Arab J Sci Eng (2018) 43:3927–3950
https://doi.org/10.1007/s13369-017-2751-4

RESEARCH ARTICLE - SPECIAL ISSUE - COMPUTER ENGINEERING AND COMPUTER SCIENCE

Image Steganography Capacity Improvement Using Cohort Intelligence and Modified Multi-Random Start Local Search Methods

Dipti Kapoor Sarmah[1] · **Anand J. Kulkarni**[1,2]

Received: 29 March 2017 / Accepted: 16 July 2017 / Published online: 28 July 2017

Abstract In this paper, we have proposed two steganographic techniques which use JPEG compression on greyscale image to hide secret text. JPEG compression is based on discrete cosine transform technique. In order to improve the capacity, a quantization table of size 16 × 16 is practiced instead of a standard JPEG quantization table. Also, the proposed work presents two novel optimization algorithms applied on steganography which are based on the concept of cohort intelligence (CI) with cognitive computing (CC) and Multi-random start local search (MRSLS) algorithm. CI is an emerging optimization algorithm inspired from social learning of one another. This algorithm is being tested to solve unconstrained, constrained and NP-hard combinatorial problems and shows promising results. CC involves self-learning systems and is an emerging area in the field of machine learning. In the proposed work, CI, CC and MRSLS which is inspired from duo-swapping approach and tested to solve NP-hard combinatorial problems, are combined and applied to steganography to produce good results. This work has modified the MRSLS algorithm and applied to steganography to test and validate our results with other comparable algorithms. Experiments are done to test six greyscale images. Experimental results will reveal the quality of stego-images and the secret message embedding capacity.

✉ Anand J. Kulkarni
kulk0003@uwindsor.ca; anand.kulkarni@sitpune.edu.in; kulk0003@ntu.edu.sg

Dipti Kapoor Sarmah
dipti.sarmah@sitpune.edu.in

[1] Symbiosis Institute of Technology, Symbiosis International University, Pune, MH 412 115, India

[2] Odette School of Business, University of Windsor, 401 Sunset Avenue, Windsor, ON N9B 3P4, Canada

Keywords Steganography · JPEG compression · Cohort intelligence · Multi-random start local search · Discrete cosine transform

1 Introduction

Now a days, digital media is widely used for communication over the internet which enables the need for a secure and robust communication. Data security is one of the primary concerns in this regard which necessitates development of a robust technique to protect sensitive information. Cryptography and steganography are two sciences in the field of information security. According to Coron [1], cryptography is the method used for transforming the secret message into some unintelligible form. The original secret message is known as plain text, and the scrambled message of unintelligible form could be referred to as cipher text. There are two processes involved in cryptography: encryption and decryption. Encryption converts the plaintext to cipher text by applying suitable cryptographic algorithm at the sender side, whereas decryption does the reverse process of encryption, converting cipher text to plain text at the receiver side. Though the complexity for extracting the secret message is increased due to cryptography, the hacker who attacks the network still may try to convert the cipher text to plain text. The number of attacks on the network is also not diminished as the opponent may easily guess that some secret information is passing through the network [2]. Due to this limitation, researchers are motivated to develop improved cryptography techniques. An alternative to cryptography is steganography [3,4] in which the secret information is hidden into a digital media. The digital media can be in the form of image, video, audio, etc. The original digital media is referred to as cover media. The media in which the secret information is

hidden referred to as stego-media. The quality of the stego-media should be maintained in such a way that it should not draw any attention of unauthorized user about the concealed information [5]. Therefore, steganography is considered significant for a variety of computer applications in order to improve communication security.

In this paper, we have considered image as a most common digital media used over the internet. In order to embed a secret message in an Image, some trivial distortions are involved in the nonsignificant parts of the image, which can be generated using quantization noise after digitalization [6]. In image-based steganography, an original image is used for hiding a secret message referred to as cover image and the unified image, after embedding the secret message referred to as stego-image [7–11]. Spatial domain steganography and transform domain steganography [7,8,12] are considered as two major categories of steganography methods. In spatial domain steganography, the pixel values of an image are directly changed for hiding the digital form of a secret message. This involves the basic concept of least significant bits (LSBs) substitution of the pixel values of the cover image with the secret message bits [13]. The transform domain steganography also referred to as frequency domain steganography. It transforms the cover image from spatial domain to frequency domain. Furthermore, discrete cosine transform (DCT) [14], discrete wavelet transform (DWT) [15], discrete fourier transform (DFT) [16], etc., are the different methods used under this category which transform the cover image into frequency components. The transformed coefficients are used to hide the digital form of the secret message. The three vital requirements of any steganography technique are capacity, security and robustness [17] while maintaining the quality of an image.

