Chaos, Solitons and Fractals 164 (2022) 112628

Contents lists available at ScienceDirect

Chaos, Solitons and Fractals

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

PixAdapt: A novel approach to adaptive image encryption

Rohan Tuli [a,*], Hitesh Narayan Soneji [b], Prathamesh Churi [b]

[a] Engineering and Informatics Department, University of Sussex, Brighton, United Kingdom
[b] Computer Engineering Department, Mukesh Patel School of Technology Management & Engineering NMIMS University, Mumbai, India

ARTICLE INFO

Keywords:
PixAdapt
Hill climb
Genetic algorithm
Simulated annealing
UACI
Chaos

ABSTRACT

Image encryption using genetic approach is a recent and advanced technique which has grabbed attention in recent years. Currently, most image encryption algorithms (using genetic approach) use a static set of parameters for image encryption without considering the features representative of the image. In this study, an innovative adaptive image encryption algorithm – PixAdapt is developed. The process of image encryption is being re-engineered in a way to calculate the fitness of encrypted image using UACI and adapting the respective parameters using genetic hill climb or simulated annealing. Pseudorandom numbers have been generated using the linear feedback shift register and chaos-based maps such as the Logistic map, Rossler map, Henon map and Tent map. PixAdapt algorithm also uses confusion and diffusion process to ensure that plain text image and cipher text image are completely un-related. The use of metaheuristic search techniques for optimization of image encryption parameters has been implemented for the first time. The results obtained show that the genetic hill climb algorithm encrypts the various images giving the most optimal value of UACI. The algorithm has been tested for fitness improvement, parameter evolution, statistical analysis, and quality of encryption. PixAdapt is not only unique but has proven the encryption parameter UACI to be an appropriate fitness function to encrypt an image efficiently.

1. Introduction

There are several types of images available varying from application to application [1,2]. Each of these images have unique features in their own regard. Images from the medical field and satellite images are much larger and higher dimensional than trivial JPEG and PNG images [3]. Despite the characteristics of the image, the security of the image while transferring through media remains vital [4,5]. Image encryption is a process of converting the original image into an unreadable image. Tremendous amount of research has been conducted in the usage of chaotic sequences for encrypting an image. These sequences are generated using a static set of parameters which do not account for the characteristics of the actual image. Currently, very few encryption methods adapt to the different types of images they are encrypting. The lack of an adaptive mechanism sparked curiosity and an effort to re-engineer the process of encrypting an image with the help of a metaheuristic search algorithm to find the right pairs of parameters to encrypt an image has been made in this paper. Metaheuristic search algorithms provide a solution to solve complex optimization problems in a relatively short span of time [6]. In this paper, two such algorithms have been explored and implemented namely: hill climb and simulated annealing. Hill climb has been implemented as a genetic algorithm while simulated annealing has been used in its primitive essence. Chaotic sequences such as Logistic map, Rossler map, Henon map and Tent map have been used to generate pseudorandom numbers in conjunction with the Linear feedback shift register.

An adaptive system in a more general sense can be defined as a system created with the mind set of dealing with and adapting to changes in the environment while optimizing performance objectives [7]. According to this definition, the proposed system does deal with changes in its environment which are the new types of images it encounters at every pass. As the environment changes i.e., the images change, the adaptive system calculates the fitness of the parameters for respective image and decides whether the parameters are required to be evolved further [8–11]. The overall process has the objective of maintaining the performance by creating a solution which lies in the acceptable fitness range. The proposed mechanism also fits the definition of an adaptive system as proposed by [12] where they described an adaptive system to satisfy natural selection in accordance with the following conditions of varying entities, having continuity and the

* Corresponding author.
E-mail addresses: rt349@sussex.ac.uk (R. Tuli), hiteshsoneji25@gmail.com (H.N. Soneji), Prathamesh.churi@ieee.org (P. Churi).

https://doi.org/10.1016/j.chaos.2022.112628
Received 11 April 2022; Received in revised form 23 August 2022; Accepted 25 August 2022
Available online 19 September 2022

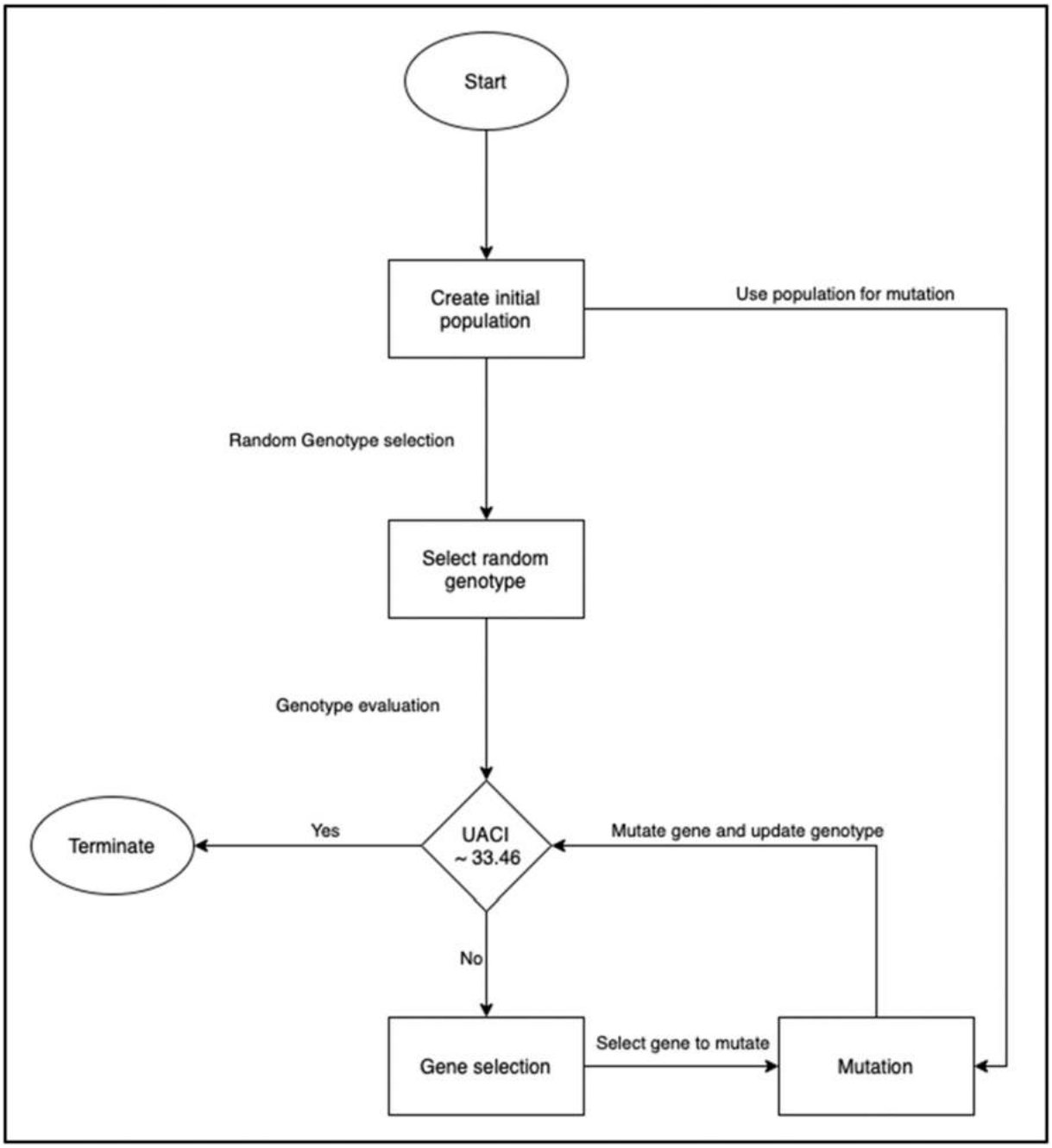


Fig. 1. Hill climb algorithm.

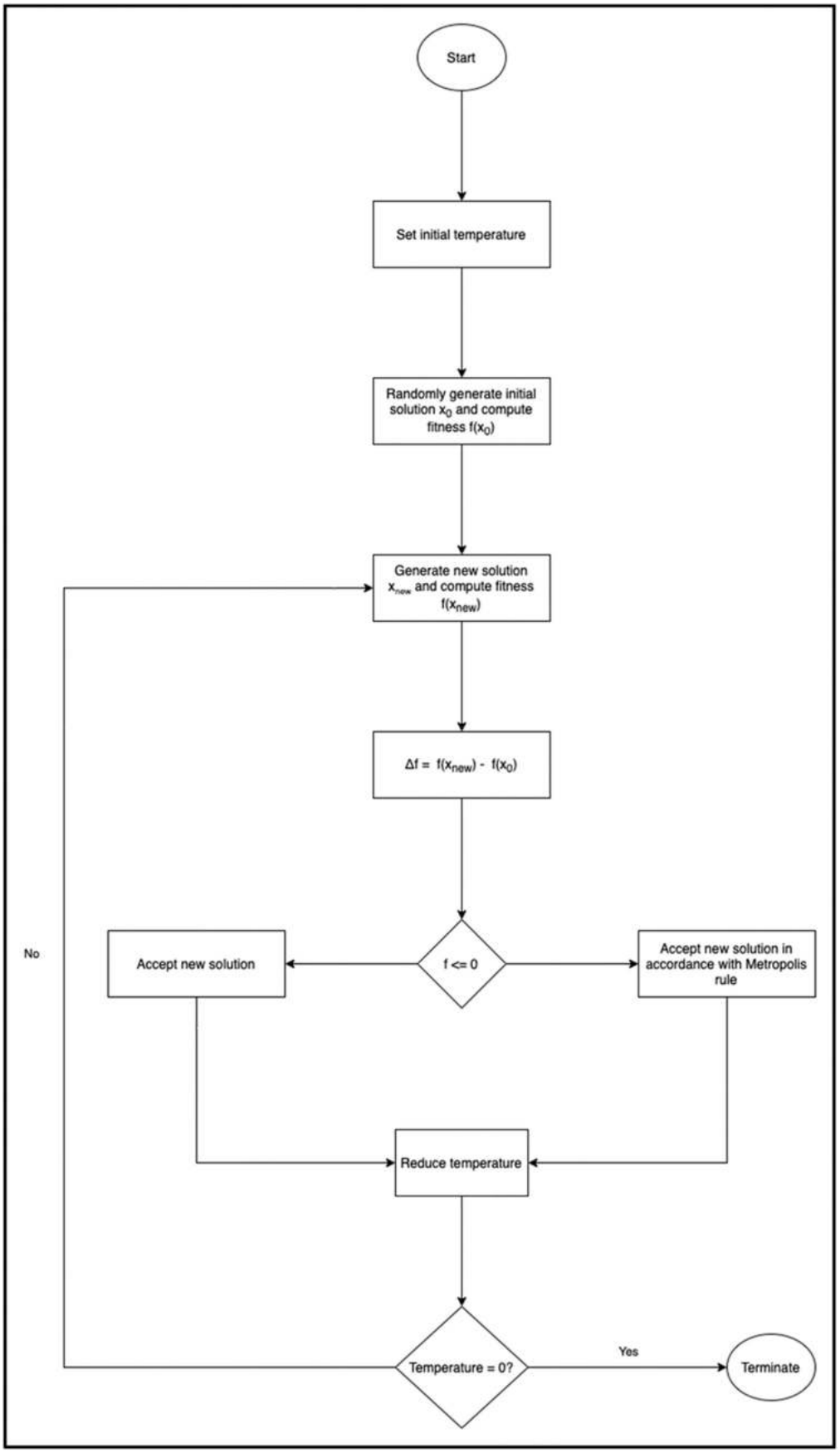


Fig. 2. Simulated annealing.

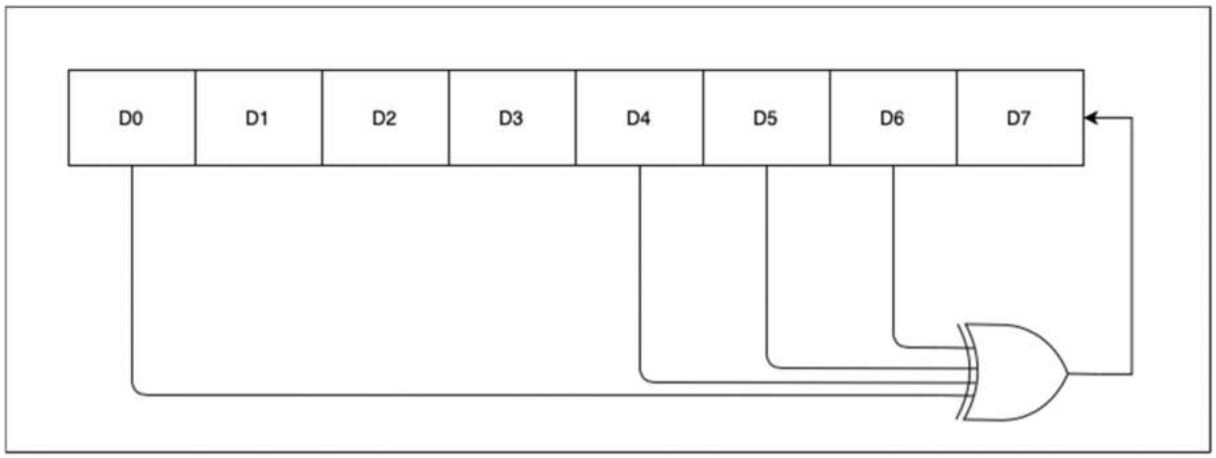


Fig. 3. Linear feedback shift register.

success of each entity differs from the other. Each of parameters used to generate chaotic sequences have varying values after each pass. These values are either derived from each other or they are continually evolved depending on the method used. After each pass, a new entity is discovered with the aim of optimizing the fitness value. To be more specific, the proposed mechanisms can also be defined as an adaptive system which changes its behaviour with the aim of accommodating changes in its environment [13]. When the necessary conditions are met to adapt the system, the current state is evolved into a new state. This definition is to do with the encryption system to be in an idle state until it comes across a set of parameters which do not produce the required fitness. The system will check for the fitness generated and only intervene when the fitness is below par.

To make the system adaptive, metaheuristic search algorithms such as genetic hill climb, and simulated annealing have been used. When the system is not producing the correct fitness, the adaptiveness becomes active and starts to evolve parameters in accordance with the algorithms. This approach has also been extended to encrypt a sequence of images which evolves the image solutions and provides a feedback mechanism to either increase or decrease the parameters acting as a negative or positive feedback in the respective cases.

To Sum up, our research seeks following research objectives which can also be classified as the advantages and novelty proposed through this algorithm:

Objective 1: To develop a novel adaptive image encryption algorithm **- PixAdapt** which uses genetic hill climb or simulated annealing algorithm.

Objective 2: To propose a fitness function with the aim of optimizing UACI value of an image using genetic hill climb or simulated annealing. Appropriate experimentation has been designed to verify the same.

Objective 3: To generate appropriate pseudorandom sequences - PixAdapt uses linear feedback shift register and chaos-based maps. A switching mechanism has also been proposed to generate the pseudo-random sequence.

Objective 4: To develop a confusion-diffusion process for PixAdapt.

Objective 5: To testify the use of metaheuristic search techniques for optimization of PixAdapt algorithm.

PixAdapt which is an acronym of the principle “Pixel Encryption through genetic adaption”. The algorithm uses genetic approach as well as it produces correct fitness and adapt to the best value of the parameters (say, UACI). The aim of the fitness function is to evolve parameters until they generate an acceptable fitness score hence adaption. The proposed method in the near future can prove to be a pioneer to encrypt an image using adaptiveness. PixAdapt is Symmetric genetic image encryption algorithm based on substitution-confusion-diffusion. The substitution process is governed by usage of various maps used in this algorithm and confusion-diffusion process is governed by regeneration of binary sequences. The process of diffusion is also explained and explored in [14,15].

The sections of the paper are as follows: Section 2 describes the background of this research. Section 3 describes the current trends in image encryption using non-adaptive and genetic approaches. The Section 4 consists of the methods used during the experimentation and overall proposed image encryption approaches. Section 5 dives deep into the results obtained from the proposed methodology. Section 6 discusses the results obtained from the respective proposed schemes while Section 7 concludes the research.

2. Background

2.1. Genetic algorithm

A genetic algorithm is an evolutionary model inspired by biological evolution based on the natural selection theory proposed by Darwin [16]. A population of genes for every parameter in the system is created. These algorithms initially use a genotype obtained from the population for the process which is then evaluated using a fitness function. This fitness function determines whether the genotype is required to undergo mutation, or the system has reached convergence.

2.1.1. Hill climb

This is a simple yet efficient evolutionary algorithm that uses chromosomes to model possible configurations of parametric sequences of weights and biases [17]. Fig. 1 depicts the process of hill climb. In this process, a population of genes is initialized. Genes from the initial population are randomly selected to perform the given task. After the task is completed, the fitness of the system is calculated. If the fitness is within the acceptable range, the algorithm terminates. A gene is only selected for mutation if it has a better fitness score than the previously highest scoring gene and less than the optimal value. If the fitness is not in the acceptable range, a random gene from the genotype is selected and mutated. The selected gene is replaced by another gene from the respective gene pair initialized at the beginning of the process. The fitness of the newly obtained genotype is then calculated. The process continues until a genotype in the acceptable range is generated.

For experimentation purposes, the initial population for analysis has been kept at 100. The fitness of each genotype is calculated by encrypted images with the respective genotype parameters and calculated UACI. The results obtained from each of these have been discussed later in the paper.

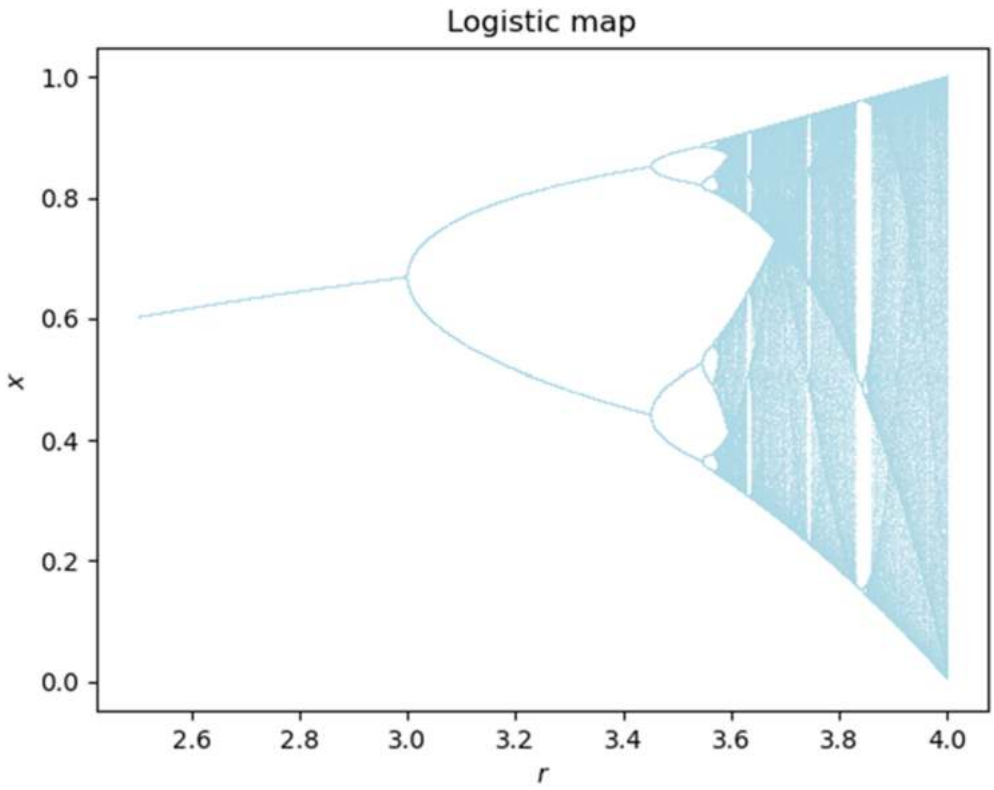


Fig. 4. Logistic map.

