} + F\left(P_{r2(i),j} - P_{r3(i),j}\right) \tag{11}$$

3. DE/rand-to-best/1

$$O_{i,j} = P_{i,j} + F\left(P_{best,j} - P_{i,j}\right) + F\left(P_{r1(i),j} - P_{r2(i),j}\right) \tag{12}$$

4. DE/best/2

$$O_{i,j} = P_{best,j} + F\left(P_{r1(i),j} - P_{r2(i),j}\right) + F\left(P_{r3(i),j} - P_{r4(i),j}\right) \tag{13}$$

5. DE/rand/2

$$O_{i,j} = P_{r1(i),j} + F\left(P_{r2(i),j} - P_{r3(i),j}\right) + F\left(P_{r4(i),j} - P_{r5(i),j}\right) \tag{14}$$

As seen above, a scaling factor, F, is required in the process of mutation. Generally, a range of [0.4, 1] is viewed effective for better mutant generation (Qin et al. 2009). The

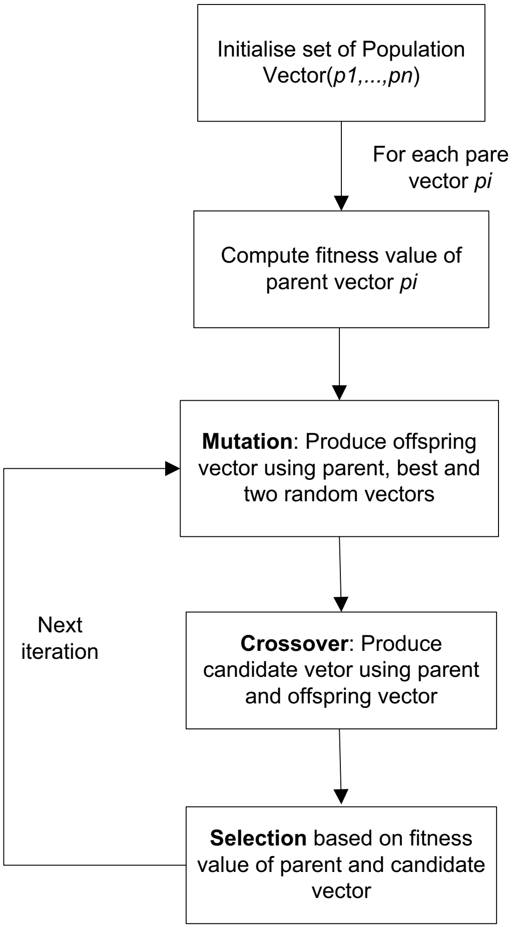

Initialise set of Population Vector(p1,...,pn)
For each parent vector pi
Compute fitness value of parent vector pi
Mutation: Produce offspring vector using parent, best and two random vectors
Next iteration
Crossover: Produce candidate vetor using parent and offspring vector
Selection based on fitness value of parent and candidate vector

Fig. 2 Differential evolution (DE) algorithm

random vector $P_{rx(i),j}$. is exclusive of both $P_{i,j}$ and $P_{best,j}$. The proposed approach uses the DE/rand-to-best/1 method for generating the corresponding mutant vectors.

2.3.2 Crossover

The mutation process produces the offspring vector as well as the parent vector. Interpolating both of them to generate a new candidate vector is done by the crossover process. For adequate crossover to take place, appropriate CR is required, which is taken as 0.8 in this case. Three specific types of crossover exist as mentioned below.

1. Single point crossover
   In this type of crossover, two vectors, i.e. the parent vector and the offspring vector are divided into two halves. The candidate vector is formed by fusing first half from one vector and the second half from another.
2. Two-point crossover
   Two-point crossover operation divides the parent vector and the offspring vector in three parts by earmarking two points for division. The new candidate vector is formed by taking each of the three parts from any of the two vectors.
3. Multi-point crossover
   Similarly, in this operation, the two vectors are divided into multiple parts by taking multiple points and then the candidate vector is generated by any of the corresponding parts from aforementioned vectors.

The multi-point approach to the crossover operation gives a better mix of the two vectors. Hence, the multi-point crossover operation is employed along with a CR value of 0.8 to produce an effective candidate.

$$C_{i,j} = \begin{cases} O_{i,j}, x_{i,j} \le CR \\ P_{i,j}, x_{i,j} > CR \end{cases}. \tag{15}$$

Here, x refers to a random value of the range [0,1) that determines the corresponding dimension of each candidate vector.

2.3.3 Selection

The final operation applied after mutation and crossover is selection. Like the name suggests, this operation simply selects the vector that will prevail for the future iterations. The fitness function is the basis on which the selection is made. The entropy $f_{x,}$ is used as the primary fitness function for this process. If the entropy of the parent vector is more, the parent vector is chosen for the next iteration. Else, the candidate vector is chosen as the next iteration parent vector.

$$P_{i,j+1} = \begin{cases} C_{i,j}, f_{Ci,j,} \ge f_{Pi,j,} \\ P_{i,j}, f_{Ci,j,} \le f_{Pi,j} \end{cases} \tag{16}$$

3 Proposed algorithm of image encryption

The algorithm of the encryption process is discussed from the very beginning. It describes the secret key generation through SHA-256 followed by DE optimization that involves chaos-based permutation and DNA diffusion. The entire flow of the image encryption algorithm is shown in Fig. 3.

3.1 Color image input

For simplifying the process of encryption, a plain color image is broken down into three two-dimensional pixel matrices—R (red), G (green) and B (blue) that are further converted to one-dimensional matrices. Table 3 gives the pseudo code for performing this input conversion.