Image Quality is one of the evaluation criteria to take a judgement on any steganography method and can be evaluated by calculating peak signal-to-noise ratio (PSNR) value (refer to Eq. 1) where the value of MAX is dependent on the number of bits per pixel. If 8 bits are considered in a sample for representation of a pixel, the value of MAX will be $(2^8 - 1)$, i.e. 255. The PSNR is measured in decibel (DB) unit, and its typical values for a lossy image are in between 30 and 50 DB for a bit depth of 8 bits. Value towards higher side indicates that image quality is better and vice versa. The PSNR is inversely proportional to mean square error (MSE) (refer to Eq. 2). The MSE calculates the difference between the pixel values of a stego-image and its corresponding cover image to indicate the percentage of error where W and H are the width and height, respectively, of the greyscale image. $S(i, j)$ and $C(i, j)$ specify the pixels values at ith and jth location of stego-image and cover image, respectively.

$$\mathrm{PSNR} = 10 \times \log_{10}\left(\frac{\mathrm{MAX}^2}{\mathrm{MSE}}\right) \tag{1}$$

$$\mathrm{MSE} = \frac{1}{\mathrm{WH}}\sum(S(i, j) - C(i, j))^2 \tag{2}$$

Capacity refers to the maximum number of bits of the secret message embedded in the cover image without disturbing (degenerating) the characteristics of the image. The capacity of the embedding secret message should be less than the size of the cover image [7]. If any steganalyst is incapable of extracting the hidden secret message, it signifies that the message is secure. In order to make the steganography method more efficient, the quality of the stego-image such as its characteristics and attributes should be maintained as similar as the cover image during embedding process [18]. Robustness can be described as the ability to resist manipulations in the stego-image after common image processing operations and compression [19]. In order to achieve high level of robustness, steganography systems should have the capability to withstand and secure the secret message during stego-attacks [17]. The preferable domain of a steganography method is to transform domain such as DCT, DFT, DWT over spatial domain as higher robustness can be achieved against image processing operations, compression and attacks [20].

Many researchers have combined the idea of cryptography and steganography in order to attain better security of secret message during transmission. To transfer the stego-image in a faster way to the recipient, the size of the cover image can be considered as a hurdle which can be avoided using different compression techniques. There are two types of compression techniques: lossless compression technique and lossy compression technique. The lossy compression technique [14] is preferred over the earlier as it may reduce the image size significantly by discarding the redundant data. However, it is difficult to get the original image after decompression as image properties may get distorted. The most popular lossy compression technique is DCT-based JPEG image compression [21]. It can be applied to greyscale as well as colour image. There are few steps associated with this compression technique such as block preparation, discrete cosine transform (DCT), quantization, zigzag scan, digital pulse code modulation (DPCM) encoding, run length encoding (RLE), Huffman encoding and frame building. In this technique, a two-dimensional image is first digitized and converted into a form of pixels. It is further divided into 8×8 blocks; each block is transformed into coefficients using the DCT, which employs standard quantization table [22], and the bits of the secret message are hidden into these quantized values. Though this method is more secure than the methods used in spatial domain steganography, the great challenge is to achieve high capacity of the hidden secret text by maintaining the image quality [3]. In this regard, quantization process plays a major role in JPEG compression technique.

Quantization is the process of mapping a large set of continuous values to a small set of discrete values. There is a

standard JPEG quantization table of size 8×8 depending upon the size of number of image blocks which decides the quality of the stego-image and also controls its compression ratio. The values in this standard quantization table can be varied based upon the application (refer to "Appendix A" for details). Researchers have modified the values of the quantization table as per the requirement of their applications [8,23–25]. In order to increase the capacity of hidden secret text, the quantization table is further stretched to the form of 16×16 matrix. Jiang et al. [26] and Almohammad et al. [27] have used the 16×16 quantization table to generate quantized values which resulted not only in to a higher capacity on greyscale images, but also a better quality stego-image. The DCT generated quantized values are divided in to three frequency zones: low, middle and high wherein selection of low–middle frequency zones are considered safe to hide the secret message [28]. Since there is less number of significant DCT coefficients of 8×8 pixel blocks, more coefficients in the low–middle frequency zones can be used for embedding in 16×16-pixel blocks, which improves the secret message capacity and retains the quality of the stego-image. However, few researchers have proposed hiding secret data bits in high frequency zone as well [29].