2.1.2. Simulated annealing

Proposed by [18], this process is a modified version of the hill climb algorithm. The algorithm has been inspired by the metallurgical process called annealing where any metal at a high temperature has the tendency to shift its internal atoms but at a lower temperature, the metal is in a more stable configuration. Using this property, the algorithm can be described as a stochastic global search optimization. Fig. 2 describes the working of Simulated annealing. An initial high temperature is set at the start of the algorithm which governs the overall behaviour of the system. Initially, a random initial solution between the parametric range is generated and its fitness is calculated. After which, another solution is generated, encrypted with those parameters and its respective fitness is calculated. The fitness of both the solutions are compared and accepted in the following manner:

$$f(x) = \begin{cases} f(x_0), f(x_0) \rangle f(x_{new}) \\ f(x_{new}), f(x_0) \langle f(x_{new}) \\ f(x_{new}), rand < metropolis\ factor \end{cases}$$

where the metropolis factor is

$$Metropolis = e^{-\left(\frac{CandidateEval - CurrentEval}{Temperature}\right)}$$

When the new candidate solution performs better than the previous candidate solution, the candidate is accepted or else, rejected. In some cases, a random number is drawn from a normal or uniform distribution. If the random number is greater than the metropolis factor, then the candidate is accepted irrespective of its performance. Thus, this overcomes the problem of stagnation at local optima point by randomly selecting a lesser fit candidate in accordance with the metropolis factor. Here, temperature plays a key role in determining the metropolis factor as well as the number of overall epochs the algorithm is going to perform. After each pass, the temperature is decreased by a small factor.

2.2. Pseudo-random sequence generation

Pseudorandom sequences are numbers that mimic the property of random numbers and are deterministic in nature. These sequences are generated using an initial seed point and may include another set of parameters that can be varied to generate more random sequences. Sequences generated using these methods are very useful in cryptography and especially in Image encryption. An important property of using pseudorandom numbers for encryption is that, given the correct set of parameters and seed values, the pseudorandom number can be replicated with relative ease. The sequence with which an image is encrypted can easily be replicated at the destination with the correct set of parameters and seed values.

2.2.1. Linear feedback shift register

A linear feedback shift register can be used to generate a sequence of random numbers based on the configuration of the pins in conjunction with the initial seed point. For generating sequences which mimic the pixel intensity of images, using an 8-bit linear feedback shift register with polynomial feedback is very useful. For experimentation purposes, the feedback polynomial used is $x^8 + x^6 + x^5 + x^4 + 1$ which has been depicted in Fig. 3.

The 0th, 4th, 5th and 6th are XORed, and the 7th bit is replaced with the output bit while moving every bit by 1 in the left direction. The 0th bit is discarded, and the new 8-bit binary sequence is used to create the next new number. This process is repeated until the generated sequence matches the number of pixels in the image. The seed value during experimentation has been varied between 0 and 255.

2.2.2. Chaotic maps

Chaos is a field of mathematics where initial seed conditions and parameters govern the behaviour of sequences that look irregular and disoriented but produce results like that or random events [19]. The random behaviour of sequences can be explained using chaos theory. Systems that produce chaotic sequences can be called as chaotic maps.

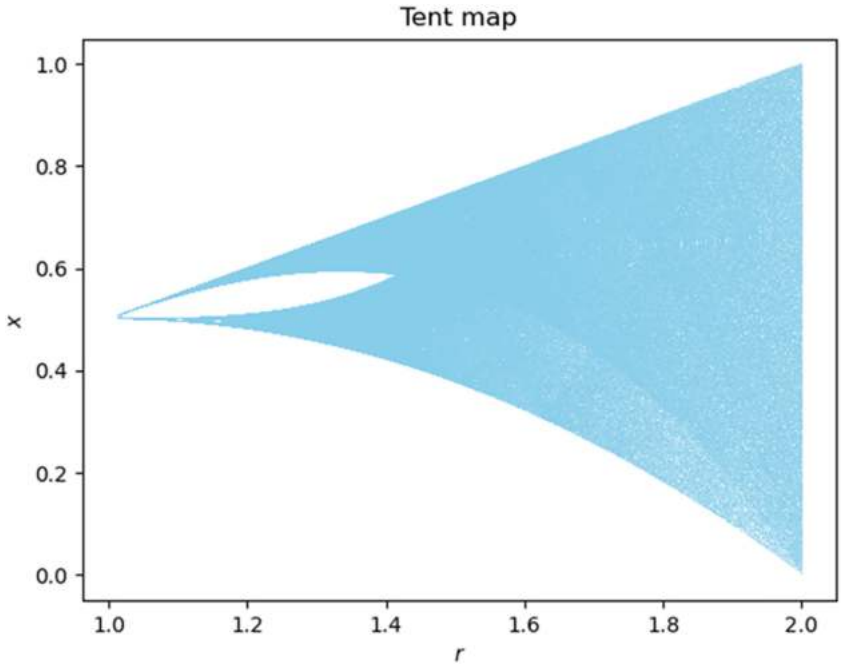


Fig. 5. Tent map.

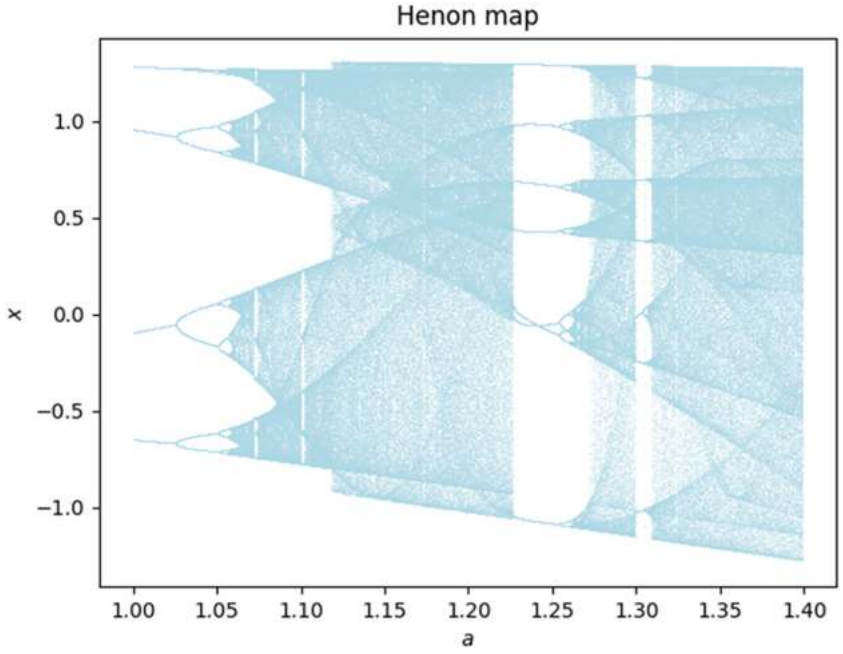


Fig. 6. Henon map.

These maps use feedback loops, reliability, and the fractal nature of the system to produce such sequences. Most chaos-based encryption systems use a single chaotic map for sequence generated while not considering the effect of the sequences on the respective images [20]. Additionally, these algorithms usually use the most chaotic range of these maps to generate pseudo-random numbers used for encryption thus, using a static value for all images. Using a static set of parameters for encrypting images can lead to inefficient encryption in some cases and this problem can be overcome by using more than one pseudo-random sequence and determining the quality of encryption for these parameters. The proposed algorithm uses four chaotic maps which are: The logistic map, Tent map, Henon map and Rossler map.

- Logistic map

This function has a second degree of polynomial mapping. It is a single variable, discrete-time system which exhibits chaos in selecting suitable "r" values. The mathematical function is as follows:

$$x_{n+1} = rx_n(1 - x_n)$$

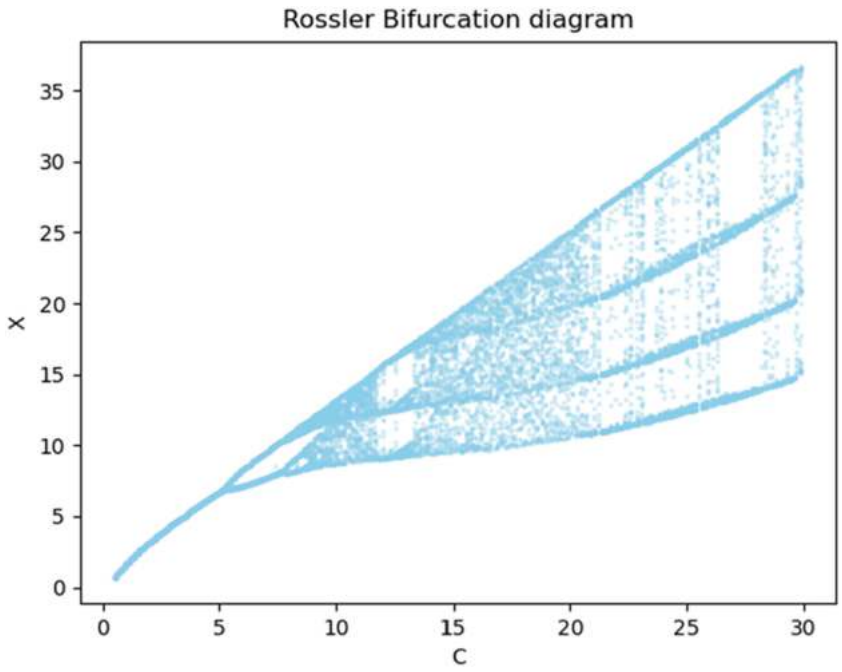


Fig. 7. Rossler map.

where x_n is the current value and x_{n+1} is the next value. The parameter r is known as the reproduction rate. Fig. 4 shows the behaviour of the logistic map on changing values of r. According to this figure, the function shows chaotic behaviour only when the r value is in the range of 3.5 to 4. For experimentation, in the Genetic Hill climb algorithm, the genes are generated between 3.6 and 4 while for the simulated annealing approach, a random value is generated within this range, and it is continuously modified by adding or subtracting a small value obtained from a Gaussian distribution with its mean and sigma values equal to 0.1 and 0.01 respectively. The initial seed value for a genetic algorithm is generated between 0.1 and 1. For the simulated annealing method, the seed value is generated between the range of 0.1 and 1 while it's continuously modified after each epoch by a random value from a Gaussian distribution with mean and sigma values equal to 0.1 and 0.2 respectively.

- Tent map

This function uses a single degree of polynomial mapping along with a parameter "r". This system can be classified as a discrete-time dynamical system. Its equation is as follows:

$$x_{n+1} = \begin{cases} rx_n, & x_n < \frac{1}{2} \\ r(1 - x_n), & x_n \geq \frac{1}{2} \end{cases}$$

where x_n is the current value and x_{n+1} is the next value. The parameter r is a real positive constant. The behaviour of the tent map is like that of the logistic map. The tent map exhibits chaotic behaviour in the range of r values above 1 and shows very chaotic behaviour at r values equal to 2 as shown in Fig. 7. For experimentation, the r value is varied in the range of 1 to 2 and the seed pixel value is varied between 0.1 and 1. In the genetic hill climb approach, genes of size 10 and 100 for each of the parameters and generated while for the Simulated Annealing approach, the initial value is chosen at random between the given range and the value for the next iteration is modified by a random value drawn from a Gaussian distribution of mean and sigma equal to 0.1 and 0.2 for the seed parameter and 0.1 and 0.01 for the r parameter respectively (see Fig. 5).

- Henon map

The Henon map is a two-dimensional discrete-time dynamical system using two parameters a and b. The Henon map exhibits chaotic behaviour when the parameter b is kept constant and a is varied in the range of 1 and 1.45. The equation is as follows:

$$\begin{cases} x_{n+1} = 1 - ax_n^2 + y_n \\ y_{n+1} = bx_n \end{cases}$$

where x_n is the current value and $x_n + 1$ is the next value, y_n is the current value and $y_n + 1$ is the next value. Parameters a and b govern the behaviour of the system. Fig. 8 shows the chaotic performance of the Henon map when b = 0.3 and a is varied. For experimentation, the parameters x, y and a are varied in a range of 0 to 1, 0 to 1 and 1 to 1.45 respectively. In the genetic hill climb approach, genes of size 10 and 100 for each of the parameters and generated while for the Simulated Annealing approach, the initial values chosen at random between the given range and the value for the next iteration is modified by a random value drawn from a Gaussian distribution of mean and sigma equal to 0.1 and 0.2 for the seed parameters (x and y) and 0.1 and 0.01 for the parameter a respectively (see Fig. 6).

- Rossler map

The Rossler map or attractor is a three-dimensional continuous-time dynamical system. The system also has three parameters a, b and c which are used to exhibit chaotic behaviour. The system is defined by three non-linear ordinary differential equations which are as follows (see Fig. 7):

Table 1
Details of sample images for image encryption.

Name	Size	Congruity
Image_1	(256, 256)	Symmetrical
Image_2	(300,300)	Symmetrical
Image_3	(300, 300)	Symmetrical
Image_4	(512, 512)	Symmetrical
Image_5	(800, 600)	Asymmetrical
Image_6	(512, 512)	Symmetrical
Image_7	(492, 600)	Asymmetrical
Image_8	(594, 670)	Asymmetrical
Image_9	(1990, 1342)	Asymmetrical
Image_10	(1026, 1024)	Asymmetrical

$$\begin{cases} \dfrac{dx}{dt} = -y - z \\ \dfrac{dy}{dt} = x + ay \\ \dfrac{dz}{dt} = b + z(x - c) \end{cases}$$

where x, y and z are coordinates in the three dimensions and, a, b and c are parameters that govern the behaviour of the system. Fig. 9 shows the chaotic behaviour of the Rossler map for the x dimension and the parameter c. For experimentation, parameters a and b were set to 0.2 and parameter c was varied. To obtain high chaoticity, parameter c was varied for values above 9 in the genetic hill climb and simulated annealing algorithm approach. For the simulated annealing approach, the c value was modified with a value obtained from a Gaussian distribution with mean and sigma values equal to 0.1 and 0.01 respectively.

3. Literature review

This section acknowledges the work done in image encryption using an adaptive approach. To the best of the knowledge, there were very few works found which involved an adaptive system to improve the performance of an encryption algorithm. Most of these algorithms showed static behaviour and did not take into the account the computation time complexity.

The initial work on genetic algorithm-based image encryption started by focusing the basic operations of genetic algorithm i.e. selection, crossover and mutation. Wang and Xu [21] proposed an encryption algorithm which uses logistic map to generate chaotic sequence. However, the algorithm lacks in doing extensive security analysis. Das et al. [22] proposed an encryption algorithm to encrypt multiple images at the same time. The proposed algorithm had less execution time as compared to other genetic encryption algorithms. Ghazvini et al. [23] proposed algorithm with genetic map and piecewise linear chaotic map (PWLCM). PWLCM has a strong base of confusion and diffusion process which algorithm resistant to brute force, statistical and differential attack. Shankar and Eswaran [24] proposed novel approach of using ECC (Elliptical curve cryptography) algorithm to generate key used in image encryption. The private key of decryption algorithm is generated by genetic algorithm. The algorithm lacks in experimenting in security analysis and more parameters on statistical analysis.

Over a period of year, perception of sensitivity of the medical images has drawn more attention in the community. To augment this, Pareek and Patidar [25] proposes genetic based image encryption algorithm which transforms plain image to newly formed image by basic genetic algorithm operations. The algorithm proven to be resistant to brute-force attack, differential attack, plaintext attack and entropy attack prominently. Enayatifar et al. [26] used the DNA sequence and logistic map along with genetic algorithm to encrypt images. However, the algorithm could be enhanced by proving the computational complexity of the proposed algorithm. Abdullah et al. [27] proposed dynamic behaviour wherein enciphered images are constructed using original images. The algorithm used highest value entropy by keeping low error rate.

Abbasi et al. [28] proposed an algorithm which has highest value of entropy through experimental analysis. The algorithm used lattice map function along with GA (genetic algorithm) to obtain the results. The algorithm achieved 2120 key space value which is resistant to brute force attack. Wang [29] focused on sensitivity of the plaintext through scrambling. The said algorithm did not reveal the computational complexity. The algorithm was resistant to brute-force and statistical attack. Abbasi et al. [28] proposed an algorithm based on chaos sequence and wavelet transform. The algorithm did not promise that the algorithm's time complexity is efficient or not. Wong et al. [30] have done detailed study on Cryptanalysis theory on image encryption algorithm. It has proven that the encryption scheme is not as secure as Biswas et al. [31] proposed an algorithm with the help of parallel processing capability for multiple bit-planes encryption. The algorithm achieved good encryption speed and is suitable for real time application. [32] proposes an algorithm with 1045 key space sensitivity with acceptable speed performance 4.7081Mbt/s and resistance to various attacks.

In the year of 2017, Kaur and Kumar [33] proposes encryption algorithm with beta chaotic map, nonsubsampled contourlet transform, and genetic algorithm. The algorithm is better at computational speed and high encryption intensity. [34] proposes improvised parallel Non-Dominated Sorting Genetic Algorithm (NSGA-II)-based encryption algorithm which has less computational complexity. [35] proposes an encryption algorithm with 7.99 entropy. The author claimed that the algorithm will also be used in maintaining the privacy of the images.