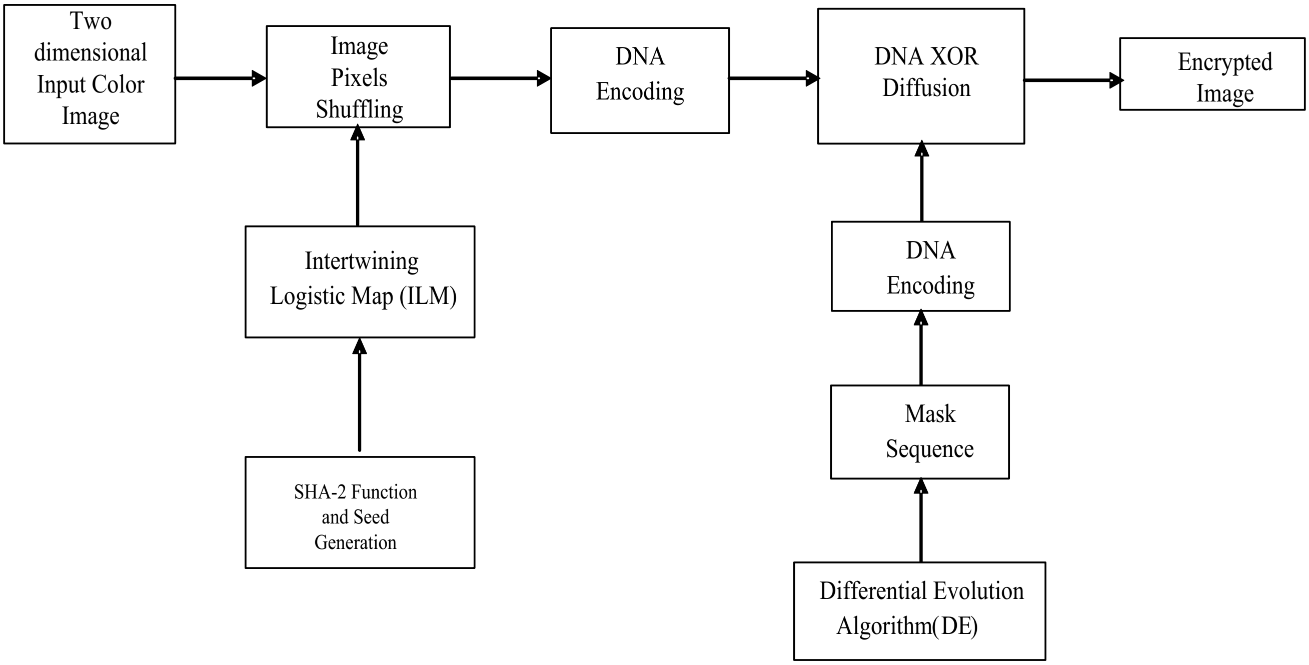

Two dimensional Input Color Image
Image Pixels Shuffling
DNA Encoding
DNA XOR Diffusion
Encrypted Image
Intertwining Logistic Map (ILM)
SHA-2 Function and Seed Generation
DNA Encoding
Mask Sequence
Differential Evolution Algorithm(DE)

Fig. 3 Block diagram of proposed approach

Table 3 Image input

```
INPUT: P_IMG(p, q)                          // Color Image (2D)
CODE:       WHILE i in range(0 : p) DO
                WHILE j in range(0 : q) DO
                    r, g, b←P_IMG(i, j)
                    IMG (i*q + j) ← [r, g, b]
                END WHILE
            END WHILE
OUTPUT:     IMG(p×q)                        //[r, g, b] 1D Matrices
```

3.2 Secret key generation using SHA-2

To have a larger key space and better key sensitivity, secure hash algorithm (SHA-2) has been used by the second step of the proposed approach to generate the seed value for the secret key. To generate this seed value, a 120-bit stochastically produced input initial secret key is used by the SHA-2 function. For three dimensions of an image, three chaotic sequences are generated by using 3 separate seeds. These seed values are generated using the pseudo code shown in Table 4.

3.3 First permutation

Using the three seed values generated in the second step, the third step of the proposed technique generates ILM function.

Table 4 SHA-2 function to generate seed values

```
INPUT:      KEY_LENGTH                                  // bits of Secret Key

CODE: //binary bit sequence
SECRET_KEY← generate_random_key(KEY_LENGTH)
            // generate hash
            KEY_ARRAY← data_hash(SECRET_KEY, [binaryMode, double, 'SHA-256'])

WHILE i in range(0 : KEY_LENGTH/2) DO
        CHECK← CHECK XOR KEY_ARRAY(i)
            END WHILE
            WHILE i in range(KEY_LENGTH/2 : KEY_LENGTH) DO
                SUM← SUM + KEY_ARRAY(i)
            END WHILE
            SEED← CHECK + SUM
            SEED← SEED/2^12
OUTPUT:     SEED
```

Table 5 First permutation

```
INPUT:      SEED                            // 3 seed values [x0,y0,z0]<-- [r,g,b]
CODE: ILM_SEQUENCE← generate_ILM(SEED) //3-LM sequences for [r, g, b]
            SHUFFLED_IMG← shuffle_img(IMG, ILM_SEQUENCE)
OUTPUT:     SHUFFLED_IMG
```

The one-dimensional R, G and B matrices are then shuffled using these three ILM generated sequences. Table 5 gives the pseudo code for this ILM based shuffling process.

3.4 Optimization through DE

In this step, the optimized mask sequence is obtained through DE. First, the population vector is randomly initialized and simultaneously, the fitness value for each is stored. For each iteration, each vector of the population undergoes the mutation, crossover and selection processes. Finally, the vector which has the best fitness value forms the optimized mask sequence. Table 6 shows the pseudo code for the same.

3.5 Final encryption

The final step includes steps 3.1–3.4 to generate the final cipher image. The seed obtained through SHA-256 is sent to generate the three ILM sequences. These sequences are used to shuffle the plain image. The optimized mask sequence is obtained through DE. This mask sequence is converted to the DNA format along with the shuffled image. The two then undergo diffusion by an operation of DNA XOR. The result is then DNA decoded to form the encrypted image. The pseudo code for the entire process is given in Table 7 and the entire flowchart is shown in Fig. 4.

Table 6 Optimized mask DNA through DE

```
INPUT:      Number of Pixels (N)//N <-- pXq
CODE: //Initialization
POP← generate random population vector of POPULATION_SIZE and N dimensions
ITERATIONS←assign number of iterations
BEST← 1
FOR iin range(POPULATION_SIZE) DO
FITNESS_VAL[i] ←fitness(POP[i])
IF FITNESS_VAL[i]>FITNESS_VAL[BEST] THEN
BEST ←i

FOR jin range(ITERATIONS) DO
FOR iin range(POPULATION_SIZE) DO
b,c← random indices other than i and BEST
//Mutation
t1← scale(POP[b], POP[c])
t2← scale(POP[BEST], POP[i])
OFFSPRING←mutation(POP[i], t1, t2)
//Crossover
CANDIDATE← crossover(OFFSPRING, POP[i])
//Selection
CAND_FITNESS← fitness(CANDIDATE)
IFCAND_FITNESS>FITNESS_VAL[i] THEN
POP[i]←CANDIDATE
FITNESS_VAL[i]←CAND_FITNESS
IFCAND_FITNESS>FITNESS_VAL[BEST] THEN
BEST←i

MASK_SEQ ← POP[BEST]
OUTPUT:     MASK_SEQ
```

Table 7 ILM-DE encryption

```
INPUT:      PLAIN_IMG, NUMBER_OF_PIXELS
CODE:       SHUFFLED_IMG ←initial_permutation(PLAIN_IMG, SEED)
            MASK_SEQUENCE ←differential_evolution(NUMBER_OF_PIXELS)
            Convert SHUFFLED_IMGandMASK_SEQUENCEto their respective DNA formats to obtain
DNA_SEQUENCE and MASK_DNA
ENCODED_IMG←dna_xor(DNA_SEQUENCE, MASK_DNA)
            Convert ENCODED_IMGto binary format from DNA to obtain ENCRYPTED_IMG
OUTPUT:     ENCRYPTED_IMG
```

4 Simulation results

This section gives the description of the experimental setup used to implement the proposed encryption technique. It also describes the different evaluation parameters that have been used to the test the encryption efficiency of the proposed method. An efficacious image encryption method should not only show resistance against differential and statistical attacks, but it should also be capable of handling brute force attacks viz. key sensitivity analysis and key space analysis. Hence, the proposed method has been evaluated against following various parameters such that the purpose of an efficient encryption technique can be achieved.