In this paper, with the aim to improve the embedding capacity in greyscale image, we divide the cover image into non-overlapping blocks of 16×16 pixels and use an 16×16 optimal quantization table [27]. Researchers Kulkarni [30] and Jiang et al. [26] have already used this quantization table in a greyscale image; however, optimization techniques have not been applied yet in this combination. In order to explore further in this direction, an effort is made to implement two optimization techniques which are used to generate a cipher text and an optimization matrix. The embedding of this optimized cipher text is done in the DCT-quantized values generated using 16×16 quantization tables. This work is inspired by Li et al. [13], in which a JPEG steganography method based on Particle Swarm Optimization (PSO) [31] algorithm was proposed, referred to as JPEG-PSO. In this approach, cover image is divided into 8×8 pixel blocks and thus a 8×8 quantization table is used. The quantization table generated for joint quantization table modification (JQTM) [8] is modified and used to increase the embedded capacity. Li et al. [13] also applied the concept of optimal least significant bit substitution (OLSBS) [9] wherein the optimal substitution matrix is used to convert the secret message into a cipher message and the cipher message is then embedded into the cover image. Though the performance with regard to PSNR and embedding capacity is improved by OLSBS in comparison with JPEG-PSO, the method is implemented on spatial domain and the computational cost of calculating PSNR gets increased for increasing the length of the secret message which in turn increases the complexity for identifying the optimal substitution matrix. Li et al. [13] implemented a method on transform domain to enhance the security level. To further improve the security of this method, PSO is used to select the optimal substitution matrix. A substitution matrix is described by a particle having 2^D dimensions where the decimal values of D can be $0, 1, 2, \ldots, 2^d - 1$. Superior particle helps to identify the optimal substitution matrix. Though the security and embedded capacity is increased, there is still a chance of improving the methods in terms of embedded capacity while retaining the good quality of stego-image. Also, there is a big challenge involved to search for an optimal substitution matrix from number of matrices. The other optimization algorithms have also been applied in the field of steganography to overcome these issues; however, it has been found expensive for embedding high capacity of data and very time consuming. Thus, there is a need of implementing a steganography method which overcomes these issues.

Researchers have applied Genetic Algorithm (GA) [31–35], Particle Swarm Optimization (PSO) [13,36–38], Ant Colony Optimization (ACO) [39,40], etc., in the field of steganography. GA comes under the category of Evolutionary algorithm and is a population-based metaheuristic optimization algorithm. PSO is also a population-based stochastic optimization techniques having random solutions in its space called particles and comes under the category of swarm inspired methods. ACO technique also comes under this category, and it is inspired by the behaviour of ants.

There are two more emerging optimization algorithms inspired by social behaviour of candidates, referred to as cohort intelligence (CI) Kulkarni et al. [41] and multi-random start local search (MRSLS) Kulkarni et al. [42]. In CI algorithm, the random number of candidates in a cohort is considered as initialized in the search space. These cohort candidates compete with each other to achieve a common goal. Every candidate owns certain quality which transforms into their behaviour. During the process, each candidate learns, accepts and adopts certain quality of itself or the other candidates to improve their behaviour which improves the overall quality, i.e. the behaviour of the cohort. The variables and the objective function taken in the system are considered as quality and the cohort behaviour, respectively. The details of the original CI methodology are presented in "Appendix B". The validity of this algorithm is checked by considering different types of problems such as unconstrained [41], 0–1 Knapsack problem [43] categorized under NP-hard combinatorial problems, constrained combinatorial problems having large data sets/variables, etc. Krishnasamy et al. [44] proposed a new algorithm by combining two techniques of K means and modified cohort intelligence (MCI) to resolve the data clustering problem. In order to get good results in less time, a mutation operator is added in CI algorithm to make MCI algorithm. CI is further been applied for shell-and-tube heat exchanger to optimize its design and manufacturing cost in order to solve real-world problems in

mechanical engineering domain [45]. Furthermore, Kulkarni et al. [46] proposed two methods to solve constrained metaheuristic problems using CI. These methods are (1) CI with static penalty function and (2) CI with dynamic penalty function. These methods are validated to solve several constrained problems and its performance found better than the algorithms such as GA, PSO, ABC, d-Ds. CI is also validated on solving discrete and mixed variable problems from truss structure and engineering domain, respectively, and obtained promising results as well [47]. Recently, CI is applied for solving control problems as well [48]. CI produces better solutions in terms of quality, and it is found computationally analogous once compared with other algorithms. Since CI is based on the probabilistic approach and when applied for solving steganography problem addressed here, the convergence of candidate's PSNR value is not necessarily obtained. So, the concept of cognitive computing (CC) is employed. Cognitive computing is a concept that simulates the human thought process in a computerized model.