[36] proposes an algorithm exclusively for medical images which has better decreasing execution time. The algorithm achieved 2^{128} key space attack. Algorithm could be enhanced by experimenting better analysis on attacks. [37] proposes PWLCM chaotic map to perform bit level encryption algorithm. SHA-1 algorithm is used to encrypt the key of an algorithm. It is observed that security and efficiency of an algorithm (in just one round) is better and encryption and decryption time is low. [38] proposed an algorithm with the use of RNA code truth table which forms initial population of genetic algorithm. The algorithm claimed to be high resistance of attacks on 256×256 image after 30 repetitions. [39] followed the 2D non-linear couple lattice map and basic genetic operations to develop novel image encryption technique. The prosed technique had better computational complexity and achieved better key space. [40] proposes the color image encryption system by integrated non-dominated sorting algorithm and basic chaotic map is used to tune the hyper parameters of 5D chaotic map. The algorithm achieved the better UACI, Entropy values as compared to exiting techniques. An Inter-twinning logistic map is used in [41] for image encryption. The primary outcome of the research is that efficiency of Differential evolution is outperformed than genetic algorithm.

Recently, the paper [42] uses quantum genetic algorithm (QGA) and compressive sensing (CS) for encryption of images. The algorithm has been found to be resistant to statistical attack and plaintext attack as compared to other algorithms. [23] uses Chen's chaotic map and Logistic-Sine map for the process of confusion and diffusion. [43] uses master slave model for image encryption to improve its computational complexity. [44] uses Keccak algorithm to generate chaotic values. Genetic operations are managed by Henon map and DNA coding. The algorithm was found to be resistant to statistical and differential attacks. [45] invented Pareto-optimal image encryption algorithm which is evaluated against standard parameters such as key sensitivity, entropy, UACI, NPCR etc. Finally, Chai et al. [46] used color image cryptosystem based on improved genetic algorithm and matrix semi-tensor product (STP).

According to the papers surveyed, none of them have used an adaptive mechanism to improve the behaviour of encryption system and have used constant parameters. Most algorithms did use genetic algorithms but those were for pseudorandom number sequence generation.

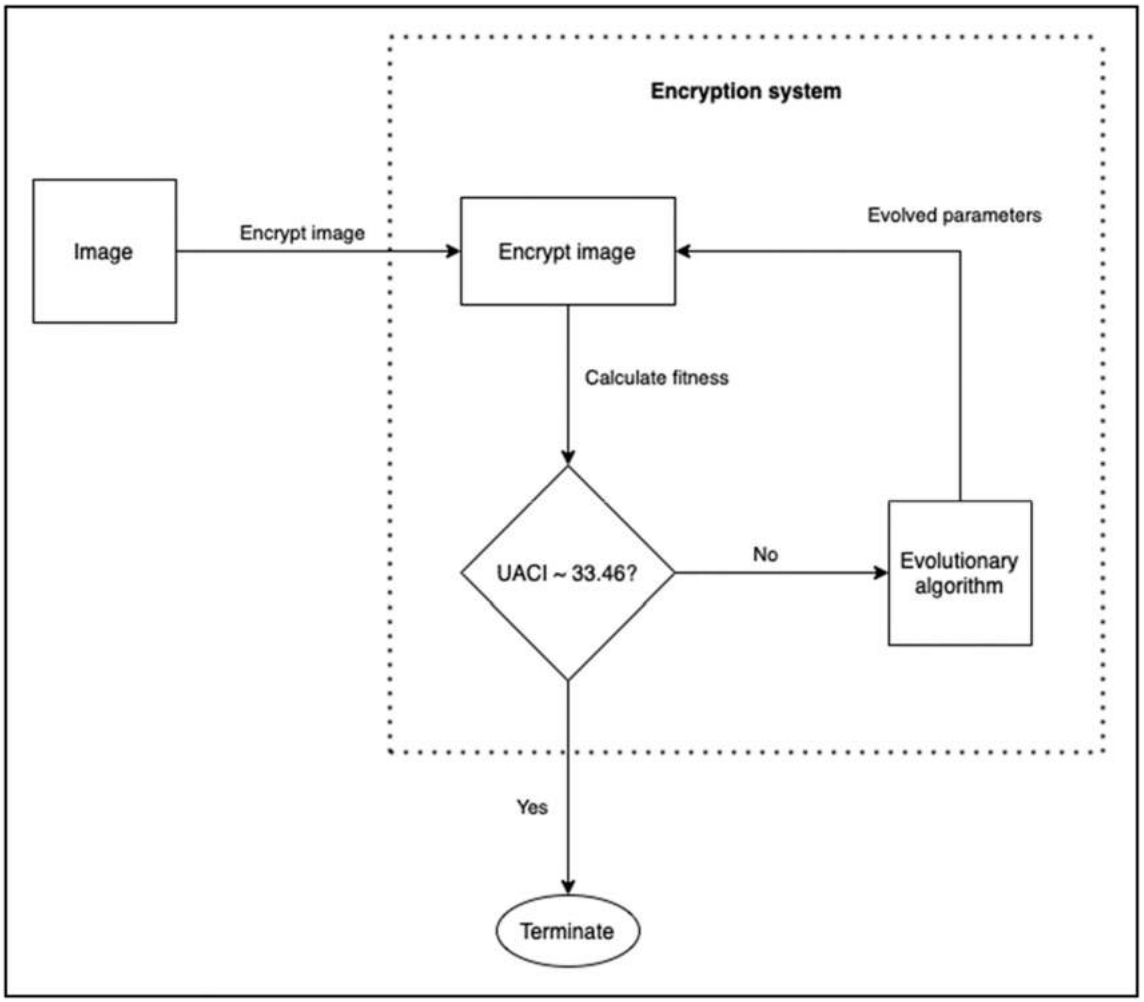


Fig. 8. Adaptive image encryption system.

Nearly all surveyed papers used chaos-based sequence generation in one way or another which confirmed the fact that chaos-based image encryption is the right way to move forward. Another key point discovered was the difficulties faced to optimize the Unified Average Changing Intensity (UACI) value in all the papers. Additionally, the lack of research in the adaptive encryption field could be bridged by re-engineering the approach as described in the sections further.

4. Proposed methodology and algorithm

4.1. Sample image description

Table 1 shows the sample image sets used during experimentation. Ten images are used to test the proposed algorithm. All the images used are of 8-bit grayscale. For the images, an equal number of 5 images each for asymmetrical and symmetrical images have been used. The variation in shapes and sizes helps in validating the viability of the proposed algorithm on different types of images. The image sizes vary from 256 $\times$ 256 to 1990 $\times$ 1342 thus, covering smaller as well as larger-sized images.

4.2. Adaptive mechanism of proposed algorithm

When a system responds to changes in the environment, it can be called an adaptive system. The method of adaptation varies from application to application, but its main objective is to achieve its defined goal and return to normalcy. A system that monitors its performance and adjusts its parameters to improve itself can also be classified as an adaptive system. In this paper, two methods of adaptation have been explored to improve image encryption. The first method is a genetic algorithm approach which updates its parameters using the hill climb algorithm. The second method used is Simulated Annealing which is an improved version of the hill climb algorithm. The goal of each of these adaptive mechanisms is to improve the UACI evaluation parameter to as close as 33.46 [21,22,24]. Fig. 8 depicts the working of the adaptive image encryption system. The image is fed into the encryption system which initially encrypts the image using a set of encryption parameters. The encrypted image is checked for its fitness using the UACI evaluation parameter. If the fitness is in the acceptable range, the algorithm is terminated. For cases where the fitness is not in the acceptable range, the parameters are fed into an evolutionary mechanism that makes manipulations in those parameters. The evolved parameters are then used to re-encrypt the image. This process continues until near-ideal fitness is obtained.

The purpose of using UACI as a metric of fitness is due to the difficulties obtained while encrypting an image and obtaining an ideal UACI value of 33.46. Other metrics such as entropy and NPCR are not as sensitive to encryption as UACI and almost all UACI values near 33.46 give us all other metrics in the near-ideal range. For experimentation purposes, an acceptable range of 33.46 $\pm$ 0.01 has been used.

4.3. Proposed PixAdapt algorithm

To encrypt an image, two adaptive approaches have been explored

and implemented. The first approach uses a genetic algorithm based on Hill Climb to improve the solution obtained after encrypting an image using selection and mutation based on a fitness criterion which is UACI. Most other image encryption parameters such as entropy, NPCR and correlation do not show much variation irrespective of the method of encryption. The second approach uses Simulated Annealing to evolve the solutions obtained from the cipher image.

4.3.1. Algorithm for generating pseudorandom sequences

Algorithms 1 and 2 contain the different methods involved in the generation of pseudorandom sequences, using chaotic methods and linear feedback shift register. PixAdapt algorithm uses the following chaotic methods namely - Logistic map, Tent map, Henon map and Rossler map.

Algorithm 1. Pseudorandom chaotic sequences.

```
function LogisticMap(r, seed_value, height, width)                ▷ Logistic map sequence generation
    logistic_array ← []
    n ← height × width
    for i ← 1, n do
        x_{n+1} ← r × (1 − x_n)
        logistic_array[i] ← x_{n+1}
    end for
    return logistic_array
end function

function TentMap(r, seed_value, height, width)                    ▷ Tent map sequence generation
    tent_array ← []
    n ← height × width
    x_n ← seed_value
    for i ← 1, n do
        if x_n ≥ 1/2 then
            x_{n+1} = r × (1 − x_n)
        end if
        if x_n < 1/2 then
            x_{n+1} = r × x_n
        end if
        tent_array[i] ← x_{n+1}
    end for
    return tent_array
end function

function HenonMap(a, b, x_seed, y_seed, height, width)            ▷ Henon map sequence generation
    henon_array_x ← []
    henon_array_y ← []
    n ← height × width
    x_n ← x_seed
    y_n ← y_seed
    for i ← 1, n do
        x_{n+1} = 1 − a × x_n^2 + y_n
        y_{n+1} = b × x_n
        henon_array_x[i] ← x_{n+1}
        henon_array_y[i] ← y_{n+1}
    end for
    return henon_array_x, henon_array_y
end function

function RosslerMap(a, b, c, x, y, z, height, width)              ▷ Rossler map sequence generation
    n ← height × width
    rossler_array_x ← []
    rossler_array_y ← []
    rossler_array_z ← []
    dx/dt = −y − z
    dy/dt = x + ay
    dz/dt = b + z × (x − c)
    x = ∫ (dx/dt) dx                                              ▷ Integrate using RK45 and return over span n
    y = ∫ (dy/dt) dx                                              ▷ Integrate using RK45 and return over span n
    z = ∫ (dz/dt) dx                                              ▷ Integrate using RK45 and return over span n
    return x, y, z
end function
```

Algorithm 2. Pseudorandom sequences - LFSR.

```
function LinearFeedbackShiftRegister(seed_value, height, width)
    lfsr_array ← []
    n ← height × width
    for i ← 1, n do
        bit ← seed[0] ⊕ seed[4] ⊕ seed[5] ⊕ seed[6]                ▷ 8 bit values
        delete seed[0]
        leftshift remaining bits
        seed[7] ← bit
        lfsr_array[i] ← seed
    end for
    return lfsr_array
end function
```

4.3.2. Algorithm for chaotic sequence generation

PixAdapt algorithm uses chaotic values which are generated through the above-mentioned pseudorandom sequences. A switching mechanism has been added into the system as represented in Fig. 9. This switching mechanism gives control to the evolutionary algorithm to activate or deactivate the participation of the sequence in generation of the final key at will. For each of the sequences, a single ON or OFF value is also inserted along with pseudorandom sequence (K_1, K_2, K_3, K_4 and K_5). A key sequence is generated by XORing all sequences which are active finally producing the secret key (K).

Algorithm 3 and Fig. 9 depicts the process of chaotic sequence generation used in PixAdapt algorithm.

Algorithm 3. Chaotic sequence generation.

```
Algorithm 3 Chaotic Sequence Generation
function GenerateChaos(Original_Image, log_params, tent_params, henon_params, rossler_params, lfsr_params)
    height ← height of the original_image
    width ← height of the original_image
    log_r, log_seed, log_on ← log_params
    tent_r, tent_seed, tent_on ← tent_params
    henon_a, henon_b, henon_seed_x, henon_seed_y, henon_on ← henon_params
    rossler_a, rossler_b, rossler_c, rossler_seed_x, rossler_seed_y, rossler_seed_z, rossler_on ←
rossler_params
    lfsr_seed ← lfsr_params

    if log_on == True then
        K_1 = LogisticMap(log_r, log_seed, height, width)
    end if

    if tent_on == True then
        K_2 = TentMap(tent_r, tent_seed, height, width)
    end if

    if henon_on == True then
        K_3 = HenonMap(henon_a, henon_b, henon_seed_x, henon_seed_y, height, width)
    end if

    if rossler_on == True then
        K_4 = RosslerMap(rossler_a, rossler_b, rossler_c, rossler_seed_x, rossler_seed_y,
rossler_seed_z, height, width)
    end if

    if lfsr_on == True then
        K_5 = LinearFeedbackShiftRegister(log_r, log_seed, height, width)
    end if

    K = K_1 ⊕ K_2 ⊕ K_3 ⊕ K_4 ⊕ K_5
    return K
end function
```

4.3.3. Algorithm for confusion and diffusion process

A new confusion-diffusion process has been implemented in PixAdapt algorithm. The process by the binary decomposition starts by taking the original image array and the chaotic sequence array generated through the pseudorandom sequences. Both the arrays are then converted into their binary form, which essentially helps in breaking the binary array into eight channels. The eight channels generated are divided into a set of four channels each. This process is called binary decomposition. The set of four channels obtained thus far, are then again converted to binary sequence, generating A_1, A_2, a_1 and a_2. The sequences A_1 and A_2, are right shifted each by the sum of the other sequence, giving A_{11} and A_{21}.

All the four binary sequences A_{11}, A_{21}, a_1 and a_2 are iterated over N times, where N represents the size of each binary sequence. This helps in calculating two different cipher image arrays C_1 and C_2, each 8 channels similar to the original image and chaotic sequence array. The first value of C_1 array is calculated by XORing the first value of A_{11} the last value of A_{11} and the first value of a_1. The first array value of C_2 is calculated in a similar manner. For the rest of the values of the arrays XOR of i^{th} value of A11, $(i-1)^{th}$ value of A_{11} and i^{th} values of a_1 is done. The rest of the values of array C_2 are calculated in a similar fashion, by XORing the i^{th} and $(i-1)^{th}$ value of A_{21} and the ith value of a_2. The arrays C_1 and C_2 are reshaped into 4 planes based on seq_1 and seq_2. C_1 and C_2 are then merged into a single array C, which generates the final cipher image. This whole process is repeated over n times, where n is a random number.

Fig. 10, and Algorithm 4 depict the whole process confusion and diffusion process.

Algorithm 4. Confusion and diffusion.

Algorithm 4 Confusion and Diffusion

```
function BinaryDecomposition(Binary_Image)
    height ← Binary_Imageheight
    width ← Binary_Imagewidth
    temp_array ← zeros(height, width, 8) ▷ Creating a temporary image array having 8 planes
    counter_1 ← height
    counter_2 ← width
    counter_3 ← 7
    for i ← 0, counter_1 do
        for j ← 0, counter_2 do
            var ← Binary_Image[i, j]
            for k ← 0, counter_3 do
                temp_array[i, j, k] ← var[k]
            end for
        end for
    end for
    return temp_array
end function

function ConfusionDiffusion(Original_Image, Chaotic_Sequence, seq_1, seq_2)
    loop ← random number
    for i ← 1, loop do
        Original_Image_binary ← ConvertToBinary(Original_Image)
        Chaotic_Sequence_binary ← ConvertToBinary(Chaotic_Sequence)
        Original_Image_decomposed ←BinaryDecomposition(Original_Image_binary)
        Chaotic_Sequence_decomposed ←BinaryDecomposition(Chaotic_Sequence_binary)
        counter_1 ← 0
        for p ← 0, seq_1 do
            Original_Image_seq_1[counter_1] ← Original_Image_decomposed[p]
            Chaotic_Sequence_seq_1[counter_1] ← Chaotic_Sequence_decomposed[p]
            counter_1 ← counter_1 + 1
        end for
        counter_2 ← 0
        for q ← 0, seq_2 do
            Original_Image_seq_2[counter_2] ← Original_Image_decomposed[q]
            Chaotic_Sequence_seq_2[counter_2] ← Chaotic_Sequence_decomposed[q]
            counter ← counter_2 + 1
        end for
        A_1 ← BinarySequence(Original_Image_seq_1)                ▷ Converting to 1D
        A_2 ← BinarySequence(Original_Image_seq_2)                ▷ Converting to 1D
        Sum_A_1 ← Sum of values of A_1
        Sum_A_2 ← Sum of values of A_2
        A_11 ← Right shift by Sum_A_2   ▷ Moving the bits in the right direction by Sum_A_2
        A_21 ← Right shift by Sum_A_1   ▷ Moving the bits in the right direction by Sum_A_1
        a_1 ← BinarySequence(Chaotic_Sequence_seq_1)              ▷ Converting to 1D
        a_2 ← BinarySequence(Chaotic_Sequence_seq_2)              ▷ Converting to 1D
        length ← 4 × height × width                ▷ height and width of Original_Image
        C_1 ← []
        C_2 ← []
        for i ← 0, length do
            if i==0 then
                C_1[i] ← A_11[i] ⊕ A_11[length − 1] ⊕ a_1[i]
                C_2[i] ← A_21[i] ⊕ A_21[length − 1] ⊕ a_2[i]
            end if
            if i!=0 then
                C_1[i] ← A_11[i] ⊕ A_11[i − 1] ⊕ a_1[i]
                C_2[i] ← A_21[i] ⊕ A_21[i − 1] ⊕ a_2[i]
            end if
        end for
        C_1 ← reshape back to 4 planes according to seq_1
        C_2 ← reshape back to 4 planes according to seq_2
        C ← Merging C_1 and C_2                    ▷ converting to 8 plane binary image
        cipher ← uint8(C)                          ▷ Single plane 8 bit image
    end for
    return cipher
end function
```

4.3.4. Algorithm for genetic hill climb

Fig. 11 shows the methodology followed during encrypting an image using genetic hill climb. The first step is to generate genes for each of the pseudo-random number generators. Five pseudo-random number generators have been used which have been described in the previous sections. For each of these generator functions, the parameters have been generated using the following ranges.

Logistic map
r: 3.6 – 4.0
Seed value: 0.01 – 1

Linear Feedback Shift Register
Seed pixel value: 0 – 255 (converted to binary values)

Rossler map
c: 9, 10, 13 or 18

Tent map
Seed value: 0.01 – 1
r: 1 – 2

Henon map
x seed value: 0.1 – 1
y seed value: 0.1 – 1
a: 1- 1.4

The above-mentioned ranges for respective parameters are used to generate a population size of 100 for each of the genes. A single pair of values from these genes is selected and a genotype is created. The genotype contains a single value from each gene which is used to encrypt the image. The cipher image is obtained through the confusion-diffusion process. This encrypted image is then used to calculate the fitness of the respective parameters. The algorithm continues till the acceptable value of UACI is achieved.