4.1 Experimental setup

For experimental setup, Python 2.7 has been used on the platform PyCharm 2.3 on a Windows 10 PC using an Intel Core i3 as the processor clocked at 1.7 GHz CPU with 4 GB RAM and 500 GB hard disk memory. Sample standard color images such as Lena, Bungee and Baboon of sizes 64×64, 128×128, 256×256 and 512×512 are tan as the input image data-set for conducting the experiments.

4.2 Key space analysis

This parameter determines the resisting ability of an encryption algorithm towards brute force attacks. It gives a measure of the key sample space from which encryption key selection is made. Hence, to reduce the feasibility of brute force the key sample space should be made very large. The proposed work uses SHA-2 function that generates a key-space of size 2^{256} that is considered to beis large enough to show resistance against brute-force attack (Alvarez and Li 2006).

4.3 Key sensitivity analysis

An efficient encryption technique is sensitive to minute changes in secret key used. This Avalanche effect is necessary for key sensitivity and it produces an output which is different from the previous one. To evaluate this parameter, a 360 bit secret key is used to encrypt the sample Bungee image of Fig. 5a is encrypted and Fig. 5b shows the encrypted output for this key.

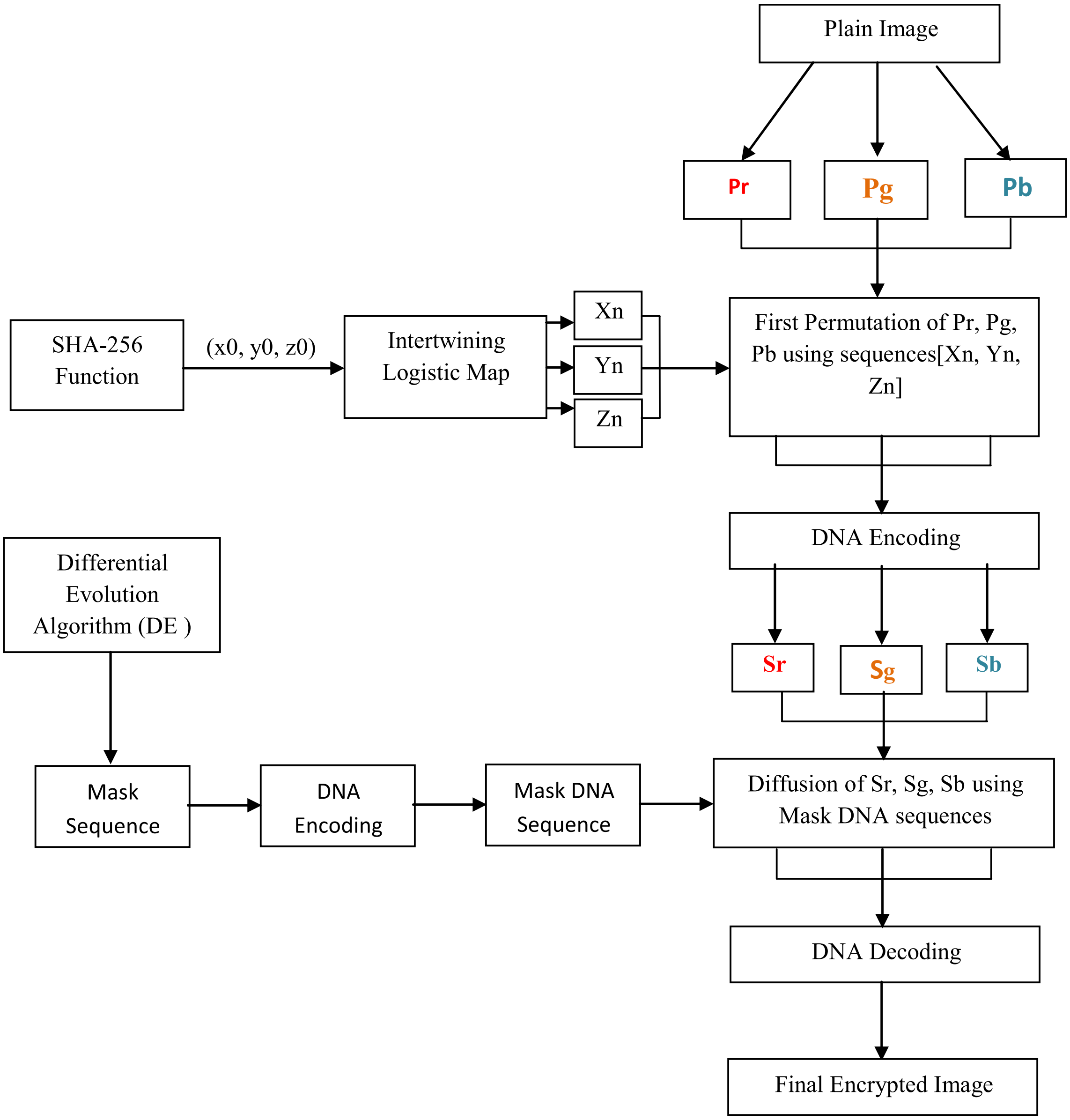

Plain Image
Pr
Pg
Pb
SHA-256 Function
(x0, y0, z0)
Intertwining Logistic Map
Xn
Yn
Zn
First Permutation of Pr, Pg, Pb using sequences[Xn, Yn, Zn]
DNA Encoding
Sr
Sg
Sb
Differential Evolution Algorithm (DE)
Mask Sequence
DNA Encoding
Mask DNA Sequence
Diffusion of Sr, Sg, Sb using Mask DNA sequences
DNA Decoding
Final Encrypted Image

Fig. 4 Flowchart of the proposed algorithm

Then, one-bit change is made in the original image and the altered one-bit image re-encrypted using the same secret key. Figure 6c shows the re-encryption results. Figure 6d shows the difference between the two cipher images 6b, c of Bungee. It can clearly be observed form the evaluation that the proposed technique is sensitive to secret keys and is able to resist exhaustive attacks. This also shows sensitivity of the encryption algorithm to the change of plain image as the two encrypted images 6b, c are generated from a plain image differing in only one pixel and yet the results are very efficient and highly independent of each other as illustrated in Fig. 6d. Apart from that, both the resultant encrypted images show expected values of parameters that are used to analyze the encryption algorithm of images which are discussed later in the paper, thus illustrating efficiency and sensitivity of encryption algorithm.