The other optimization method MRSLS is also used to solve constrained combinatorial problems. In this technique, the solutions defined in the search space interchange pairwise, i.e. the elements of the adjacent solutions are swapped with each other randomly. This method is used to solve NP-hard combinatorial problem such as the Cyclic Bottleneck Assignment Problem (CBAP) [42]. Both the algorithms solve the problems as described earlier in a very efficient manner; however, they have not been explored yet in the field of image processing, specifically in the area of steganography. Also, a random solution is accepted in MRSLS as an initial solution which may be an infeasible solution some time. Thus, there is a need to generate the initial feasible solutions which facilitated to modify MRSLS algorithm to produce better results.

Therefore, the work is done to implement two new algorithms referred to as cohort intelligence with cognitive computing (CICC) and modified multi-random start local search (M-MRSLS) algorithms. In this paper, CICC and M-MRSLS are used to identify the optimal substitution matrix. Also, they are used to convert the plain secret text into cipher text which is further used to hide the cipher text into 16×16 quantized values in JPEG as generated by the 16×16 quantization table as described before. Thus, there are two methods presented in the next section as follows.

(a) Steganography method using CICC optimization algorithm with 16×16 quantization table.
(b) Steganography method using M-MRSLS optimization algorithm with 16×16 quantization table.

These two novel methods make the system more secure and increase the stego-capacity by retaining the image quality.

Rest of the paper is organized as follows. Section 2 presents the proposed work using 16×16 quantization table. Section 3 deals with the results and discussions based on evaluation parameters and presents comparative results with respect to evaluation parameters using 16×16 quantization table and important snapshots. Conclusions and future directions are presented in Sect. 4. The methodology of JPEG compression, CI and MRSLS is explained in "Appendices A, B and C", respectively. References are written at the end.

2 Proposed Work Using 16 × 16 Quantization Table

In the proposed work, two optimization techniques CICC and M-MRSLS are developed to search for an optimal substitution matrix and convert the plain secret text to cipher text. The following Sect. 2.1 is used to describe the embedding procedure of secret text in JPEG image having 16×16 quantization table. The whole embedding procedure is divided into four phases. An illustration for searching an optimal substitution matrix using CICC and M-MRSLS is described in Subsection 2.1.1 Part 2, Figs. 3 and 5, respectively. The extraction procedure of secret text is also described in Subsection 2.1.2.

2.1 Algorithms

There are four phases considered in the embedding procedure as shown in Fig. 1: Greyscale image Pre-processing (Phase 1), Message encryption and optimal matrix identification (Phase 2), secret message insertion or embedding (Phase 3), and run length encoding (RLE), digital pulse code modification (DPCM), entropy coding (EC), Huffman coding (HC), frame building (FB) and getting a stego-image (Phase 4). We employ CICC and M-MRSLS to identify the optimal substitution matrix and convert plain secret message to cipher message. Since we have put an effort to solve the limitations of steganography by using 16×16 quantization table and by implementing and applying two optimization algorithms in JPEG greyscale image, it enables us to include the problem definition, objective function and constraints.
Problem definition: Increase the embedded capacity of secret message and improve the security of the system by increasing the quality of the image.
Inputs: There are three inputs considered for implementation of two steganography methods: greyscale cover image, secret message and number of substitution matrices.
Objectives: The following objectives are identified:

- Encrypt the secret message and embed the encrypted secret message (cipher message) into cover image to enhance the security level.
- Improve PSNR value for stego-image.
- Upgrade the cipher message capacity.