For cases where the UACI value is either greater than or less than the acceptable range, a single gene from the genotype is selected and mutated. This mutation occurs by replacing this gene in the genotype from the initial population. The replaced gene is selected from the right or left neighbour of the current gene stored in the population gene sequence. The new genotype is then used to re-encrypt the image until the UACI value is found to be in the acceptable range.

Algorithm 5 and Fig. 11 depict the process of genetic hill climb algorithm.

Algorithm 5. Genetic hill climb.

```
function Fitness(encryptedImage, originalImage)
    M ← height of encryptedImage
    N ← width of encryptedImage
    UACI = Σ_{i=1}^{M} Σ_{j=1}^{N} [|C_1(i,j) − C_2(i,j)| / 255] × 100% / (M × N)
    return |33.46 − UACI|
end function

function DiscreteGene(start, stop)                          ▷ Create discrete gene
    return integer genes in range(start, stop)
end function

function ContinuousGene(start, stop)                        ▷ Create continuous gene
    return float genes in range(start, stop)
end function

function Update(genotype)                                   ▷ Mutate Genotype
    return random ordered mutated gene
end function

function Main                          ▷ Genetic Hill climb for image encryption
    ImageGenes ← generate all discrete and continuous genes
    Generate genotype
    fitness = EncryptImage(originalImage, ImageGenes)
    while True do
        if fitness > 0.01 then
            Select and update genotype
        end if
        if fitness < 0.01 then
            select genotype and break loop
        end if
    end while
end function
```

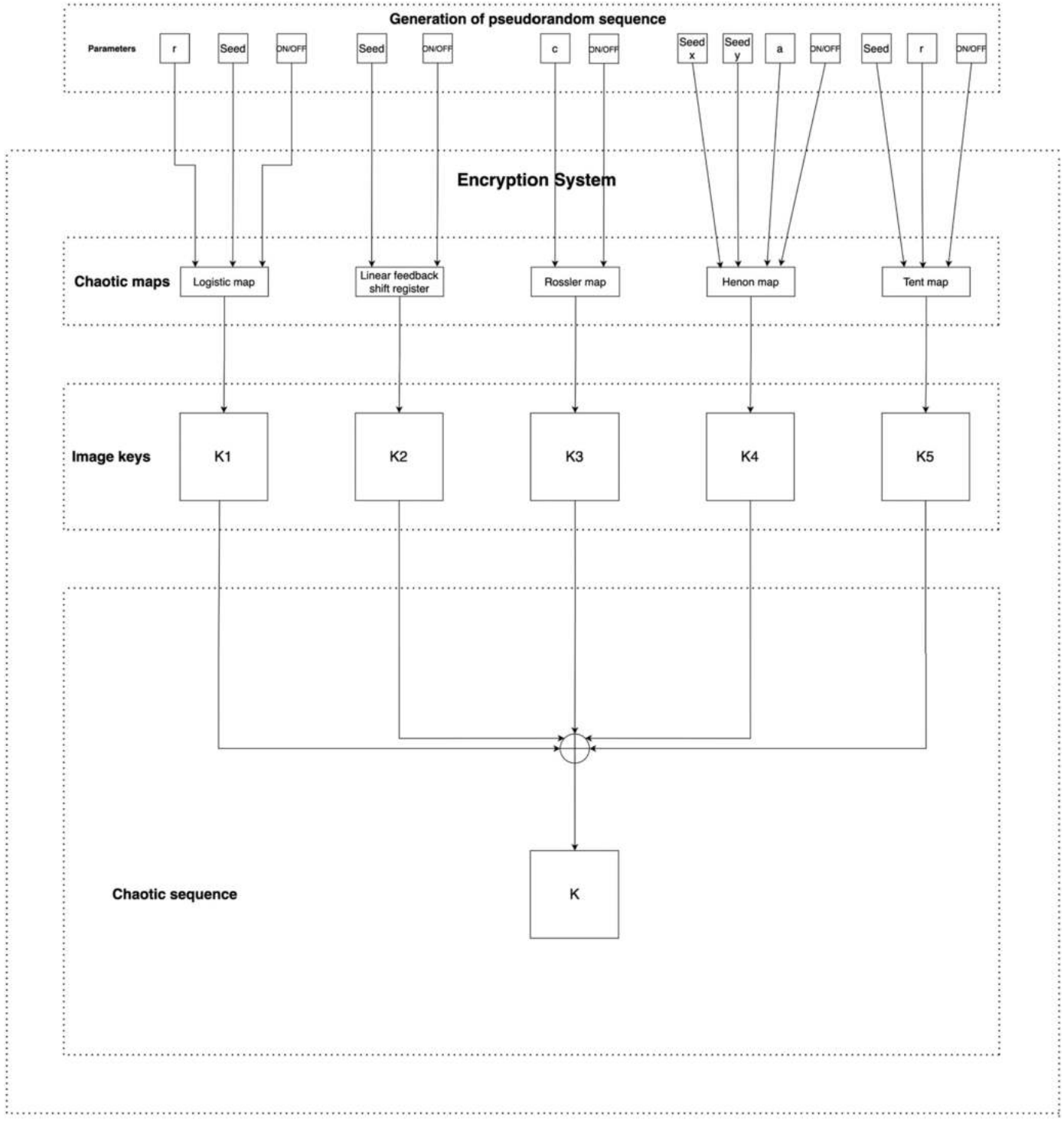


Fig. 9. Pseudorandom sequence generation.

4.3.5. Algorithm for simulated annealing

The second approach used to encrypt an image uses Simulated Annealing. Fig. 12 shows the adaptive mechanism used to encrypt an image using Simulated Annealing. In this process, an initial temperature is set to an arbitrary value. After which, a set of parameters are defined which are used to encrypt the image. The initial parameters for all images are the same and a range between which the parameters are most likely to produce the most chaotic values are also stored. They are as

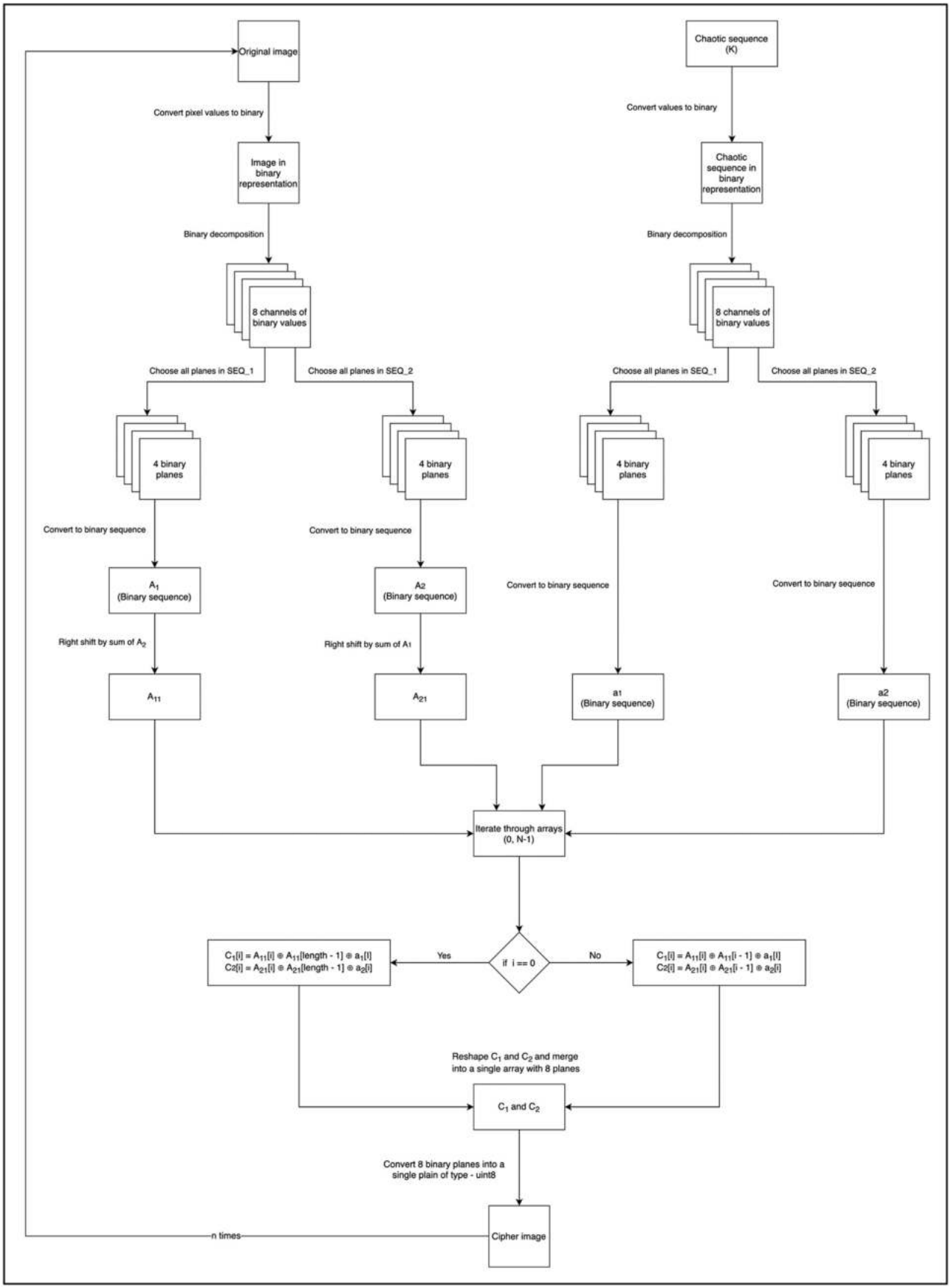


Fig. 10. Confusion-Diffusion process.

follows:

Logistic map
r: 3.6 (range: 3.6 – 4.0)
Seed value: 0.01 (range: 0.01 – 1)

Linear Feedback Shift Register
Seed pixel value: 1 (range: 1– 255)

Rossler map
c: 9 (range: 9 – 18)

Tent map
Seed value: 0.01 range: (0.01 – 1)
r: 1 (range: 1 – 2)

Henon map
x seed value: range: (0.1 – 1)
y seed value: range: (0.1 – 1)
a: (range: 1- 1.4)

Each of the above-mentioned parameters is evolved using the simulated annealing principle. Each of parameters obtained are fed into the respective pseudo-random number generation systems. The cipher image is obtained through the confusion-diffusion process. The parameters obtained are also restricted to be used in their reference ranges only.

The fitness score of the parameters used to encrypt an image are calculated by computing the UACI of the ciphered image. After which, the parameters are evolved using by either adding or subtracting a random value obtained from a Gaussian distribution. The newly generated values are then evaluated for their fitness and the difference between the old parameter fitness score and the new parameter fitness score is generated. If the new set of parameters performs better than the previous set of parameters, the new set is accepted. If the new set does not perform better than the old set, the new set may be accepted such that the metropolis factor of the temperature and fitness difference is greater than a random number generated at that instant. Accepting underperforming parameters from time to time ensures reduction in the possibility of stagnation in the local optima. After the solution is accepted or rejected, there is a reduction in the temperature. The reduction value used during experimentation was equal to 0.1. This process occurs until the temperature reaches 0 or an acceptable range of UACI is obtained (33.46 ± 0.01).

Algorithm 6 and Fig. 12 depict the process of simulated annealing.

Algorithm 6. Simulated annealing.

```
function FITNESS(encryptedImage, originalImage)
    M ← height of encryptedImage
    N ← width of encryptedImage
    UACI = Σ_{i=1}^{M} Σ_{j=1}^{N} |C1(i,j) − C2(i,j)| / 255 × 100% / (M×N)
    return |33.46 − UACI|
end function

function UPDATEPARAMS(oldParam, σ, μ)
    newParam = oldParam + Gaussian(σ, μ)
    return newParam
end function

function MAIN(initialTemperature, alpha)
    params ← generate all parameters
    Temperature ← initialTemperature
    fitness1 = EncryptImage(originalImage, params)
    while Temperature > 0.1 or fitness > 0.01 do
        newParams ← updateParams(σ, μ)
        fitness2 = EncryptImage(originalImage, newParams)
        if fitness1 < fitness2 then
            params = newParams
        end if
        if fitness1 > fitness2 then
            metropolis = e^{−(fitness1 − fitness2)/Temperature}
            if metropolis > random_uniform_number then
                params = newParams
            end if
        end if
        Temperature = Temperature - alpha
    end while
end function
```

5. Results and analysis

Traditional Image encryption processes use a static set of values to encrypt images and do not consider the features of an individual image. These drawbacks were overcome by using an adaptive approach to encrypt images. The proposed algorithms were tested on the images described in the previous sections. For the genetic hill climb algorithm, the algorithm was tested using population size of 100. For the Simulated Annealing technique, the initial temperature was set to 20 and its reduction step size was set at 10.

For the non-adaptive approach, the pseudo random sequence generation parameters were set to their most chaotic or random values. They are as follows: logistic map seed = 0.01, logistic map r = 3.99, linear feedback register seed = 10001001, tent map seed = 0.01, tent map r = 1.99, Rossler map c = 9, Henon map x seed = 0.01, Henon map y seed = 0.01 and Henon map a = 1.4.

For this method, results for fitness improvement, parameter evolution, statistical features, and quality of encryption and key space are discussed.

5.1. Fitness improvement

Analyzing the fitness of each of the parameters is an important way of determining the adaptiveness of the algorithm. Only if the algorithm evolves its parameters correctly, the fitness of the respective image will reach the acceptable range of its UACI value.

The first method involves using static parameters to encrypt images. The results for image_3 and image_8 are in the acceptable range. On the other hand, Image_4, Image_5, Image_9, Image_10 are closest to reach near the acceptable range but still not within the threshold value of acceptance. It is observed from Table 2, when the parameters are adapted using the genetic algorithm, they perform much better than the static parameters. While for Image_6 the UACI value is observed to be very poor. The values obtained by genetic hill climb algorithm for population size 100 and simulated annealing are in ideal range value of acceptance. Because of the adaptiveness of the algorithm it can be observed that the UACI value for Image_6 also within the acceptable range.

Figs. 13 and 14 show the fitness generated by the images (Image_1, Image_2, Image_3, Image_6, Image_9, Image_10) with respect to the number of epochs for genetic hill climb algorithm and simulated annealing respectively. It is observed for most of the images, the fitness value either undershoots or overshoots. It gradually stabilizes to its most optimal value of UACI which is the primary requirement of a fitness function. For example, the performance of fitness function in for Image_9 is linear (see Fig. 13). It means that the optimal value of UACI for Image_9 gradually increases till the optimal value is achieved. For the same image_9, in case of simulated annealing, it directly hits the optimal value of UACI.

5.2. Parameter evolution

An overall of eight parameters (stated in above section) have been used to generate pseudo-random sequences and five other parameters have been used to switch ON and OFF the respective sequences.

5.2.1. Logistic map

Fig. 15 depicts the adaptive behaviour of the r and seed value parameters while using hill climb and simulated annealing. For the genetic hill climb algorithm, it can be observed that for, Image_6 and Image_7 map was OFF at all times. The UACI values observed were initially very low in both the images. For Image_6 the initial value was between 12.5 and 15.0, it increases to around 17.5. Lastly, it increases to the optimal value of UACI. For Image_7, the initial value was between 12.5 and 15.0 and then it increases to the optimal value of UACI. For the genetic simulated annealing algorithm, the map remained OFF all the times for

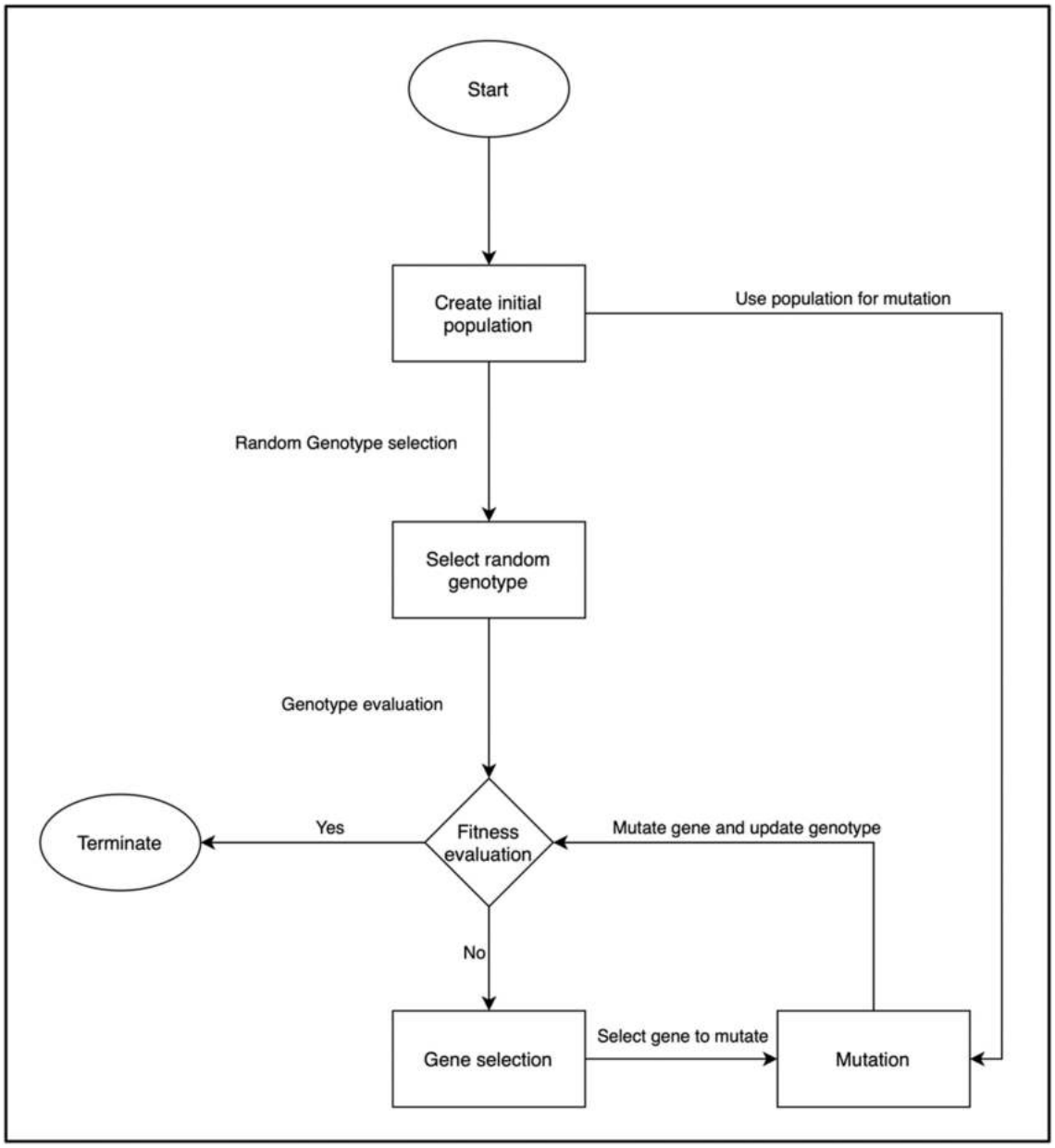


Fig. 11. Image encryption using Genetic Hill Climb.