4.4 Differential attack

To perform this attack, two encrypted images are produced by the attacker by doing trivial changes in the original image, where one encrypted image is generated from the original image and the second encrypted image is produced from the changed original image. Hence, effort is made to establish a correlation by comparing the encrypted and the original image. Two parameters are used

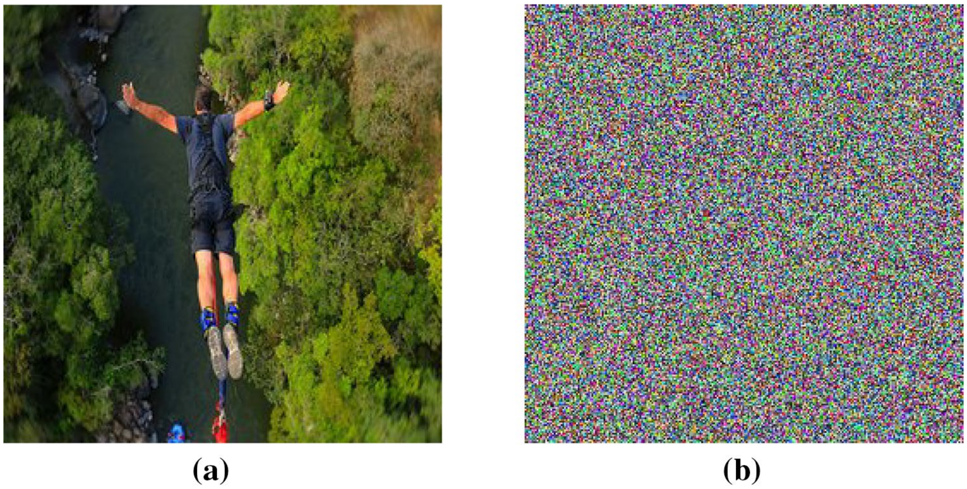


Fig. 5 **a** Plain bungee image. **b** Encrypted-image

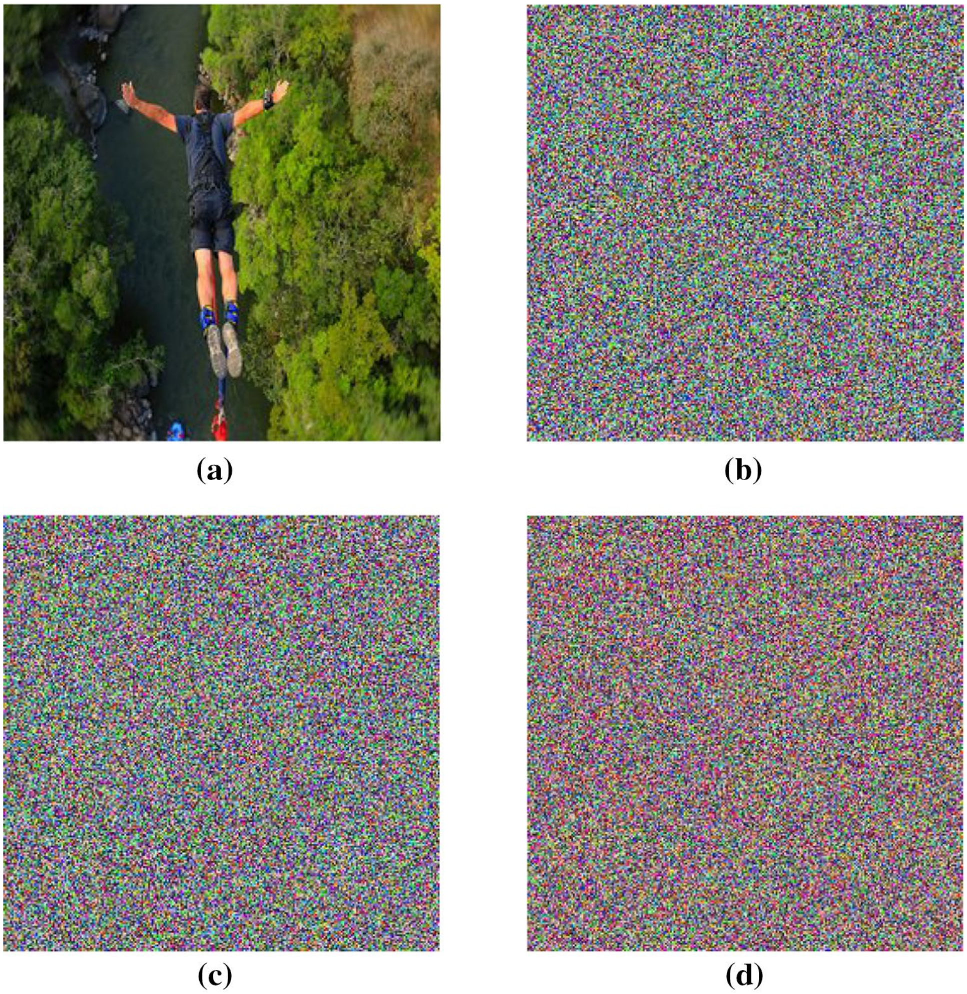


Fig. 6 **a** Sample bungee image. **b** Encrypted image of sample bungee image. **c** Re-encrypted image of one bit-altered sample bungee image. **d** Difference image of encrypted and re-encrypted image

for testing the degree of differential attack, viz. Unified average changing intensity (UACI), which is the percentage of the average change in intensity of corresponding pixels, and Number of pixel change rate (NPCR) which signifies the percentage of pixels in the encrypted images that changed. These are mathematically expressed as:

$$UACI = \frac{1}{P \times Q}\sum_{j=1}^{P}\sum_{k=1}^{Q}\frac{\left|\mathrm{c}_1(\mathrm{j},\mathrm{k}) - \mathrm{c}_2(\mathrm{j},\mathrm{k})\right|}{255} \times 100\% \tag{17}$$

$$NPCR = \frac{1}{P \times Q}\sum_{j=1}^{P}\sum_{k=1}^{Q}\mathrm{D}(\mathrm{j},\mathrm{k}) \times 100\% \tag{18}$$

where $\mathrm{D}(\mathrm{j},\mathrm{k})$ is given as

$$D(j,k) = \begin{cases} 1, & \mathrm{c}_1(\mathrm{j},\mathrm{k}) \neq \mathrm{c}_2(j,k) \\ 0, & otherwise \end{cases} \tag{19}$$

where two encrypted images are denoted by c_1, c_2 and the pixel value at index $[i,j]$ in the image is denoted by $c[i,j]$.

The evaluated values of UACI and NPCR are shown in Table 8. It can be observed from the results that the proposed encryption method is very close to ideal values that are above 99 for NPCR and about 30 for UACI. Thus, it can be concluded from the obtained values of these two parameters that the proposed technique ensures efficacy in resisting differential and plain-text attack effectively.

Table 9 shows an in-depth comparison of basic LM + DNA + DE and ILM + DNA + DE techniques on the basis of the parameters—NPCR, UACI, CC and entropy for Fig. 5a. High NPCR and UACI combined with low CC and an entropy closer to eight definitely prove a better chaotic efficiency of ILM as compared to LM.