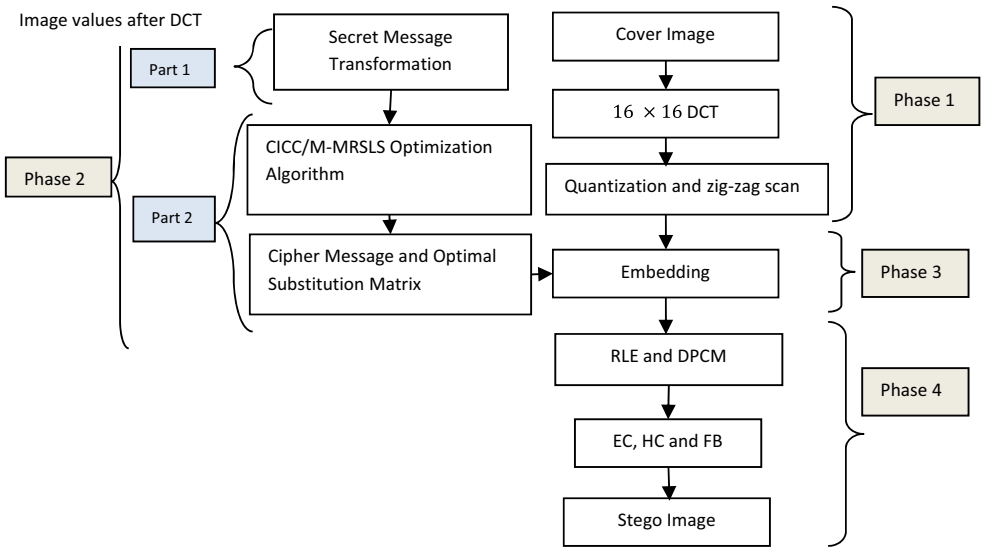


Fig. 1 Block diagram of embedding procedure

The Sect. 2.1.1 describes the Block Diagram of the embedding algorithm.

2.1.1 Embedding Algorithm

As discussed in the previous section, about the four phases of embedding procedure is described in this subsection. Figure 1 shows the flowchart for whole procedure. We have drawn every step of the two steganography methods in its respective phases.

(i) *Phase 1 (Segmentation of cover image and quantization):* This is pre-processing phase used to segment the greyscale cover image in to 16 × 16 pixels of non-overlapping image blocks. Then, DCT is applied to each block to transform its pixel value into DCT coefficients. DCT coefficients are further scaled using 16 × 16 quantization table as shown in Table 1. In this quantization table, the positions where $(value\ of\ the\ table\ element = 1)$ are used to embed the secret message. The quantized DCT coefficients are then rounded off to the nearest integers as similar to the JPEG image compression (refer "Appendix A"). In order to select the coefficients for embedding, a traversal is done in zigzag scan order. The table for zigzag scanning is shown in Table 2. There will be total 256 values starting from 1 to 256 for 16 × 16 image block matrix.

(ii) *Phase 2*: This phase describes the conversion of secret plain text to cipher text and is inspired from OLSBS method [9]. This phase is divided into two parts. Part-1 explains the secret message transformation, Part-2 describes and applies the optimization algorithm in the secret text and selects the optimal substitution matrix.

(a) Part-1 (Transformation of secret message)

We accept the secret message and apply the optimization algorithm either CICC or M-MRSLS to achieve the optimized matrix and get the cipher text. Accepted secret message is converted in to smaller number of bits. A decimal value d is selected to make individual groups having d number of bits. The decimal value d signifies the number of LSBs used from each DCT coefficient to hide d number of secret bits. The possible number of secret bit combinations is in the range from 0 to $2^d - 1$, which signifies the range of decimal values. The value of d is assumed as 2 for computational easiness. Thus, the range of decimal values for secret bits will be from 0 to 3. A substitution matrix M is considered to convert the plain secret bits to cipher secret bits. Substitution matrix M is represented as follows:

$$M = \left\{ m_{ij}, \quad 0 \le i, j \le 2^d - 1 \right\}$$
$$\text{where } m_{ij} = \begin{cases} 1, & if\ i \text{ replaces } j \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

Wang et al. [10] has considered the substitution matrix in the form of identity matrix and its variations. The total number of substitution matrices for the value d is $2^d!$ denoted as $M_1, \ldots, M_{2^d!}$. Thus, the total number of substitution matrices for the value $d = 2$ is 24 having size of 4 × 4 of each M.

The important steps for this phase are as follows:

(i) Accept the secret message.
(ii) Convert the secret message in to number of bits.
(iii) Select d number of bits to make separate groups having d bits.
(iv) Evaluate the decimal values of each individual group.