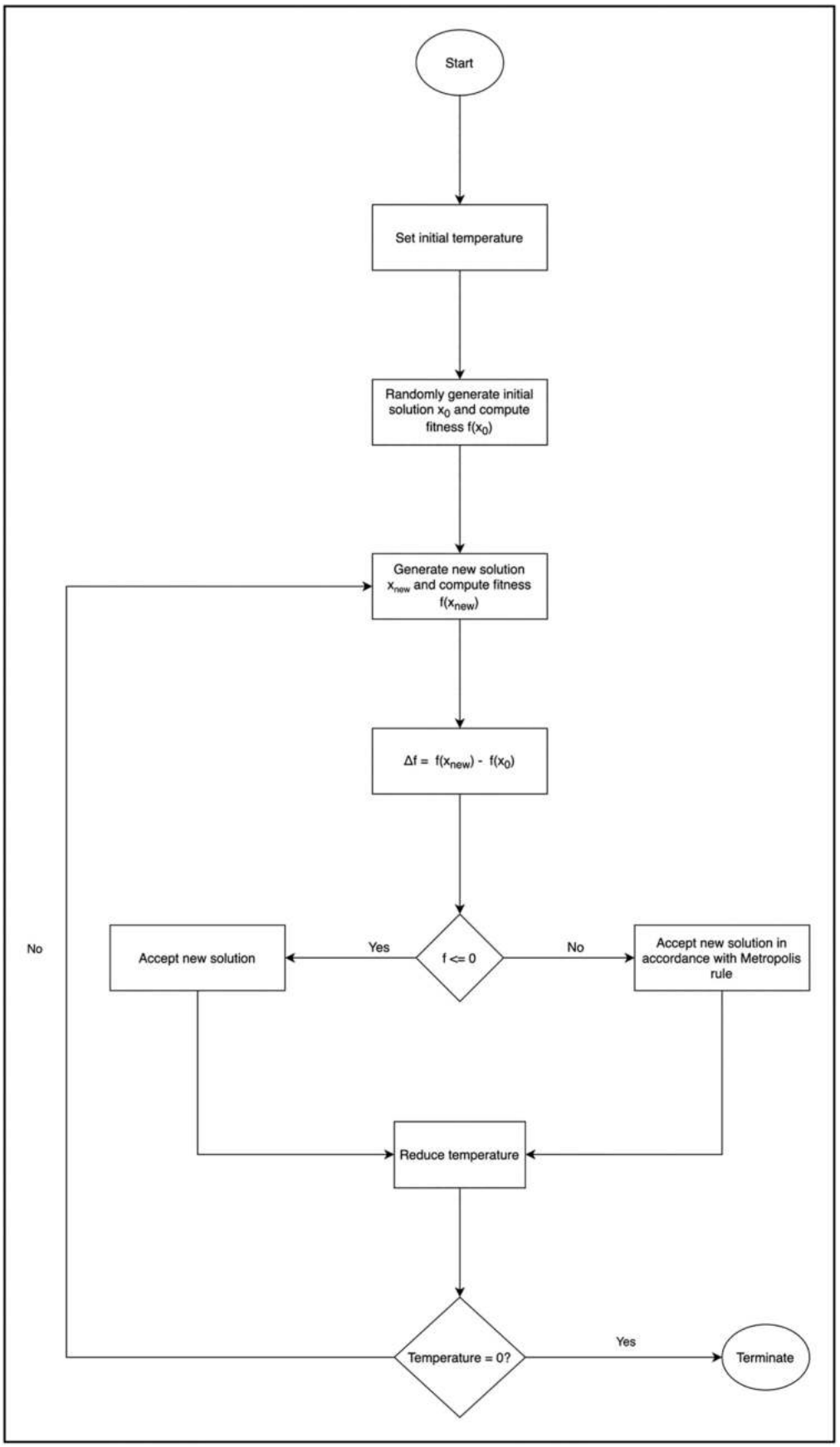


Fig. 12. Image encryption using Simulated Annealing.

Table 2
Fitness analysis (UACI).

Image name	UACI value (without an adaptive mechanism)	UACI value (with Genetic Hill Climb)	UACI value (with simulated annealing)
Image_1	33.382036	33.458419	33.456576
Image_2	33.411194	33.466784	33.460854
Image_3	33.463399	33.469307	33.467704
Image_4	33.436743	33.460255	33.462827
Image_5	33.444696	33.467931	33.465042
Image_6	14.230341	33.452366	33.450543
Image_7	33.528377	33.469921	33.461253
Image_8	33.466027	33.455051	33.451908
Image_9	33.480454	33.456628	33.463717
Image_10	33.456094	33.460107	33.469974

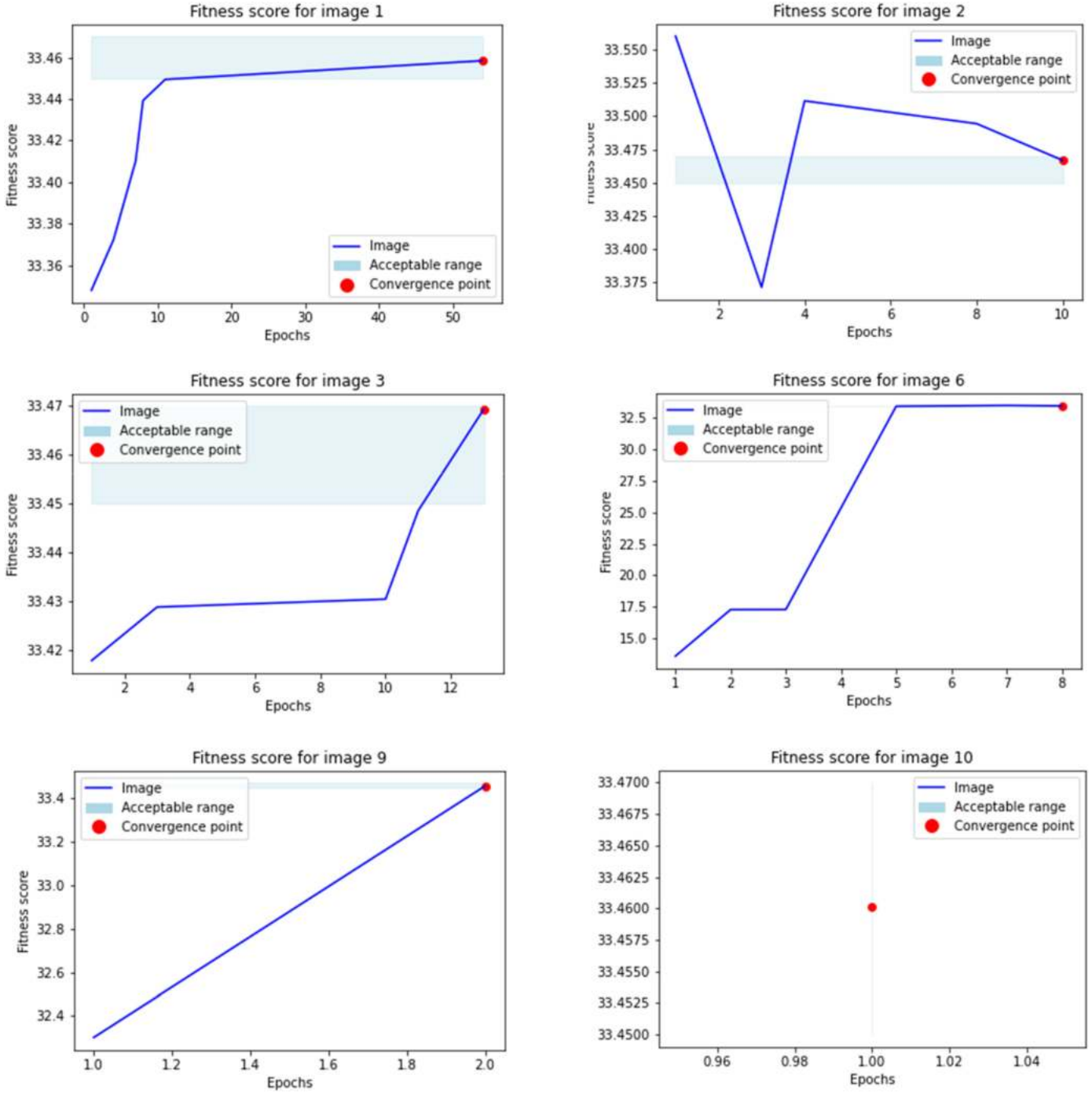


Fig. 13. Fitness improvement for genetic hill climb algorithm.

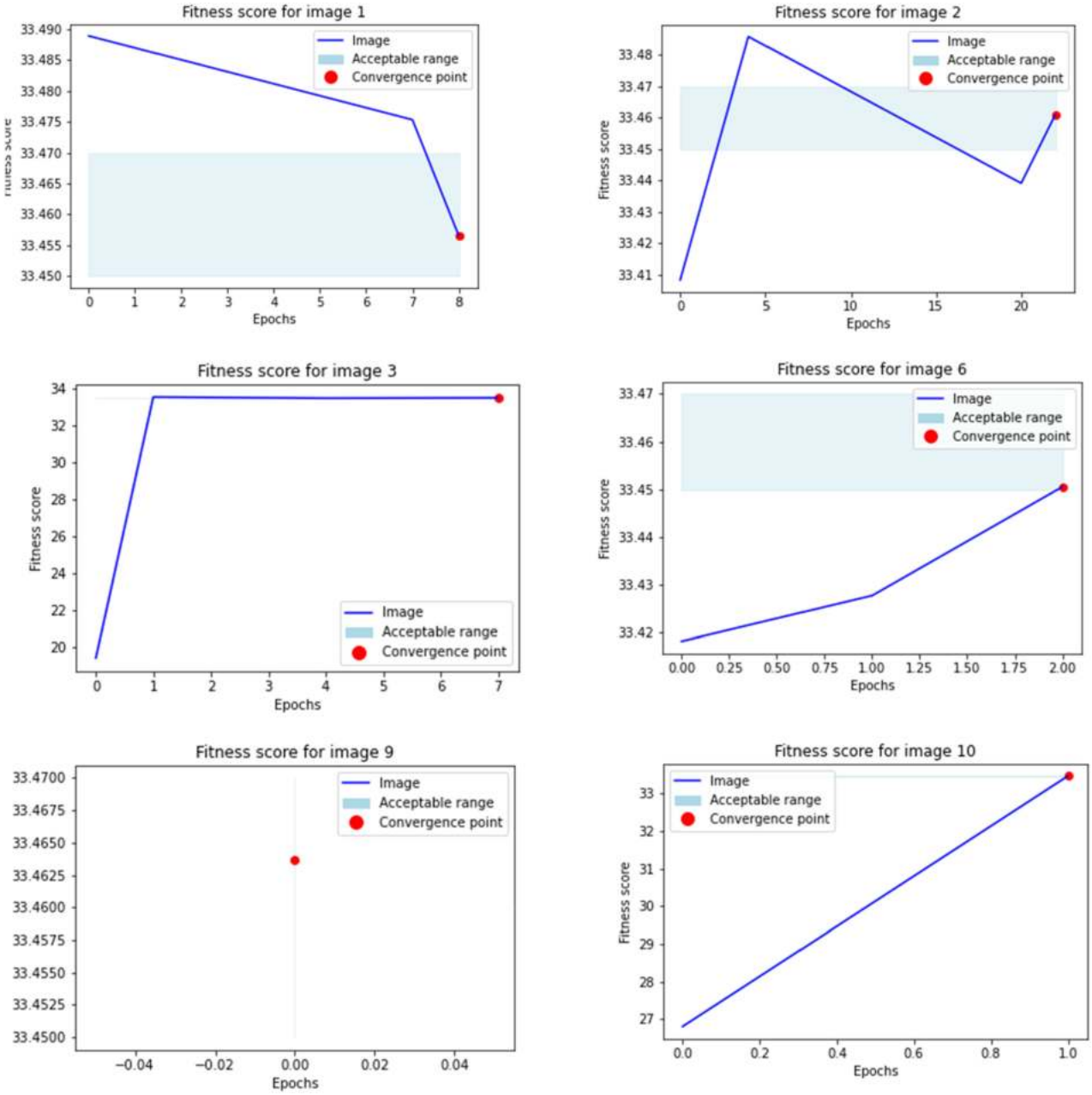


Fig. 14. Fitness improvement for simulated annealing.

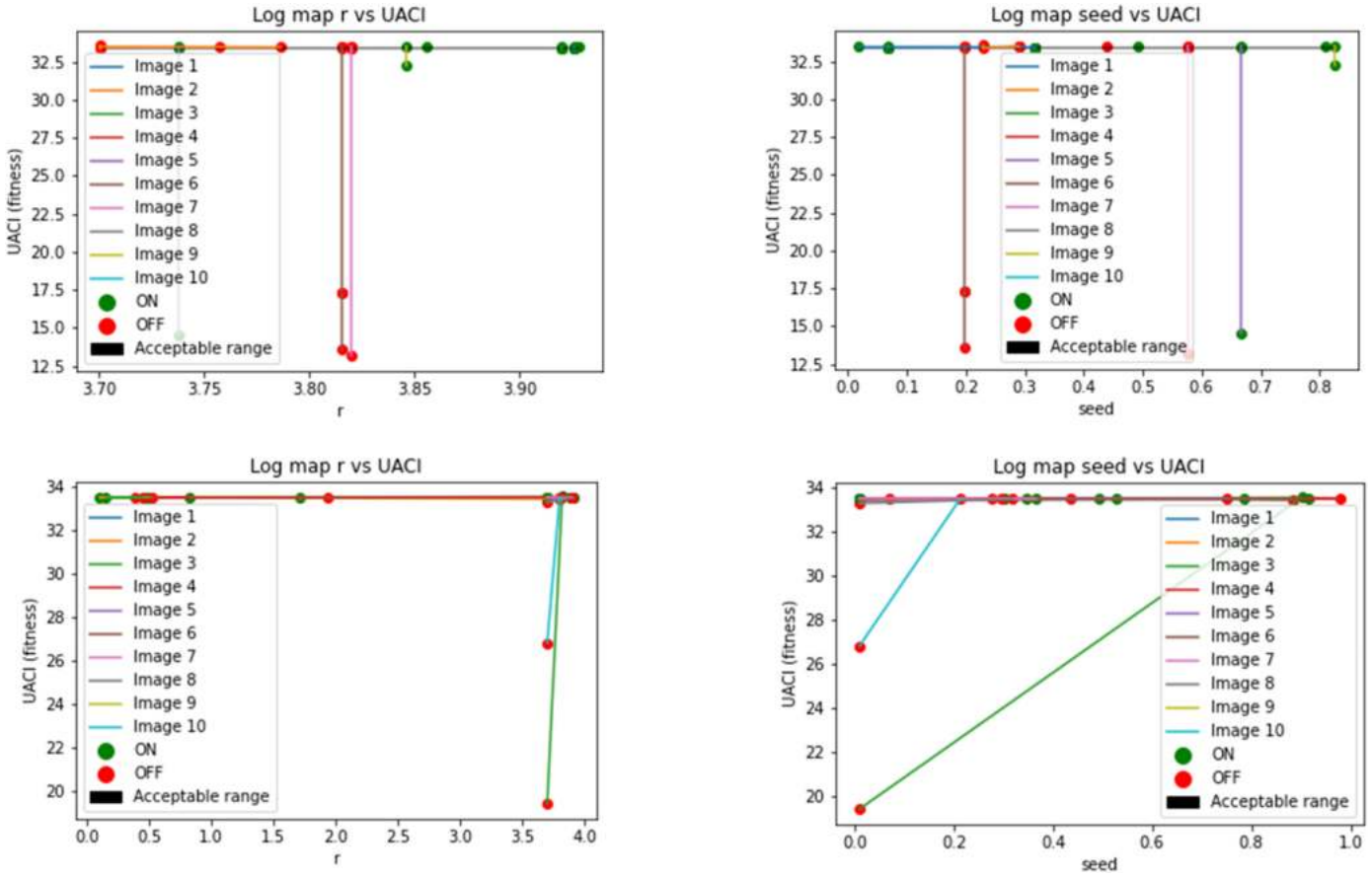


Fig. 15. Logistic map parameters r and seed value parameters.

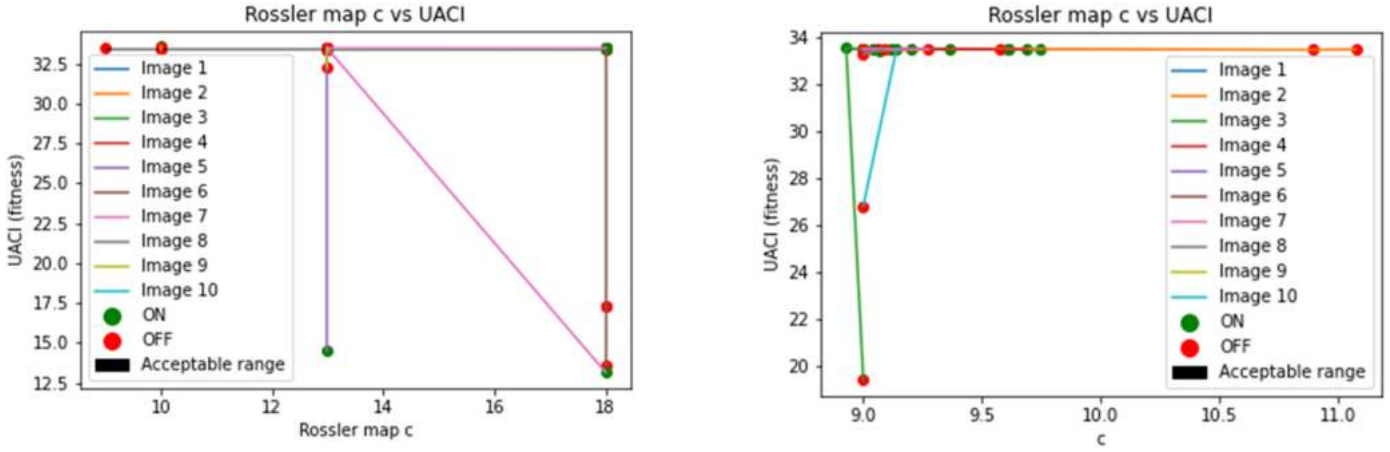


Fig. 16. Rossler map c parameter.