4.5 Histogram analysis

Histogram delineates the frequency of pixel distribution throughout the image and is an integral statistical feature. Histogram is a plot of frequency of each pixel value in the image. Technically, a cipher image should have flat histograms in contrast to the steep slope of plain image histogram. This increases the level of randomness and makes it difficult to fetch information from images.

The results of Histogram evaluation have been shown in Table 10. It can be observed that encrypted images using proposed method are having regular distribution in comparison to the source image that has an irregular or non-uniform distribution. Hence, it proves the ability of the proposed method against statistical attack.

Table 9 Comparison with earlier proposed methods

Parameters	LM	ILM
NPCR	99.5973	99.6475
UACI	31.3682	32.9997
CC-horizontal	0.0548	0.0035
CC-vertical	0.0494	0.0468
CC-diagonal	– 0.0325	– 0.0493
Entropy	7.9959	7.9967
Contrast	10.7914	10.8104

Variance analysis is a quantitative analysis which is used for evaluating uniformity of encrypted images. It is a mathematical representation of histogram analysis. The value of variance is inversely proportional to the uniformity in encrypted images, i.e., lesser the variance, more is the uniformity in ciphered images (Zhang and Wang 2014). Variance is evaluated as:

$$var(X) = \frac{1}{n^2}\sum_{j=1}^{n}\sum_{k=1}^{n}\frac{1}{2}\left(x_j - x_k\right)^2 \tag{20}$$

where X is the vector for histogram values and $\mathrm{X} = \{\mathrm{x}_1, \mathrm{x}_2, \ldots, \mathrm{x}_{256}\}$, where x_i is the number of pixels with value equal to *i*. For color image, variance is calculated by taking average of variance of three matrix corresponding to R, G and B components.

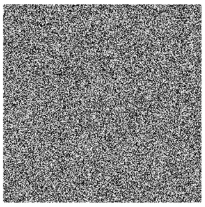
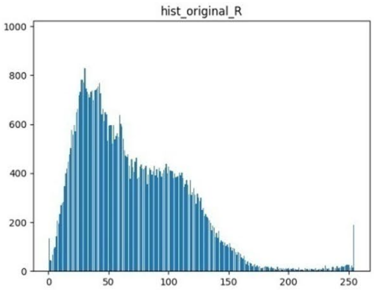
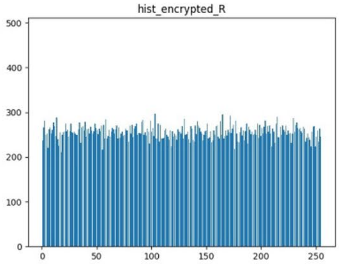
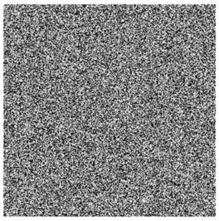
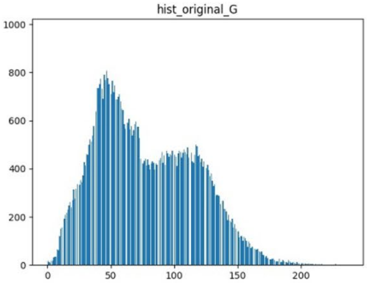
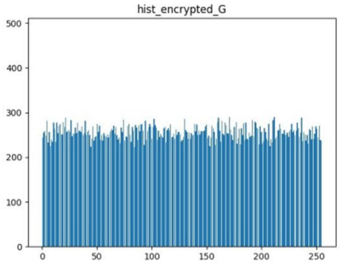
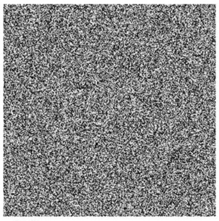
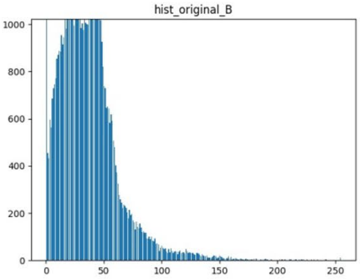
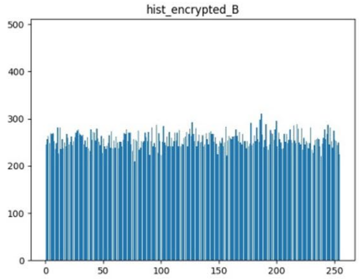
There are two ways for analysis using variance, first one is comparing variance of ciphered image with that of plain image. Variance of color bungee image was evaluated as 742,453.35 whereas the ciphered image has variance value 6497.44, thus showing more uniformity in ciphered image as compared to plain image.

Second method includes comparing variance value of multiple ciphered images resulting from encrypting the same bungee image with different encryption keys. All the

Table 8 Analysis parameters tabulated for color data-set images

Parameters	Component red (R)			Component green (G)			Component blue (B)		
	Lena	Bungee	Baboon	Lena	Bungee	Baboon	Lena	Bungee	Baboon
NPCR	99.6459	99.6475	99.6017	99.5590	99.6109	99.5956	99.5849	99.5590	99.6063
UACI	32.9558	32.9997	29.5848	30.4491	31.1707	27.9389	29.5262	32.2094	30.6470
CC-horizontal	– 0.0062	0.0035	– 0.0560	0.0502	0.0222	0.0209	– 0.0002	0.0473	0.0196
CC-vertical	– 0.0341	0.0468	– 0.0338	– 0.0243	0.0143	0.0261	– 0.0012	0.0238	– 0.0045
CC-diagonal	– 0.0091	– 0.0493	0.0011	0.0090	0.0132	– 0.0070	0.0350	– 0.0351	– 0.0386
Entropy	7.9985	7.9967	7.9968	7.9973	7.9970	7.9972	7.9969	7.9969	7.9969
Contrast	10.8621	10.8104	10.7894	10.7568	10.8562	10.8325	10.7965	10.9102	10.8235

Table 10 Histogram analysis

Color (R,G,B) Component	Ciphered Image	Original Image Histogram	Ciphered Image Histogram
	 Red-component		
	 Green-component		
	 Blue-component		

values were observed in the range 6300–6600, thus depicting the efficiency of algorithm in uniformity of ciphered images and making the statistical attacks useless for the proposed algorithm.

4.6 Correlation coefficient (CC) analysis

To establish and realize a linear association between two adjacent image pixels the term Correlation is used. Plain or original image pixels have a high correlation whereas an encrypted image should have low CC value. The correlation coefficient is given by the formula r_{xy}:

$$r_{xy} = \frac{cov(x,y)}{\sqrt{D(x)}\sqrt{D(y)}} \tag{21}$$

where

$$cov(x,y) = \frac{1}{S}\sum_{i=1}^{S}\left(x_i - E(x)\right)\left(y_i - E(y)\right) \tag{22}$$

$$D(x) = \frac{1}{S}\sum_{i=1}^{S}\left(x_i - E(x)\right)^2. \tag{23}$$

$$E(x) = \frac{1}{S}\sum_{i=1}^{S}x_i \tag{24}$$

where two successive pixels are denoted by x, y and randomly selected pixel pairs (x, y) are denoted by S. Expectation and variance of x are denoted by $E(x)$ and $D(x)$, respectively.