Table 1 16 × 16 Quantization table [27]

7	7	7	7	7	1	1	1	1	1	1	1	1	1	1	1
7	7	7	7	1	1	1	1	1	1	1	1	1	1	1	17
7	7	7	1	1	1	1	1	1	1	1	1	1	1	17	18
7	7	1	1	1	1	1	1	1	1	1	1	1	17	18	20
7	1	1	1	1	1	1	1	1	1	1	1	17	18	20	22
1	1	1	1	1	1	1	1	1	1	1	17	18	20	22	24
1	1	1	1	1	1	1	1	1	1	17	18	20	22	24	26
1	1	1	1	1	1	1	1	1	17	18	20	22	24	26	28
1	1	1	1	1	1	1	1	17	18	20	22	24	26	28	30
1	1	1	1	1	1	1	17	18	20	22	24	26	28	30	33
1	1	1	1	1	1	17	18	20	22	24	26	28	30	33	36
1	1	1	1	1	17	18	20	22	24	26	28	30	33	36	39
1	1	1	1	17	18	20	22	24	26	28	30	33	36	39	42
1	1	1	17	18	20	22	24	26	28	30	33	36	39	42	45
1	1	17	18	20	22	24	26	28	30	33	36	39	42	45	49
1	17	18	20	22	24	26	28	30	33	36	39	42	45	49	52

Table 2 16 × 16 Zigzag scanning

1	2	17	33	18	3	4	19	34	49	65	50	35	20	5	6
21	36	51	66	81	97	82	67	52	37	22	7	8	23	38	53
68	83	98	113	129	114	99	84	69	54	39	24	9	10	25	40
55	70	85	100	115	130	145	161	146	131	116	101	86	71	56	41
26	11	12	27	42	57	72	87	102	117	132	147	162	177	193	178
163	148	133	118	103	88	73	58	43	28	13	14	29	44	59	74
89	104	119	134	149	164	179	194	209	225	210	195	180	165	150	135
120	105	90	75	60	45	30	15	16	31	46	61	76	91	106	121
136	151	166	181	196	211	226	241	242	227	212	197	182	167	152	137
122	107	92	77	62	47	32	48	63	78	93	108	123	138	153	148
183	198	213	228	243	244	229	214	199	184	169	154	139	124	109	94
79	64	80	95	110	125	140	155	170	185	200	215	230	245	246	231
216	201	186	171	156	141	126	111	96	112	127	142	157	172	187	202
217	232	247	248	233	218	203	188	173	158	143	128	144	159	174	189
204	219	234	249	250	235	220	205	190	175	160	176	191	206	221	236
251	252	237	222	207	192	208	223	238	253	254	239	224	240	255	256

(v) Generate the cipher text by using substitution matrices

(vi) Apply CICC algorithm or M-MRSLS algorithm to search for an optimal substitution matrix and its corresponding cipher text.

A demonstration is shown below to transform the secret text into cipher text with the help of the following steps:

Step 1: Let's consider a substitution matrix M.

$$M = \begin{array}{c} \\ 0 \\ 1 \\ 2 \\ 3 \end{array} \begin{array}{c} \begin{array}{cccc} 0 & 1 & 2 & 3 \end{array} \\ \begin{bmatrix} 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{array}$$

(a) M

Step 2: Assume the secret message bits are {10110100}. As the value of $d = 2$, divide the number of bits into a group of 2 bits. Now the secret message is

(b) Secret message : $\{\underbrace{10}\ \underbrace{11}\ \underbrace{01}\ \underbrace{00}\}$

Step 3: Calculate the decimal value of each group of a secret message as present in **Step 2 (b).**

(c) Decimal values of secret message : {2 3 1 0}

Step 4: Replace the decimal values of secret message which is considered as the row index of M by its respective column index where the value of M is found as 1.

(d) Substitution result of (c) using (a) :{1 3 0 2}

Step 4: Convert the decimal values as shown in **Step 4 (d)** in to a binary form, ensuing the cipher text of a secret text using the substitution matrix M as mentioned in Step 2 and Step 1, respectively.

(e) Binary value of (d) :{01 11 00 10}

The bits of the cipher message are hidden into the selected DCT coefficients as specified in Table 1. Decryption of a cipher text can also be done by using transpose of M. As we have seen that there are total 24 possible numbers of combinations of M for $d = 2$ and for each combination of M, the stego-image quality may be different. Thus, there is a need to implement the optimization algorithm which could result the optimal substitution matrix and its corresponding cipher text.