Image_3 and Image_10. The initial value of Image_3 is around 20 while initial value of Image_10 is observed between 26.0 and 28.0.

5.2.2. *Rossler map*

According to Fig. 16, in the genetic hill climb approach, c values around 13 did not work well and in fact, when the Rossler map was off, the UACI value was in the acceptable range. For most images, Rossler map did not contribute to increasing fitness. Using the simulated annealing approach, for Image_3 and Image_10 achieves optimal value of UACI when the map was ON.

5.2.3. *Tent map*

According to Fig. 17, for the simulated annealing approach, Image_10 draws an attention to the optimal value of UACI. Using simulated annealing approach, when the map was ON both the seed value and r value, UACI value observed was low and when the map was OFF, it was in the acceptable range.

5.2.4. *Henon map*

Henon map using genetic hill climb approach gives better results in case of 5 images (see Fig. 18). While for Image_5, it can be deduced that when the map was ON, the fitness value was low. Henon map using

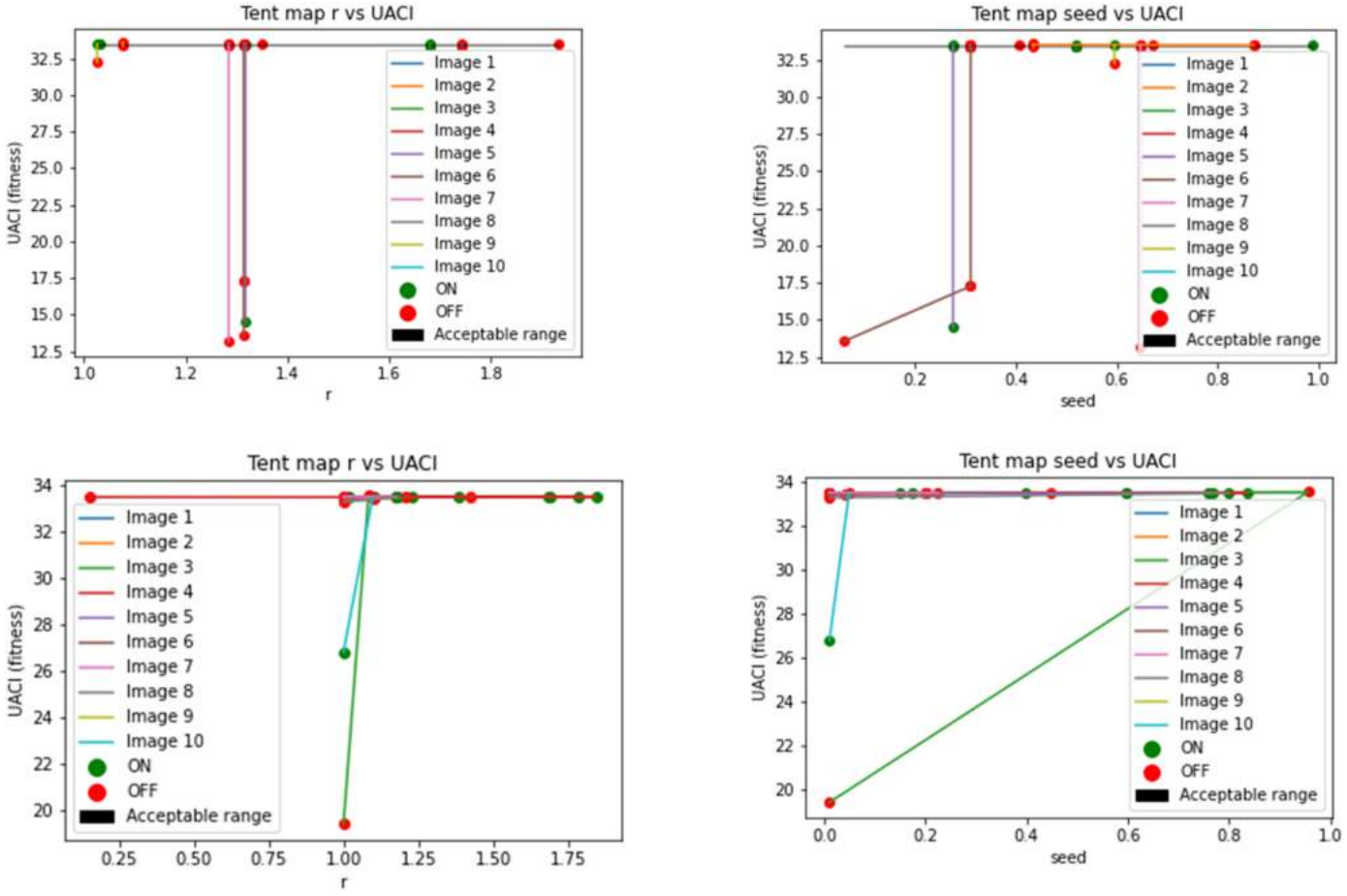


Fig. 17. Tent map parameters r and seed value parameters.

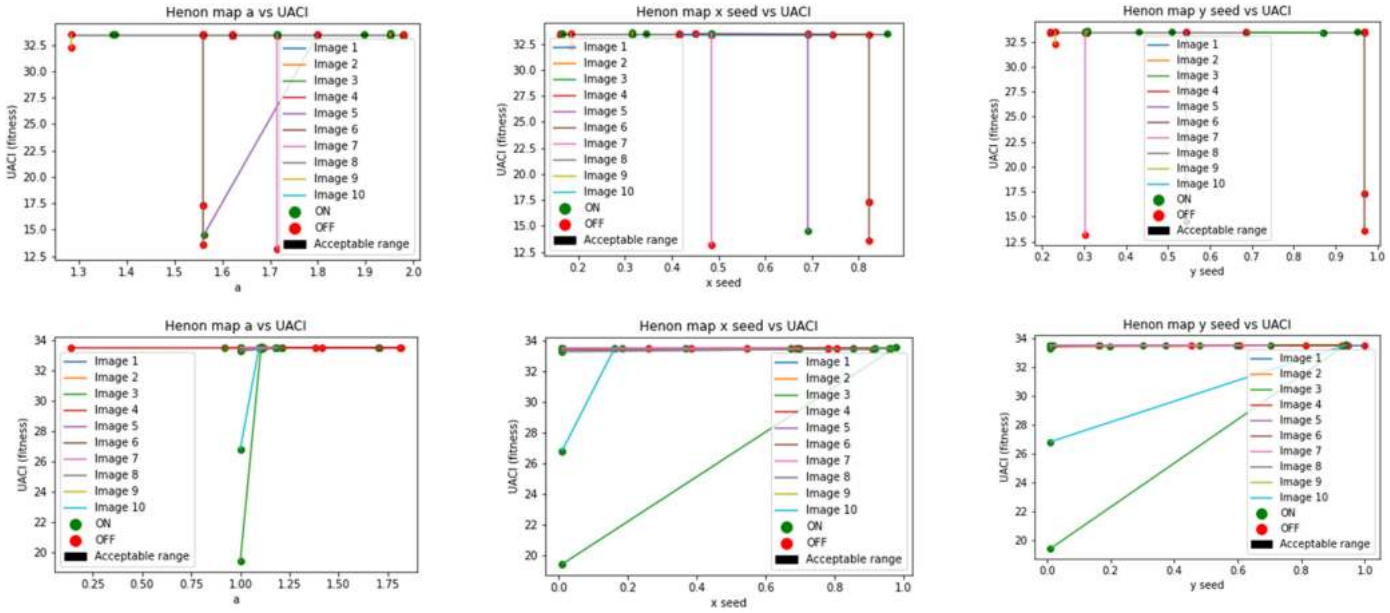


Fig. 18. Henon map parameters x seed, y seed and a value parameter.

Table 3
image entropy readings.

Image name	Entropy			
	Original Image	Encrypted image		
		Non-adaptive	Genetic algorithm	Simulated annealing
Image_1	7.360105	7.996818	7.996992	7.997248
Image_2	6.889566	7.997899	7.998016	7.997651
Image_3	4.961201	7.997850	7.997777	7.998098
Image_4	7.558386	7.999226	7.999277	7.999323
Image_5	4.931948	7.999658	7.999618	7.999557
Image_6	7.052831	7.999248	7.999314	7.999311
Image_7	6.559510	7.999307	7.999323	7.999279
Image_8	7.452617	7.999585	7.999521	7.999580
Image_9	7.535319	7.999932	7.999932	7.999931
Image_10	7.314666	7.999833	7.999818	7.999838

simulated annealing approach creates a positive impact when the map is ON.

According to results obtained from the above analysis, it can be concluded that a single pseudo-random number does not always give the correct pseudorandom sequence to encrypt the image with. Logistic map parameters had the most to contribute to but for the other images, the tent and Henon map sequences worked very well. For the most part, the switching mechanism provides another level of adaptation to the already evolving solutions to check for the presence and absence of a few pseudorandom sequences.

5.3. Statistical analysis

It is essential for a cryptosystem to be resistant against statistical attacks. Parameters such as (i) entropy, (ii) correlation, (iii) contrast, (iv) histogram analysis, and (v) chi-square test provide statistical insights into the behaviour of the plain and cipher image. The results obtained are as follows:

5.3.1. Entropy

The algorithm's robustness is a crucial factor. For any encryption algorithm to be successful, the measure of entropy should be as high as possible (ideally ~7.999 for 8-bit images). Image entropy is calculated using the following:

$$H(m) = \sum p_i log_2\left(\frac{1}{p_i}\right)$$

where p_i denotes the likelihood of the i^{th} pixel value.

Table 3 shows the entropy values obtained from the original image, non-adaptive (static parameter) image encryption, genetic hill climb image encryption and simulated Annealing image encryption. According to the results, it can be observed that the cipher image obtained from the three methods of encryption has a much higher value of entropy than the original image. The cipher images have entropy values in the ideal range thus, the encryption algorithm with static parameters and the adaptive image encryption system demonstrates high resistance to entropy attack.

5.3.2. Correlation

Correlation can be defined as a method of determining the similarity between the plain image and the encrypted image. Correlation is calculated in the vertical, horizontal, and diagonal directions. Plain images have a high degree of correlation in all directions while a successfully encrypted image should have near zero correlation. The correlation coefficient is defined as follows:

$$r_{xy} = \frac{\sum_{i=1}^{M}\left(x_i - \frac{1}{M}\sum_{j=1}^{M}x_j\right)\left(y_i - \frac{1}{M}\sum_{j=1}^{M}y_j\right)}{\sqrt{\sum_{i=1}^{M}\left(x_i - \frac{1}{M}\sum_{j=1}^{M}x_j\right)^2}\sqrt{\sum_{i=1}^{M}\left(y_i - \frac{1}{M}\sum_{j=1}^{M}y_j\right)^2}}$$

where x_i and y_i are pairs of horizontal, vertical, and diagonal i^{th} adjacent coordinates and M represents the overall number of adjacent pixel pairs under consideration. For our analysis, 10,000-pixel pairs adjacent to each other in the horizontal, diagonal and verticals directions are chosen at random, and their correlation coefficients have been calculated.

Fig. 19 depicts the distribution of the correlation coefficients before and after encryption. The correlation distribution of the original images is high and concentrated in a single region. After encryption, the distribution of the image is scattered and the correlation coefficients are highly dispersed thus, the image after encryption has a very low degree of correlation in comparison to the original image.

In Table 4, the individual correlation coefficient of the test images in the vertical, diagonal, and horizontal directions have been shown. The correlation coefficient of the original image is very close to unity thus, showing high correlation. After encryption, using the genetic hill climb as well as the simulated annealing in all three directions show very low correlation to zero correlation. Thus, after encrypting an image using the proposed algorithm, the cipher image pixels are uncorrelated to each other in the vertical, diagonal, and horizontal directions exhibiting high resistance against statistical attacks.

5.3.3. Contrast

Contrast of an image is the overall variation in intensity of pixels with respect to their neighbours. Images after encryption should exhibit a high degree of contrast in comparison to the original image. Contrast is calculated using the following mathematical expression:

$$C = \sum_{i,j=0}^{levels-1} P_{i,j} \times (i-j)^2$$

where P is grey-level co-occurrence matrix containing the intensity values which are calculated from the original image, and i and j are the row and column values for the respective GLCM matrix. The total number of levels is 256.

Table 5 shows the contrast values obtained before and after encryption in the vertical, horizontal, and diagonal directions. According to the results, before encrypting the image, the contrast of the images is significantly lesser than after encryption. For both encryption schemes, the contrast is amplified significantly thus, there is very low predictability of the plain image from the cipher image and high security is demonstrated.

5.3.4. Histogram analysis

For a cryptosystem to be resistant against statistical attacks, the histogram of cipher image must be uniformly distributed.

According to Fig. 20, the histogram prior to encryption depicts the features present in the image and has a regular shape. After encryption, the image has no similarity with respect to the original image and this argument is solidified by looking at the flattened histogram obtained after encryption. Thus, the proposed cryptosystem is resistant to histogram statistical attack.

5.3.5. Chi square test

Statistical features can be further analysed by using the chi square test. The mathematical formula is as follows:

$$x^2_{exp} = \sum_{i=0}^{N_v-1}\frac{(o_i - e_i)^2}{e_i}$$

In the above equation, o_i is the observed frequency occurrence (between 0 and 255), N_V is the intensity levels (256), e_i is the occurrence expected from the uniformly distributed histogram obtained from the

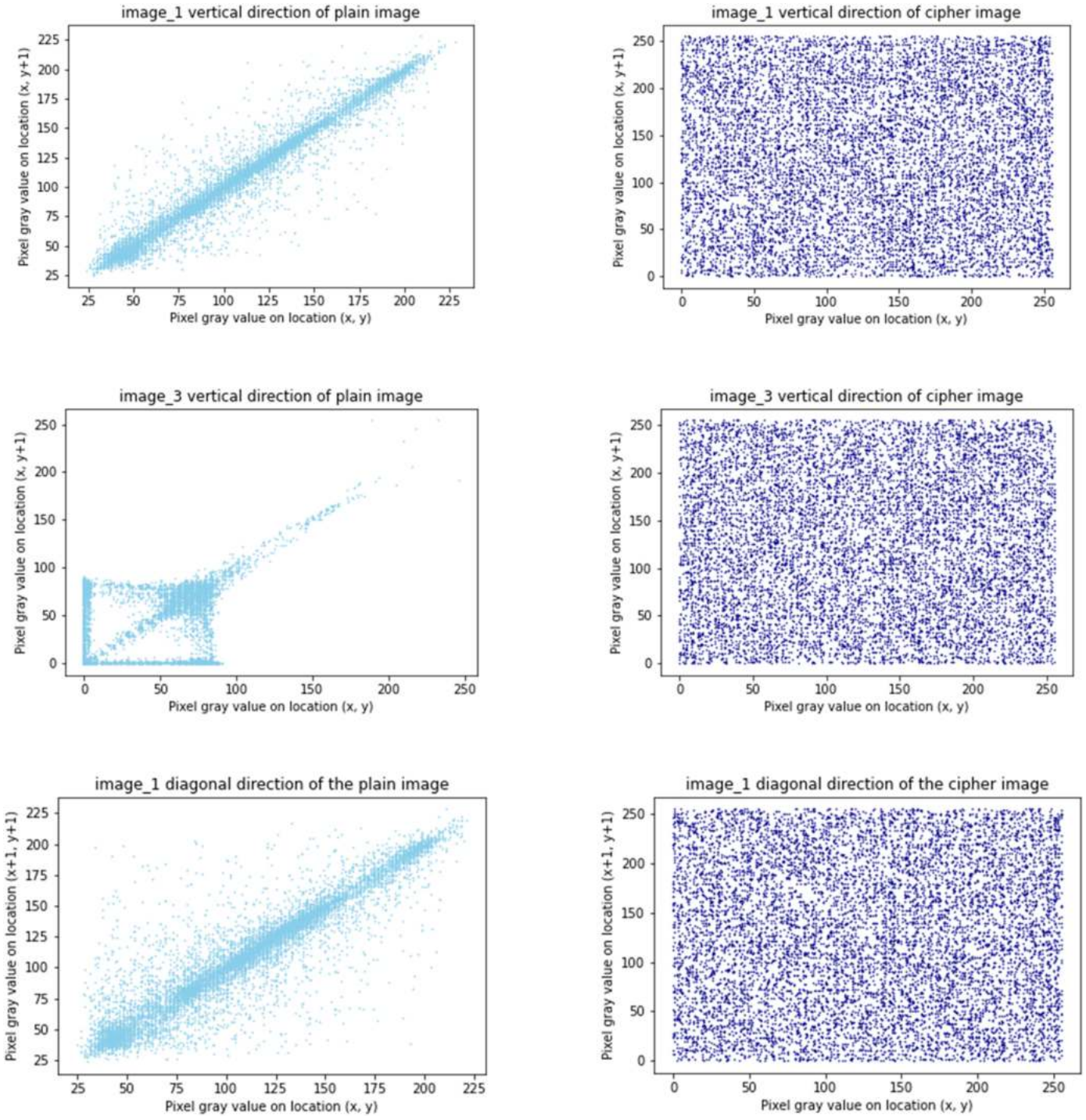
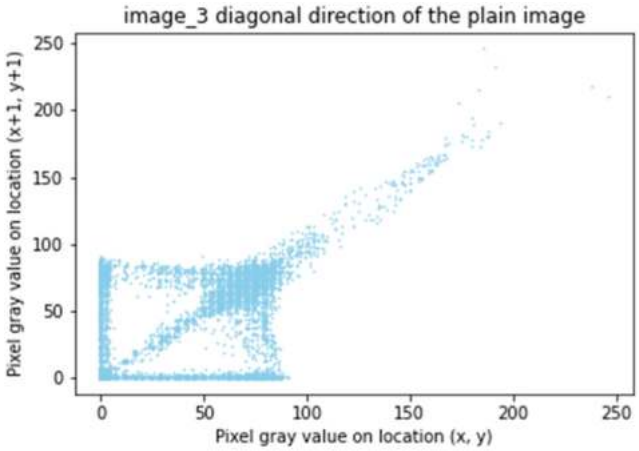
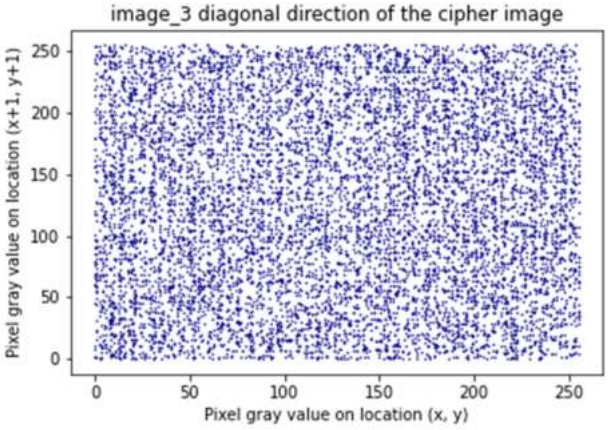
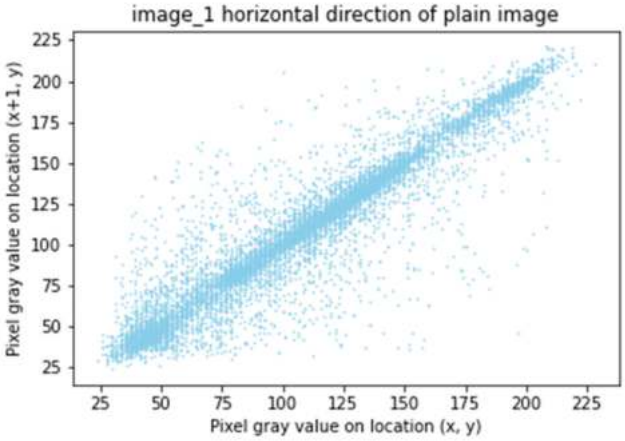
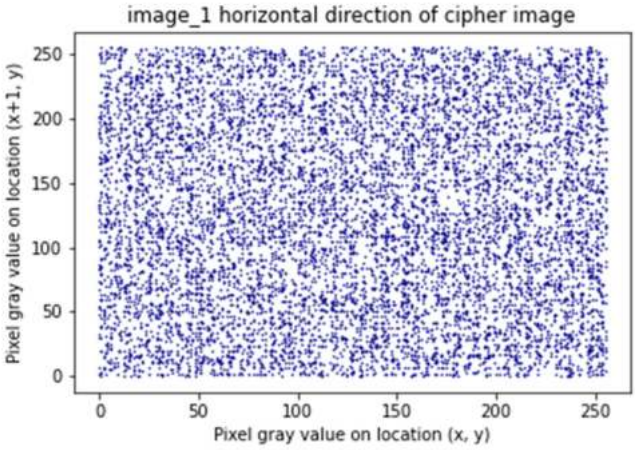