The results in Table 8 show the CC values-horizontal, vertical and diagonal obtained using the proposed method. The CC has been calculated by choosing randomly 1000 pairs of pixels from the image and then, using the duplets from this selected set of pixels to compute the coefficient value. The obtained results show that CC values are very low i.e. closer to zero, in case of encrypted images.

4.7 Resistance attack analysis

Sections 4.1 to 4.6 depict different metrics that help avoid the classical attacks (Wang et al. 2012; Bisht et al. 2018; Jaroli et al. 2018) based on the assumption that the mechanism of the cryptosystem is known thoroughly by a cryptanalyst barring the initial seed. The classical four types of attacks are mentioned below:

- *Ciphertext only* where the attacker has the knowledge of a couple of cipher texts.
- *Known plaintext* where the attacker has the knowledge of the plaintext and the corresponding cipher text.
- *Chosen plaintext* where the attacker has selective access of encryption system from which the corresponding ciphertext can be extracted from the chosen plaintext.
- *Chosen chipertext* where the attacker has selective access of decryption system from which the corresponding plaintext can be extracted from the chosen cipher text.

Since the proposed algorithm has a good key space and the chaos system is sensitive to the initial seed, the algorithm is resistant against chosen plaintext attack, which is one of the most common attacks. The DNA diffusion and optimization through DE along with ILM and SHA-256 make the cryptosystem more secure and resistant towards the aforementioned attacks.

4.8 Information analysis

Information analysis parameter termed as entropy is used to define the level of randomness or uncertainty. Low entropy signifies less ergodicity and high entropy signifies increase in the level of randomness (Bisht et al. 2019a, b). The ideal value for image entropy is eight and the numerical representation of entropy is defined as:

$$H(m) = \sum_{i=0}^{2^N-1} P\left(m_i\right) \log_2 \frac{1}{P\left(m_i\right)} \tag{25}$$

where the number of gray levels denoted by the variable N, and the total number of symbols are denoted by the variable M $(=2^n)$. Thevariable $m_i \in M$ and the variable $P\left(m_i\right)$ denotes the probability of having m_i levels in the image.

Table 8 shows the information entropy values of R, G, B components for the encrypted color Bungee image. The results justify the good information entropy values of the encrypted images.

4.9 Contrast analysis

This parameter is used to compute the intensity difference between the successive pixels of an image (Khan et al. 2015, 2017). In other words, this parameter enables a user to make distinction between various entities existing in the image. Hence, this parameter mainly emphasizes on intensity computation of a pixel and the computation is performed over the full image. The mathematical expression for contrast parameter is shown as (Khan et al. 2017):

$$C = \sum\nolimits_{i,j=1}^{N} |i-j|^2 p(i,j) \tag{26}$$

where gray-level co-occurrence matrices (GLCM) is defined by $p(i,j)$. Number of rows and columns are denoted by N. The evaluated results for the input image data set by using proposed method have been shown in Table 8.

4.10 Grayscale and binary image analysis

A grayscale image, commonly known as black-and-white image, is an image containing only one component, where each pixel depicts the intensity of light. Unlike grayscale image that has a range of 256 pixel values to choose from, binary images have only a choice of two pixel values (mostly black and white). However, like grayscale image it is also a type of digital image. The proposed technique of image encryption works for all three (grayscale, binary and color) types of images due to architecture flexibility.

4.11 Time comparison

The run time for the two EA algorithms i.e., GA and DE are compared by executing the programs on the experimental specifications described in Sect. 4.1. These EA algorithms depend on the two input parameters that are population size and number of iterations of the algorithm. More are the iterations or the population size, more is the time taken irrespective of the two algorithms. Though the time increases with these parameters, but the time taken by GA always remains manifolds higher than that of DE. The comparison between the two algorithms based upon time requirement is shown in Fig. 7.

4.12 Comparison with existing approaches

In the recent years, the researchers not only have used multi-dimensional chaos maps instead of one dimensional chaos maps, but have also combined the chaos maps with different techniques such as DNA, optimization methods, cellular automata to build more efficient and secure image encryption technique. The proposed work in this paper is also a perfect example of one such technique. Table 11 exhibits the comparison of the proposed technique with some of the earlier proposed image encryption algorithms that have used LM, DNA and genetic algorithms (Guesmi et al. 2016; Abdullah et al. 2012; Suri and Vijay 2017; Wang and Xu 2014). It also compares the approach with encryption techniques which involves manipulation of bits of pixels for encryption in their algorithms. One technique combines ILM with Reversible Cellular Automata (RCA), in which only higher 4-bit part of pixel is used as data to be encrypted (Wang and Luan 2013). Another technique involves cyclic shift of bits in pixel for encrypting the image data (Wang et al. 2015a, b, c). Motivated by the results of these method, the proposed algorithm in the paper contributes in two ways. Firstly, it engages the significant contributions presented by the aforementioned earlier methods. Secondly, and most importantly, it optimizes upon the former approaches by engaging the Evolutionary Algorithm that provide all the good features-augmented key space, high randomness and fast process. Thus, the proposed approach embodies the best of all the elements, providing an efficacious image encryption (Table 12).

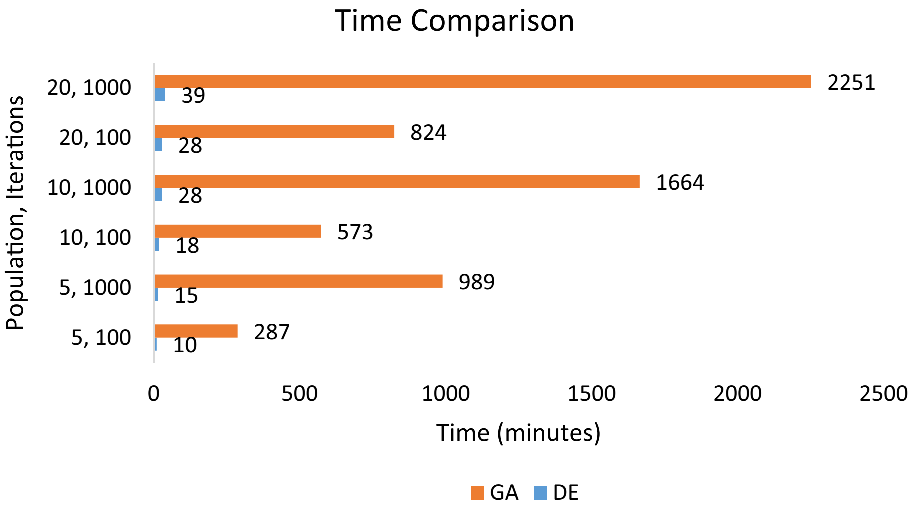


Fig. 7 Time comparison between DE and GA for different populations and iterations