(b) Part 2 (Optimal substitution matrix identification using CICC/M-MRSLS)

This part is divided in to two stages. Stage-1 discusses the identification of optimal substitution matrix and its corresponding cipher message by using CICC, whereas Stage-2 describes about M-MRSLS algorithm to identify the optimal substitution matrix and the corresponding cipher message. Illustrations are also shown for CICC and M-MRSLS algorithms in Stage-1 and Stage-2, respectively.

(a) *Stage-1: optimal substitution matrix identification using CICC*

As discussed before about the need of optimization algorithm to identify the optimal substitution matrix, a well-known optimization algorithm CI (for more details refer "Appendix B"), developed by Kulkarni et al. [41], is used along with the concept of cognitive computing (CC). The flowchart of CICC is shown in Fig. 2. Each substitution matrix M is considered as a candidate M_c in the cohort where $c = 1, 2, \ldots, C$. There are total 24 candidates for $k = 2$. In order to understand the behaviour of each candidate and to get more clarity, total 4 candidates are selected randomly out of 24. Here, behaviour of any candidate refers to its quality which further helps to identify the behaviour of a cohort (refer "Appendix B"). PSNR is considered as a fitness function to evaluate the quality of the stego-image. The position of row of substitution matrix M having value 1 decides the quality of a candidate. Thus, each candidate possesses 4 qualities due to the size of M, i.e. 4×4.

We have considered 20 runs for implementation to analyse our results. Since the overall behaviour of the cohort may improve after every iteration, the maximum number of iterations under each run is selected 40 and the threshold limit is set as 0.0001. The steps, flowchart and a sample illustration are shown below to describe the overall procedure of each stage.

Step 1: Initialize the cohort with random number of candidates where any candidate is represented as

$$M_c, \quad c = 1, 2, \ldots, C.$$

Step 2: Evaluate the $PSNR$ value for each candidate M_c with respect to the secret message.

Step 3: Evaluate the selection probability P^{M_c} of any candidate M_c as follows:

$$P^{M_c} = \frac{\text{PSNR}_{M_c}}{\sum_{c=1}^{C} \text{PSNR}_{M_c}} \tag{4}$$

Step 4: Apply a roulette wheel algorithm by any candidate $M_c, c = 1, 2, \ldots, C$ to select and follow a candidate as per the results generated by the algorithm. Here, the term 'follows' refers to the quality adapted by the follower candidate to the followed candidate. Each candidate generates a random integer from within 1 to 2^d. This random integer decides the row value of the follower candidate to be replaced with the corresponding row value of the followed candidate.

Step 5: Apply cognitive computing (CC) approach with CI to obtain better solution. A probabilistic approach is used in CI which does not guarantee to have convergence for a candidate's PSNR value. Thus, the concept of CC is used along with CI which enables each candidate to accept a better quality and improves the overall behaviour of the cohort. According to CICC approach if the candidate's PSNR in the current iteration is better than the PSNR of the previous iteration then accepting that change else retaining earlier.

Step 6: Execute the conditions concurrently as described below to achieve the cohort saturation:

(a) If the maximum number of learning efforts is reached.
(b) If the cohort does not improve its behaviour after certain number of runs, i.e. there is no considerable change or difference identified between the PSNR values of all the candidates in the continuous learning efforts.

The flowchart of CICC is shown below in Fig. 2, and its whole process is explained through a sample illustration.