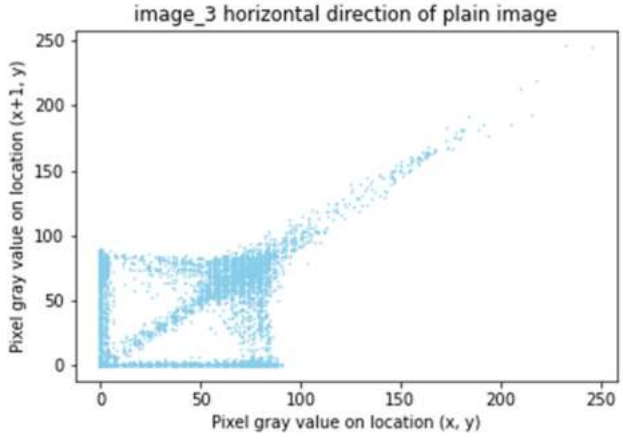
Fig. 19. Correlation coefficients of the plain and encrypted image in the horizontal, vertical and diagonal.











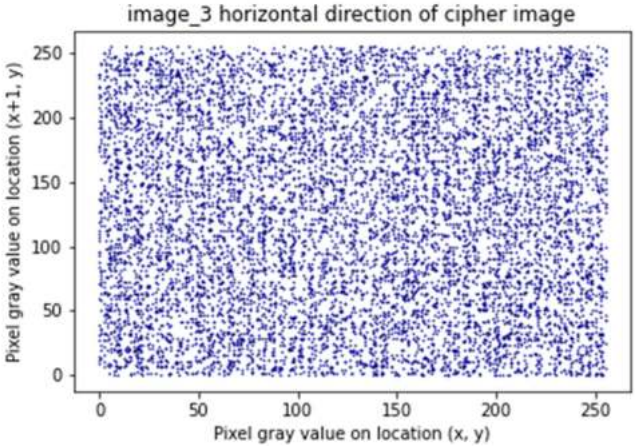


Fig. 19. (*continued*).

Table 4
Plain image and cipher image correlation coefficient in the vertical, diagonal, and horizontal direction.

Image name	Original image			Genetic hill climb			Simulated annealing		
	Vertical	Diagonal	Horizontal	Vertical	Diagonal	Horizontal	Vertical	Diagonal	Horizontal
Image_1	0.936756	0.914034	0.966708	9.7e-05	0.000809	−0.001513	−0.000741	0.003105	−0.003962
Image_2	0.845865	0.753027	0.852888	−0.008226	−0.002724	0.005994	−0.004255	−0.001084	−0.000428
Image_3	0.442394	0.440309	0.440271	0.003212	0.001237	−0.000249	−0.002066	0.003409	−0.000793
Image_4	0.978764	0.969448	0.978875	−0.000439	−0.00223	0.000922	0.002257	0.000504	0.00059
Image_5	0.940822	0.919419	0.941767	0.001529	−0.000761	2.5e-05	0.002624	0.000459	−0.001285
Image_6	0.983025	0.973168	0.989951	−0.001511	0.00029	−0.002639	-4e-06	−0.002383	0.003106
Image_7	0.99759	0.995939	0.998192	0.001126	−0.000423	0.002554	0.002422	−0.001185	0.000181
Image_8	0.882058	0.834933	0.933329	0.001117	0.003009	0.000799	0.001033	0.000127	−0.002202
Image_9	0.968638	0.936685	0.962609	6.8e-05	−9.4e-05	−6.5e-05	0.250232	0.25383	0.255514
Image_10	0.887737	0.826111	0.89895	0.001656	−0.000524	−0.001162	0.00178	0.000171	−0.000249

Table 5
Horizontal, vertical, and diagonal contrast values for plain and cipher images.

Image name	Original image			Genetic hill climb			Simulated annealing		
	Vertical	Diagonal	Horizontal	Vertical	Diagonal	Horizontal	Vertical	Diagonal	Horizontal
Image_1	254.039614	345.894487	134.060218	10,898.589246	10,888.302284	10,913.363189	10,851.312914	10,810.927213	10,885.171768
Image_2	907.680569	1454.248991	865.94194	11,037.591828	10,978.066957	10,883.186132	10,984.660814	10,952.946723	10,946.485095
Image_3	1540.183266	1547.189125	1546.264705	10,890.990691	10,913.429391	10,930.336154	10,964.742798	10,905.218051	10,953.53262
Image_4	122.666811	176.441297	121.98689	10,925.40336	10,944.926479	10,910.346307	10,875.98856	10,894.944895	10,894.370861
Image_5	718.630033	978.933667	707.220831	10,910.639439	10,935.793089	10,927.328274	10,900.421233	10,924.496117	10,943.913765
Image_6	130.736213	206.838002	77.355163	10,923.918619	10,904.579256	10,935.780157	10,922.493594	10,948.272169	10,888.418018
Image_7	34.25758	57.695769	25.726665	10,948.818031	10,966.063085	10,933.311349	10,894.412743	10,933.073599	10,917.516756
Image_8	935.513768	1308.701815	528.775022	10,907.32826	10,886.679989	10,910.832138	10,899.459805	10,909.019908	10,934.546124
Image_9	146.138616	295.044101	174.229581	10,931.009161	10,932.918384	10,932.399051	7394.536379	7359.426842	7342.757700
Image_10	479.069466	742.119379	433.447717	10,901.4611	10,925.377448	10,932.471862	10,876.090467	10,893.788095	10,898.309924

following equation:

$$e_i = \frac{M \times N}{256}$$

The ideal chi-square value of a histogram to be resistant to histograms attacks with a significance level of 0.05 should be, $x^2_{th}(255, 0.05) = 293$. Thus, according to Table 6, all chi-square values for the genetic hill climb algorithm and other than two values from the simulated annealing method are less than the ideal value thus, showing resistance to histogram attacks. For the encrypted image_9 and image_10 using simulated annealing, the chi-square value is greater than ideal thus, it is not resistant to histogram attacks. According to the chi-square results, genetic hill climb outperformed simulated annealing for image encryption.

5.3.6. Quality of encryption

Each of the images encrypted have been ciphered using adapting parameters to optimize the UACI value. While UACI does play an important role in determining the encryption, the parameters such as NPCR, key space and key sensitivity give us additional insights into the success or failure of the encrypted image.

5.3.7. NPCR

Number of pixels change rate is a parameter used to determine the resistance of cipher images to differential attacks. The difference between the plain image and the ciphered image in percentage is defined as the NPCR. The ideal value for NPCR is above 99.60 %. It is defined as follows:

$$NPCR = \frac{\sum D(i,j)}{H \times W} \times 100\%$$

$$D(i,j) = \begin{cases} 0, & C1 \neq C2 \\ 1, & C1 = C2 \end{cases}$$

where $D(i, j)$ is a bipolar array and, C_1 and C_2 are cipher and original images.

According to Table 7, the NPCR value after encryption is best for genetic hill climb. All the values obtained using genetic hill climb are above the ideal threshold. For simulated annealing, few values (image_1 and image_3) have NPCR values less than the ideal threshold. The non-adaptive method performs the worst and has several images with less-than-ideal threshold. Thus, genetic hill climb approach gave the ideal NPCR values.

5.3.8. Key space

Key space can be defined as the total number of possible keys which can be produced to be used in the encryption method. To create an encryption system which is resistant to brute force attacks, the cryptosystem should have a key space which is greater than 2^{100} [47] The proposed algorithm consists of five pseudorandom sequence generators namely, (i) Logistic map (r and x_{seed}), (ii) Tent map (r and x_{seed}), (iii) Henon map (x_{seed}, y_{seed} and a), (iv) Rossler map (c) and (v) Linear feedback shift register ($lfsr_{seed}$). For the first four methods, a precision of 10^{-16} can be used for each parameter. For the proposed encryption scheme, an image is encrypted only if a minimum of three sequences are used.

The minimum key space can be summarized as:

$$\text{Key space} = \left(4 \times \left(10^{16}\right) + 2^8\right) = 2^{220}$$

The maximum key space can be summarized as:

$$\text{Key space} = \left(8 \times \left(10^{16}\right) + 2^8\right) = 2^{432}$$

Since the minimum key space is greater than 2^{100}, the proposed encryption scheme is resistant to brute force attacks.

5.3.9. Key sensitivity

Even a small change in the keys of the encryption system should lead to generation of a new chaotic sequence dissimilar from the original one. In the proposed algorithm, four dynamical systems in their chaotic states are used to generate the chaotic sequences used for encryption. Even a small change in the parameters in the chaotic range of parameters for

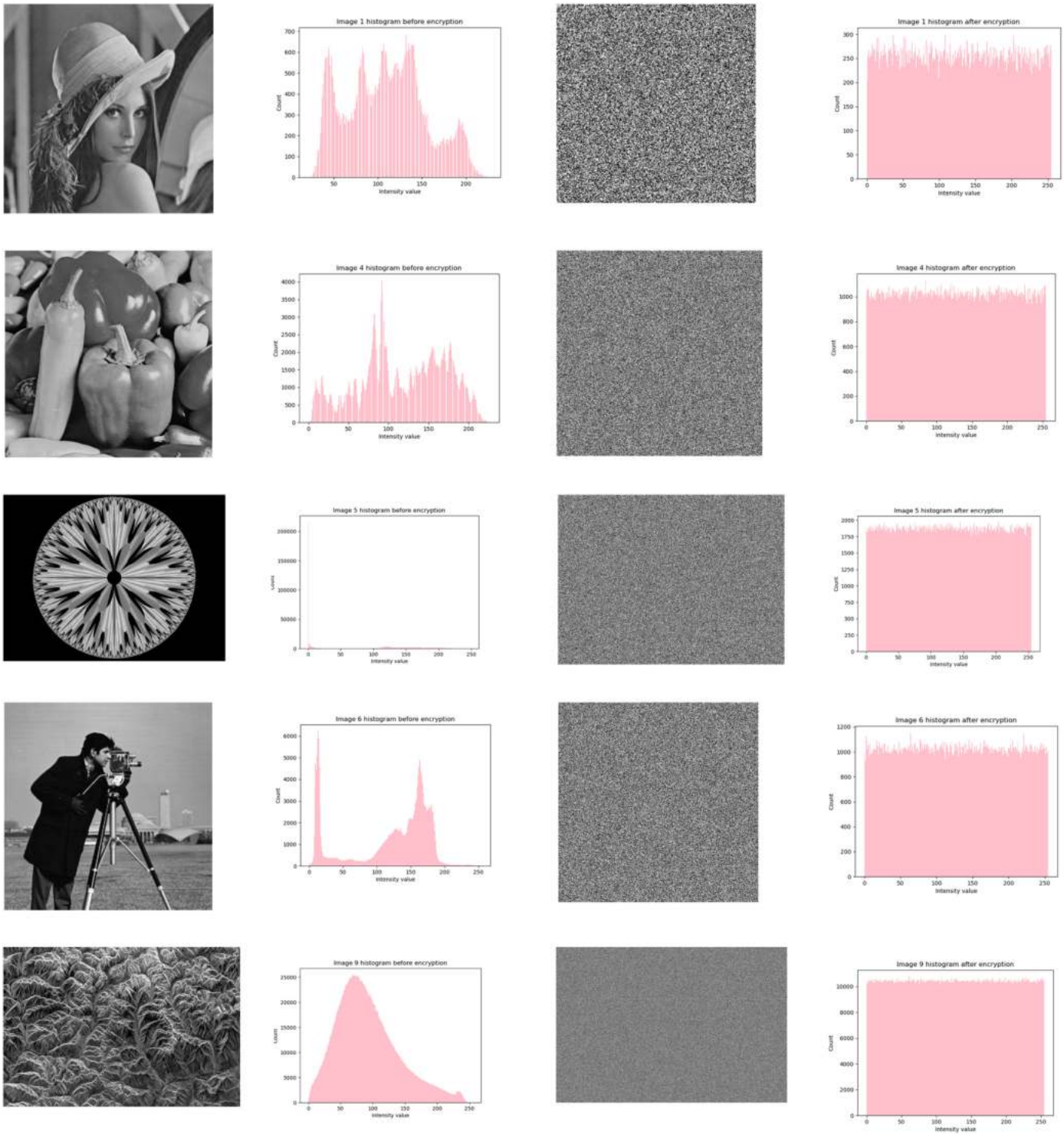


Fig. 20. Analysis of histograms (original image, original image histogram, cipher image, cipher image histogram).

Table 6
Chi-square test results for genetic hill climb algorithm and simulated annealing ciphered images with 0.05 significance level.

Image name	Genetic algorithm Chi-square	Simulated annealing Chi-square
Image_1	229.664062	278.765625
Image_2	237.458689	293.897435
Image_3	216.894586	237.658119
Image_4	275.691406	219.435546
Image_5	254.260266	294.988799
Image_6	248.544921	225.015625
Image_7	276.844752	245.379011
Image_8	264.498069	231.903474
Image_9	272.774805	305.953887
Image_10	254.779239	319.307992

Table 7
NPCR values for ciphered non adaptive, genetic hill climb and simulated annealing approaches.

Image name	NPCR		
	Non-adaptive	Genetic algorithm	Simulated annealing
Image_1	99.600220	99.644470	99.597168
Image_2	99.658889	99.637777	99.618889
Image_3	99.595556	99.602222	99.578889
Image_4	99.624634	99.615478	99.603653
Image_5	99.587708	99.603750	99.606458
Image_6	96.174240	99.621200	99.614334
Image_7	99.608062	99.621612	99.603997
Image_8	99.618323	99.609779	99.613548
Image_9	99.607950	99.606714	99.609785
Image_10	99.589197	99.606805	99.613468

these dynamical system leads to a large change in the value. It was found that a change of 10^{-16} in the original parameters caused a completely new chaotic sequence to be formed thus, chaoticity adds to the sensitivity of the sequence generated.

6. Discussion

According to the results obtained, it can be concluded that the inclusion of an adaptive mechanism did improve the quality of encryption. Using an adaptive system almost certainly guaranteed better encryption results in comparison to using a non-adaptive system. Two metaheuristic approaches for parameter evolution were discussed which were the genetic hill climb and simulated annealing algorithm. Both these algorithms showed improved results in comparison to the non-adaptive encryption system. Simulated annealing, genetic hill climb and non-adaptive image encryption showed similar performance in terms of entropy, correlation, and contrast. The UACI value obtained from non-adaptive image encryption was below par. Genetic hill climb algorithm outperformed simulated annealing algorithm in terms of NPCR and chi-square test.

As described in the introductory section, PixAdapt fits all definitions of an adaptive system. Both these approaches show adaptive behaviour with the main aim to optimize the quality of encryption and display resistance against statistical, brute-force, plain-text and cipher-text attacks. UACI was used as a parameter to measure the fitness of the encrypted image. The results obtained also reaffirmed the fact that using UACI as a fitness parameter was appropriate. The entropy, NPCR, correlation, contrast, and histogram analysis revealed that using UACI as a fitness parameter did optimize the other parameters as well. These image evaluation parameters showed near ideal results for adaptive image encryption as well adaptive sequence encryption approaches. Further, the cryptosystem showed a very high key space and key sensitivity which proved that the overall adaptive mechanism worked well for all kinds of images.

According to Tables 2, 3, 4 and Fig. 20, the UACI, entropy, correlation values for both techniques (Genetic Hill Climb & Simulated Annealing) showed similar results. While for metrics such as Contrast, chi-square, and NPCR, the Genetic Hill Climb algorithm displayed slightly better results in comparison to Simulated Annealing. Both the adaptive approaches showed significantly better results than the static parameters in terms of statistical, histogram and quality of encryption analysis.

The experimentation conducted was to test the performance of the adaptive mechanism with the aim of optimizing fitness. PixAdapt was tested for parameter evolution, fitness optimization, statistical analysis, and quality of encryption using non-adaptive, simulated annealing, and genetic hill climb methods. Genetic hill climb performed better than all other methods therefore, the system also extended its adaptiveness by activating and deactivating a few pseudo-random sequences based on image thus, creating a dependency on the original image as well. The addition of a switching mechanism proved to increase the overall adaptiveness of the algorithm. This mechanism improved the search space of the metaheuristic and genetic algorithms by adding more detail into the behaviour of the cipher image. Without the switching mechanism, the non-adaptive image encryption sequence did not produce results in the acceptable range thus, cementing its place in the adaptive mechanism.

During experimentation, all the parameters were adapted to optimize the solutions and it can be concluded that using a single set of parameters to encrypt an image is not ideal. Through this paper, it was also discovered that using more than a single pseudo-random sequence may be needed to generate a key that will produce ideal image encryption results.

In the proposed image encryption scheme, there are several parameters used to generate chaotic maps and a loop which is used to reach convergence (fitness in the acceptable range). These factors contribute to high time complexity for the overall algorithm and can be overcome using approaches such as parallel processing [48], map reduce [49] or GPUs [50]. Additionally, the permutation process can be split into several parts and the ideal range of values could be used to encrypt the image [51].