Table 11 Analysis parameters tabulated for grayscale and binary images

Parameters	Grayscale			Binary		
	Lena	Bungee	Baboon	Lena	Bungee	Baboon
Contrast	10.8982	10.7985	10.8741	10.8550	10.8514	10.8927
Entropy	7.9965	7.9975	7.9970	7.9968	7.9972	7.9975
CC-vertical	− 0.0341	− 0.0031	0.0433	0.0092	0.0174	− 0.0014
CC-horizontal	− 0.0062	0.0222	− 0.0125	0.0148	− 0.0024	0.0217
CC-diagonal	− 0.0091	0.0421	0.0400	− 0.0124	0.0025	0.0128
UACI	32.9558	34.0691	30.6110	31.9746	36.7423	32.7832
NPCR	99.6459	99.6546	99.6322	99.6259	99.6320	99.6142

Table 12 Comparison with earlier proposed methods

Encryption technique and parameters	Proposed technique						
	Abdullah et al. (2012)	Enayatifar et al. (2014)	Guesmi et al. (2016)	Suri and Vijay (2017)	Wang and Luan (2013)	Wang et al. (2015a, b, c)	Proposed approach
Technique	LM + GA	LM + DNA + GA	Lorenz System + DNA	LM + DNA + Weighted GA	ILM	ILM	ILM + DNA + DE
Image type	Gray	Gray	Color	Gray	Gray	Gray	Color
Key space	120-bits	120-bit	SHA-2	120-bit	280-bit	3 × 256 × 256	SHA-2
Entropy	7.9978	7.9992	–	7.8091	7.9993	7.9984	7.9985
CC							
Vertical	0.0093	0.0072	0.0792	– 0.0586	0.0192	0.0342	– 0.0341
Horizontal	– 0.0054	0.0058	0.0265		0.0011	0.0096	– 0.0062
Diagonal	– 0.0009	0.0031	0.0625		0.0045	0.0205	– 0.0091
NPCR	97.1394	99.7103	–	99.707	99.79	99.82	99.64
UACI	33.1084	33.6297	–	30.5706	33.35	33.46	32.95
Contrast	–	–	–	–	–	–	10.8621
Observations	Optimization method GA enhances the performance of LM based approach. LM suffers from the problems of uneven distribution of sequences, stable and blank windows, and a weak key		SHA-2 increases key space and DNA increases the encryption efficiency	Using weighted GA gives balanced results with respect to applied fitness functions than using a single fitness function	Better sensitivity to initial values, more robust and resistant against common attacks, large key space, high level security and sensitivity to secret key		Use of ILM-DNA with DE optimization has provides faster time and better encryption parameter values than LM-DNA-GA based encryption parameters

5 Conclusion and future work

The algorithm has been proposed to develop an efficacious approach to obtain an optimized image encryption. The approach is further an integration of SHA-2 for generating the seed, ILM for permuting the image pixels by using the location map and DNA diffusion. Moreover, the algorithm is optimized with the help of DE that produces a mask sequence, which is further converted to DNA and utilized in DNA diffusion process. The optimization plays a crucial role in providing an efficient encryption. High entropy values and low CC values directly infer better results for an optimized encryption. The results of DE optimization are also compared with that of GA optimization. Theoretical analysis and experimental results reinforce that the algorithm using DE demonstrates better encoding efficiency than GA. The results also corroborate the fact that encryption using DE is faster than encryption using GA. Hence, DE can be used to obtain a quicker and more secure encryption process.

Compliance with ethical standards

Conflict of interest All the authors declare that the submitted manuscript and the authors do not have any conflict of interest.

References

Abdullah AH, Enayatifar R, Lee M (2012) A hybrid genetic algorithm and chaotic function model for image encryption. AEU Int J Electron Commun 66(10):806–816

Adleman LM (1994) Molecular computation of solutions to combinatorial problems. Science 266(5187):1021–1024

Alvarez G, Li S (2006) Some basic cryptographic requirements for chaos-based cryptosystems. Int J Bifurc Chaos 6(8):2129–2151

Bisht A, Jaroli P, Dua M, Dua S (2018) Symmetric multiple image encryption using multiple new one-dimensional chaotic functions and two-dimensional cat map. In: IEEE international conference on inventive research in computing applications (ICIRCA). Coimbatore, pp 676–682

Bisht A, Dua M, Dua S (2019a) A novel approach to encrypt multiple images using multiple chaotic maps and chaotic discrete fractional random transform. J Ambient Intell Hum Comput 10(9):3519–3531

Bisht A, Dua M, Dua S, Jaroli P (2019) A color image encryption technique based on bit-level permutation and alternate logistic maps. J Intell Syst

Chen G, Mao Y, Chui CK (2004) A symmetric image encryption scheme based on 3D chaotic cat maps. Chaos Solitons Fractals 21(3):749–761

Chen J, Zhou J, Wong K (2011) A modified chaos-based joint compression and encryption scheme. IEEE Trans Circuits Syst II Express Briefs 58(2):110–114

Chen Y-Y, Hsia C-H, Jhong S-Y, Lin H-J (2018) Data hiding method for AMBTC compressed images. J Ambient Intell Hum Comput 208:1–9

Enayatifar R, Abdullah AH, Isnin IF (2014) Chaos-based image encryption using a hybrid genetic algorithm and a DNA sequence. Opt Lasers Eng 56:83–93

Enayatifar R, Sadaei HJ, Abdullah AH, Lee M, Isnin IF (2015) A novel chaotic based image encryption using a hybrid model of deoxyribonucleic acid and cellular automata. Opt Lasers Eng 71:33–41

Enayatifar R, Abdullah AH, Isnin IF, Altameem A, Lee M (2017) Image encryption using a synchronous permutation-diffusion technique. Opt Lasers Eng 90:146–154

Fridrich J (1998) Symmetric ciphers based on two-dimensional chaotic maps. Int J Bifurc Chaos 8(6):1259–1284

Gómez J, Dasgupta D, González F (2003) Using adaptive operators in genetic search. In: Genetic and evolutionary computation conference, 2724, pp 1580–1581

Guesmi R, Farah M, Kachouri A, Samet M (2016) A novel chaos-based image encryption using DNA sequence operation and secure hash algorithm SHA-2. Nonlinear Dyn 83(3):1123–1136

Head T, Rozenberg G, Bladergroen R, Breek C, Lommerse P, Spaink H (2000) Computing with DNA by operating on plasmids. Biosystems 57(2):87–93

Ilonen J, Kamarainen J-K, Lampinen J (2003) Differential evolution training algorithm for feed-forward neural networks. Neural Process Lett 17(1):93–105

Jaroli P, Dua AB, Dua S (2018) A color image encryption using four dimensional differential equations and arnold chaotic map. In: IEEE international conference on inventive research in computing applications (ICIRCA). Coimbatore, pp 869–876

Joshi R, Sanderson A (1999) Minimal representation multisensor fusion using differential evolution. IEEE Trans Syst Man Cybern Part A Syst Hum 29(1):63–76