Fig. 21 shows the skeleton workflow of an encryption scheme using parallel processing. The chaotic sequence and the flattened original image will be partitioned into smaller parts such that each of the sub-partitions will be handed by a separate processor or a GPU. Individual chaotic sequence will be used to encrypt the partitioned image as shown in the figure. The encrypted partitions will then be combined to obtain the final image. Additionally, for the algorithm proposed in this paper, each set of parameters from the initial population can be used to generate a chaotic sequence which will be handled by a single processor or process in n processes/processors. These chaotic sequences will encrypt the entire image and they will be checked against the fitness condition. If the fitness of the newly encrypted image from a single process is better than the current benchmark, the new sequence will be used to check the fitness with the next set of parameters until convergence is achieved.

7. Conclusion and future scope

Most image encryption algorithms use only a static set of image encryption parameters to encrypt an image and do not consider the features present in the image. In this paper, PixAdapt – an adaptive image encryption process was proposed and implemented. PixAdapt uses an evolutionary algorithm to encrypt an image by evolving parameters to achieve optimal fitness in an accepted range of values. The evolutionary algorithm or metaheuristic algorithm activates itself to evolve the parameters when the optimal fitness is not obtained. Through this research, it was observed that UACI is an appropriate parameter to be used as a fitness function. A novel approach towards activating and deactivating the pseudorandom sequences was implemented in this paper. The addition of this process demonstrated better results than non-

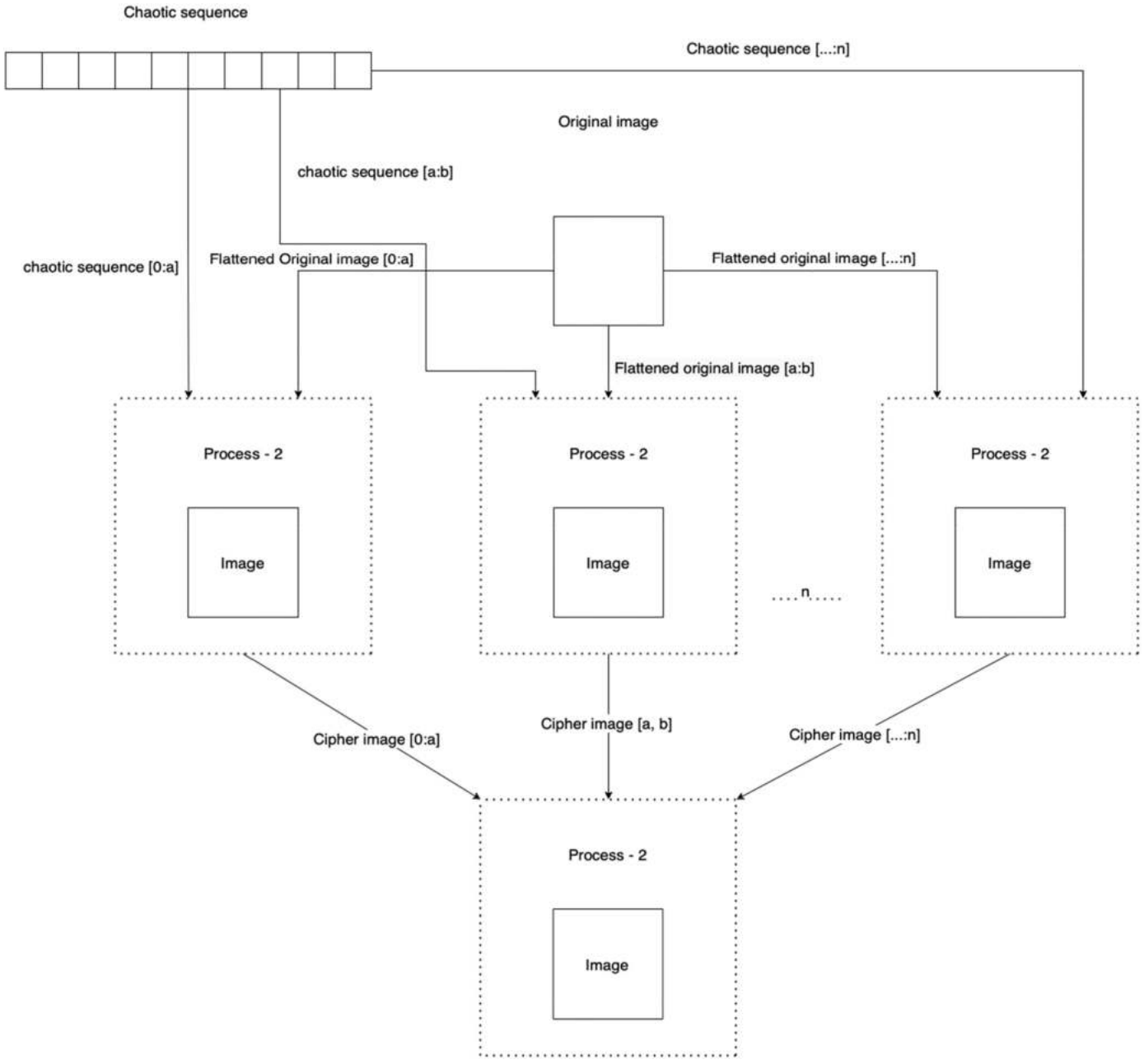


Fig. 21. Parallel image encryption.

adaptive parameters which did not use the switching mechanism. An additional factor which was observed that the switching mechanism did provide an extra layer of adaptiveness. The same algorithm can be tested on more image encryption evaluation parameters. To efficiently encrypt images, the method can be implemented using parallel processing. The adaptive mechanisms can be improved by running the algorithm on more parameters and incorporating new genetic algorithms and meta-heuristic search techniques. Further, PixAdapt can also be extended to encrypting Medical and color images.

Funding

There is no funding associated with this research study.

CRediT authorship contribution statement

Serial number	Author name	Contribution
1	Rohan Tuli	Conceptualization, Methodology, original draft preparation, implementation, analysis & proof reading
2	Hitesh Soneji	Methodology, implementation, analysis, & editing
3	Prathamesh Churi	Methodology, literature survey, analysis, & reviewing

Declaration of competing interest

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

Data availability

Data will be made available on request.

Acknowledgment

The authors would like to thank the University of Sussex for Open Access support. We would also like to acknowledge the feedback given by Dr Chris Johnson (Informatics department, University of Sussex) for incorporation of adaptation in Image Encryption.

References

[1] Kumar K, Roy S, Rawat U, Malhotra S. IEHC: an efficient image encryption technique using hybrid chaotic map. Chaos, SolitonsFractals 2022;158:111994. https://doi.org/10.1016/j.chaos.2022.111994.
[2] Yildirim M. Optical color image encryption scheme with a novel DNA encoding algorithm based on a chaotic circuit. Chaos, SolitonsFractals 2022;155:111631. https://doi.org/10.1016/j.chaos.2021.111631.
[3] Loan NA, Parah SA, Sheikh JA, Akhoon JA, Bhat GM. Hiding Electronic Patient Record (EPR) in medical images: a high capacity and computationally efficient technique for e-healthcare applications. J Biomed Inform 2017;73:125–36. https://doi.org/10.1016/j.jbi.2017.08.002.
[4] Wu J, Xie J, Bardakoff A, Blattner T, Keyrouz W, Bhattacharyya SS. CGMBE: a model-based tool for the design and implementation of real-time image processing applications on CPU–GPU platforms. JReal-Time Image Process 2021;18(3): 561–83. https://doi.org/10.1007/s11554-020-00994-9.
[5] Wang X, Du X. Pixel-level and bit-level image encryption method based on Logistic-Chebyshev dynamic coupled map lattices. Chaos, SolitonsFractals 2022;155: 111629. https://doi.org/10.1016/j.chaos.2021.111629.
[6] Dokeroglu T, Sevinc E, Kucukyilmaz T, Cosar A. A survey on new generation metaheuristic algorithms. ComputIndEng 2019;137:106040. https://doi.org/10.1016/j.cie.2019.106040.
[7] Narendra KS. Chapter two - hierarchical adaptive control of rapidly time-varying systems using multiple models. In: Vamvoudakis KG, Jagannathan SBT-CofCS, editors. Butterworth-Heinemann; 2016. p. 33–66. https://doi.org/10.1016/B978-0-12-805246-4.00002-1.
[8] Hua Z, Zhang K, Li Y, Zhou Y. Visually secure image encryption using adaptive-thresholding sparsification and parallel compressive sensing. Signal Process 2021; 183:107998. https://doi.org/10.1016/j.sigpro.2021.107998.
[9] Liu W, Sun K, Zhu C. A fast image encryption algorithm based on chaotic map. Opt Lasers Eng 2016;84:26–36. https://doi.org/10.1016/j.optlaseng.2016.03.019.
[10] Wang Jun, Zhi X, Chai X, Lu Y. Chaos-based image encryption strategy based on random number embedding and DNA-level self-adaptive permutation and diffusion. Multimed Tools Appl 2021;80(10):16087–122. https://doi.org/10.1007/s11042-020-10413-7.
[11] Wu Y, Zhang L, Qian T, Liu X, Xie Q. Content-adaptive image encryption with partial unwinding decomposition. Signal Process 2021;181:107911. https://doi.org/10.1016/j.sigpro.2020.107911.
[12] Hsu C-Y, Cheng Y-C, Lin S-F. Efficient and accurate image alignment using TSK-type neuro-fuzzy network with data-mining-based evolutionary learning algorithm. EURASIP JAdvSignal Process 2011;2011(1):96. https://doi.org/10.1186/1687-6180-2011-96.
[13] Bruha I. Defining adaptive and learning systems. CybernSyst 1989;20(1):77. https://doi.org/10.1080/01969728908902194.
[14] Li C-L, Zhou Y, Li H-M, Feng W, Du J-R. Image encryption scheme with bit-level scrambling and multiplication diffusion. Multimed Tools Appl 2021;80(12): 18479–501. https://doi.org/10.1007/s11042-021-10631-7.
[15] Zhou Y, Li C, Li W, Li H, Feng W, Qian K. Image encryption algorithm with circle index table scrambling and partition diffusion. Nonlinear Dyn 2021;103(2): 2043–61. https://doi.org/10.1007/s11071-021-06206-8.
[16] Nature-inspired optimization algorithms. In: Yang X-SBT-N-IOA, editor. Apple Academic Press. Elsevier; 2014. i. https://doi.org/10.1016/B978-0-12-416743-8.00016-6.
[17] Chira C, Horvath D, Dumitrescu D. Hill-climbing search and diversification within an evolutionary approach to protein structure prediction. BioData Min 2011;4(1): 23. https://doi.org/10.1186/1756-0381-4-23.
[18] Bertsimas D, Tsitsiklis J. Simulated annealing. StatSci 1993;8(1):10–5. https://doi.org/10.1214/ss/1177011077.
[19] Rohith S, Bhat KNH, Sharma AN. Image encryption and decryption using chaotic key sequence generated by sequence of logistic map and sequence of states of Linear Feedback Shift Register. In: 2014 International Conference on Advances in Electronics Computers and Communications; 2014. p. 1–6. https://doi.org/10.1109/ICAECC.2014.7002404.
[20] Naik RB, Singh U. A review on applications of chaotic maps in pseudo-random number generators and encryption. AnnData Sci 2022. https://doi.org/10.1007/s40745-021-00364-7.
[21] Wang X, Xu D. Image encryption using genetic operators and intertwining logistic map. Nonlinear Dyn 2014;78(4):2975–84. https://doi.org/10.1007/s11071-014-1639-z.
[22] Das S, Mandal SN, Ghoshal N. Multiple-image encryption using genetic algorithm. AdvIntellSystComput 2015;343:145–53. https://doi.org/10.1007/978-81-322-2268-2_16.
[23] Ghazvini M, Mirzadi M, Parvar N. A modified method for image encryption based on chaotic map and genetic algorithm. Multimed Tools Appl 2020;79(37): 26927–50. https://doi.org/10.1007/s11042-020-09058-3.
[24] Shankar K, Eswaran P. An efficient image encryption technique based on optimized key generation in ECC using genetic algorithm. In: Artificial intelligence and evolutionary computations in engineering systems; 2016. p. 705–14.
[25] Pareek NK, Patidar V. Medical image protection using genetic algorithm operations. Soft Comput 2016;20(2):763–72. https://doi.org/10.1007/s00500-014-1539-7.
[26] Enayatifar R, Abdullah AH, Isnin IF. Chaos-based image encryption using a hybrid genetic algorithm and a DNA sequence. Opt Lasers Eng 2014;56:83–93. https://doi.org/10.1016/j.optlaseng.2013.12.003.
[27] Abdullah AH, Enayatifar R, Lee M. A hybrid genetic algorithm and chaotic function model for image encryption. AEU-Int J Electron C 2012;66(10):806–16. https://doi.org/10.1016/j.aeue.2012.01.015.
[28] Abbasi AA, Hosseini R, Mazinani M. A novel image encryption model based on hybridization of genetic algorithm, chaos theory and lattice map. JAdvComputRes 2018;9(4):129–44. http://jacr.iausari.ac.ir/article_661390.html, http://jacr.iausari.ac.ir/article_661390.html.
[29] Wang Jian. Digital image encryption algorithm design based on genetic hyperchaos. IntJOpt 2016;2016:2053724. https://doi.org/10.1155/2016/2053724.
[30] Wong K-W, Yap W-S, Wong DC-K, Phan RC-W, Goi B-M. Cryptanalysis of genetic algorithm-based encryption scheme. Multimed Tools Appl 2020;79(35):25259–76. https://doi.org/10.1007/s11042-020-09191-z.
[31] Mozaffari S. Parallel image encryption with bitplane decomposition and genetic algorithm. Multimed Tools Appl 2018;77(19):25799–819. https://doi.org/10.1007/s11042-018-5817-8.
[32] Liu H, Zhao B, Huang L. A novel quantum image encryption algorithm based on crossover operation and mutation operation. Multimed Tools Appl 2019;78(14): 20465–83. https://doi.org/10.1007/s11042-019-7186-3.
[33] Kaur M, Kumar V. Beta chaotic map based image encryption using genetic algorithm. IntJBifurcationChaos 2018;28(11):1850132. https://doi.org/10.1142/S0218127418501328.
[34] Kaur M, Kumar V. Parallel non-dominated sorting genetic algorithm-II-based image encryption technique. Imaging SciJ 2018;66:1–10. https://doi.org/10.1080/13682199.2018.1505327.
[35] Ferdush J, Mondol G, Prapti AP, Begum M, Sheikh MNA, Galib SM. An enhanced image encryption technique combining genetic algorithm and particle swarm optimization with chaotic function. IntJComputApplic 2021;43(9):960–7. https://doi.org/10.1080/1206212X.2019.1662170.
[36] Nematzadeh H, Enayatifar R, Motameni H, Guimarães F, Coelho V. Medical image encryption using a hybrid model of modified genetic algorithm and coupled map

lattices. Opt Lasers Eng 2018;110. https://doi.org/10.1016/j.optlaseng.2018.05.009.
[37] Hasheminejad A, Rostami MJ. A novel bit level multiphase algorithm for image encryption based on PWLCM chaotic map. Optik 2019;184. https://doi.org/10.1016/j.ijleo.2019.03.065.
[38] Mahmud M, Rahman A, Lee M, Choi J. Evolutionary-based image encryption using RNA codons truth table. OptLaser Technol 2020;121:1–8. https://doi.org/10.1016/j.optlastec.2019.105818.
[39] He Y, Li P, Wang X-Y. A new color image encryption scheme based on 2DNLCML system and genetic operations. Opt Lasers Eng 2020;128:106040. https://doi.org/10.1016/j.optlaseng.2020.106040.
[40] Kaur M, Singh D, Sun K, Rawat U. Color image encryption using non-dominated sorting genetic algorithm with local chaotic search based 5D chaotic map. Future GenerComputSyst 2020;107:333–50. https://doi.org/10.1016/j.future.2020.02.029.
[41] Dua M, Wesanekar A, Gupta V, Bhola M, Dua S. Differential evolution optimization of intertwining logistic map-DNA based image encryption technique. JAmbient IntellHumComput 2020;11. https://doi.org/10.1007/s12652-019-01580-z.
[42] Cheng G, Wang C, Xu C. A novel hyper-chaotic image encryption scheme based on quantum genetic algorithm and compressive sensing. Multimed Tools Appl 2020;79. https://doi.org/10.1007/s11042-020-09542-w.
[43] Gupta A, Singh D, Kaur M. An efficient image encryption using non-dominated sorting genetic algorithm-III based 4-D chaotic maps: image encryption. JAmbient IntellHumComput 2020;11(3):1309–24. https://doi.org/10.1007/s12652-019-01493-x.
[44] Niu Y, Zhou Z, Zhang X. An image encryption approach based on chaotic maps and genetic operations. Multimed Tools Appl 2020;25613–25633. https://doi.org/10.1007/s11042-020-09237-2.
[45] Suri S, Vijay R. A pareto-optimal evolutionary approach of image encryption using coupled map lattice and DNA. Neural ComputApplic 2020;32(15):11859–73. https://doi.org/10.1007/s00521-019-04668-x.
[46] Chai X, Zhi X, Gan Z, Zhang Y, Chen Y, Fu J. Combining improved genetic algorithm and matrix semi-tensor product (STP) in color image encryption. Signal Process 2021;183:108041. https://doi.org/10.1016/j.sigpro.2021.108041.
[47] Tuli R, Soneji H, Vahora S, Churi P, Bangalore NM. PixJS: a novel chaos-based approach for image encryption. ConcurrComputPractExp 2022:e6990. https://doi.org/10.1002/cpe.6990. n/a(n/a).
[48] Song W, Zheng Y, Fu C, Shan P. A novel batch image encryption algorithm using parallel computing. Inform Sci 2020;518:211–24. https://doi.org/10.1016/j.ins.2020.01.009.
[49] Al-Khasawneh MA, Uddin I, Shah SAA, Khasawneh AM, Abualigah L, Mahmoud M. An improved chaotic image encryption algorithm using Hadoop-based MapReduce framework for massive remote sensed images in parallel IoT applications. ClustComput 2022;25(2):999–1013. https://doi.org/10.1007/s10586-021-03466-2.
[50] Bharadwaj B, Saira Banu J, Madiajagan M, Ghalib MR, Castillo O, Shankar A. GPU-accelerated implementation of a genetically optimized image encryption algorithm. Soft Comput 2021;25(22):14413–28. https://doi.org/10.1007/s00500-021-06225-y.
[51] Wang X, Zhao H. Fast image encryption algorithm based on parallel permutation-and-diffusion strategy. Multimed Tools Appl 2020;79(27):19005–24. https://doi.org/10.1007/s11042-020-08810-z.