Julstrom BA (1995) What have you done for me lately? Adapting operator probabilities in a steady-state genetic algorithm. In: 6th international conference on genetic algorithm (ICGA). CINII, pp 81–87

Khade PN, Narnaware PM (2012) 3D chaotic functions for image encryption. IJCSI Int J Comput Sci Issues 9(3):1–6

Khan JS, Rehman AU, Ahmad J, Habib Z (2015) A new chaos-based secure image encryption scheme using multiple substitution boxes. In: Conference on information assurance and cyber security (CIACS), pp 16–21

Khan FA, Ahmed J, Khan JS, Ahmad JC, Khan MA (2017) A novel image encryption based on Lorenz equation, Gingerbreadman chaotic map and S8 permutation. J Intell Fuzzy Syst 33(6):3753–3765

Kumar M, Kumar S, Budhiraja R, Das MK, Singh S (2016) Intertwining logistic map and cellular automata based color image encryption model. In: IEEE international conference on computational techniques in information and communication technologies (ICCTICT). New Delhi, pp 618–623

Li S, Chen G, Cheung A, Bhargava B, Lo K (2007) On the design of perceptual mpeg-video encryption algorithms. IEEE Trans Circ Syst Video Technol 17(2):214–223

Liu H, Wang X (2010) Color image encryption based on one-time keys and robust chaotic maps. Comput Math Appl 59(10):3320–3327

Liu H, Wang X (2011) Color image encryption using spatial bit-level permutation and high-dimension chaotic system. Opt Commun 284(16–17):3895–3903

Liu H, Wang X, Kadir A (2012a) Image encryption using DNA complementary rule and chaotic maps. Appl Soft Comput 12(5):1457–1466

Liu L, Zhang Q, Wei X (2012b) A RGB image encryption algorithm based on DNA encoding and chaos map. Comput Electr Eng 38(5):1240–1248

Masuda N, Jakimoski G, Aihara K, Kocarev L (2006) Chaotic block ciphers: from theory to practical algorithms. IEEE Trans Circuits Syst I Regul Pap 53(6):1341–1352

Ott E, Grebogi C, Yorke JA (1990) Controlling chaos. Phys Rev Lett:2837

Qin AK, Huang VL, Suganthan PN (2009) Differential evolution algorithm with strategy adaptation for global numerical optimization. IEEE Trans Evol Comput 13(2):398–417

Rhouma R, Safya B (2008) Cryptanalysis of a new image encryption algorithm based on hyper-chaos. Phys Lett A 372(38):5973–5978

Sneha PS, Sankar S, Kumar AS (2019) A chaotic colour image encryption scheme combining Walsh-Hadamard transform and Arnold-Tent maps. J Ambient Intell Hum Comput 2019:1–20

Solak E, Çokal C (2011) Algebraic break of image ciphers based on discretized chaotic map lattices. Inf Sci 181(1):227–233

Solak E, Çokal C, Yildiz OT, Biyikoglu T (2010) Cryptanalysis of fridrich's chaotic image encryption. Int J Bifurc Chaos 20(5):1405–1413

Storn R (1995) Differrential evolution-a simple and efficient adaptive scheme for global optimization over continuous spaces. Technical report, International Computer Science Institute 11

Storn R (1996) On the usage of differential evolution for function optimization. In: Proceedings of North American fuzzy information processing, pp 519–523

Suneja K, Dua S, Dua M (2019) A review of chaos based image encryption. In: IEEE 3rd international conference on computing methodologies and communication (ICCMC). Erode, pp 693–698

Suri S, Vijay R (2017) A bi-objective genetic algorithm optimization of chaos-DNA based hybrid approach. J Intell Syst 28(2):333–346

Suri S, Vijay R (2019) A synchronous intertwining logistic map-DNA approach for color image encryption. J Ambient Intell Hum Comput 10(6):2277–2290

Tuson A, Ross P (1998) Adapting operator settings in genetic algorithms. Evol Comput 6(2):161–184

Wang l, Luan D (2013) A novel image encryption algorithm using chaos and reversible cellular automata. Commun Nonlinear Sci Numer Simul 18(11):3075–3085

Wang X, Xu D (2014) Image encryption using genetic operators and intertwining logistic map. Nonlinear Dyn 78(4):2975–2984

Wang X-Y, Yang L, Liu R, Kadir A (2010) A chaotic image encryption algorithm based on perceptron model. Nonlinear Dyn 62(3):615–621

Wang X, Teng L, Qin X (2012) A novel colour image encryption algorithm based on chaos. Signal Process 92(4):1101–1108

Wang X, Liu L, Zhang Y (2015a) A novel chaotic block image encryption algorithm based on dynamic random growth technique. Opt Lasers Eng 66:10–18

Wang X-Y, Gu S-X, Zhang Y-Q (2015b) Novel image encryption algorithm based on cycle shift and chaotic system. Opt Lasers Eng 68:126–134

Wang X-Y, Zhang Y-Q, Bao X-M (2015c) A novel chaotic image encryption scheme using DNA sequence operations. Opt Lasers Eng 73:53–61

Wang X, Feng L, Zhao H (2019) Fast image encryption algorithm based on parallel computing system. Inf Sci 486:340–358

Xiao G, Lu M, Lai XQ (2006) New field of cryptography: DNA cryptography. Chin Sci Bull 51(12):1413–1420

Zhang Y (2015) Cryptanalysis of a novel image fusion encryption algorithm based on DNA sequence operation and hyper-chaotic system. Optik 126(2):223–229

Zhang Y, Fu LH (2012) Research on DNA cryptography. In: Sen J (ed) Applied cryptography and network security. Rijeka, Intechopen, pp 357–376

Zhang Y-Q, Wang X-Y (2014) A symmetric image encryption algorithm based on mixed linear–nonlinear coupled map lattice. Inf Sci 273:329–351

Zhang Y-Q, Wang X-Y (2015) A new image encryption algorithm based on non-adjacent coupled map lattices. Appl Soft Comput 26:10–20

Zhang Q, Guo L, Wei X (2010a) Image encryption using DNA addition combining with chaotic maps. Math Comput Model 52(11–12):2028–2035

Zhang Q, Wang Q, Wei X (2010b) A novel image encryption scheme based on dna coding and multi-chaotic maps. Adv Sci Lett 3(4):447–451

Zhang Y, Li C, Li Q, Zhang D, Shu S (2012) Breaking a chaotic image encryption algorithm based on perceptron model. Nonlinear Dyn 69(3):1091–1096

Zhang Q, Guo L, Wei X (2013) A novel image fusion encryption algorithm based on DNA sequence operation and hyper-chaotic system. Optik Int J Light Electron Opt 124(18):3596–3600

Zhang Y-Q, Wang X-Y, Liu J, Chi Z-L (2016) An image encryption scheme based on the MLNCML system using DNA sequences. Opt Lasers Eng 82:95–103

Zheng X, Xu J, Li W (2009) Parallel DNA arithmetic operation based on n-moduli set. Appl Math Comput 212(1):177–184