(b) *Stage-I: a sample illustration of a greyscale image using CICC*

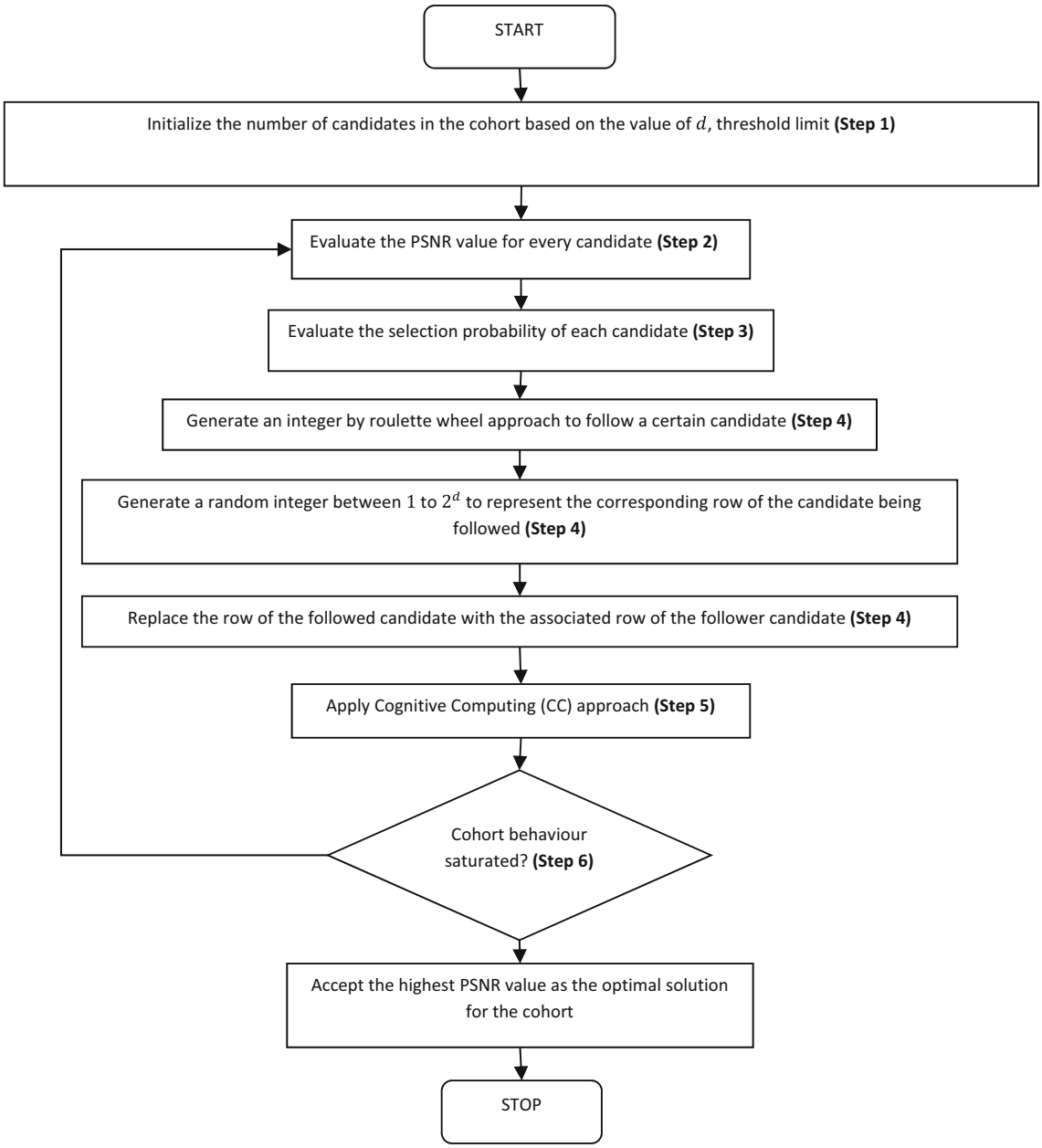


Fig. 2 Cohort intelligence with cognitive computing (CICC) flowchart

Total 4 candidates are selected from the given number of candidates $M_c, c = 1, 2, \ldots, 2^d$ as the value of d is considered 2. A greyscale image of Woman having size 256×256 is considered for this illustration. A single iteration is presented (Fig. 3).

(c) *Stage-2: optimal substitution matrix identification using M-MRSLS*

An original algorithm MRSLS is developed by Kulkarni et al. [42] which is explained in detail in "Appendix C". MRSLS algorithm is modified, implemented and applied on a JPEG greyscale image having 16×16 quantized coefficients for steganography. This algorithm is referred as M-MRSLS. Flowchart and its illustration are shown in Figs. 4 and 5, respectively. In the original algorithm of MRSLS (refer "Appendix C"), a duo-swapping approach is used which enables every solution in the set M_p to generate a neighbouring solution, whereas M-MRSLS algorithm generates a random solution which depends upon the associated solution's behaviour. The number of substitution matrices as described in Part 1 is considered as the number of solutions for a set $\{M_p, p = \{1, 2, 3 \ldots P)\, where\, P = 2^d!\}$. Since d

Number of Candidates

$$\begin{bmatrix}0&0&0&1\\0&0&1&0\\0&1&0&0\\1&0&0&0\end{bmatrix}\begin{bmatrix}1&0&0&0\\0&0&1&0\\0&0&0&1\\0&1&0&0\end{bmatrix}\begin{bmatrix}0&1&0&0\\1&0&0&0\\0&0&1&0\\0&0&0&1\end{bmatrix}\begin{bmatrix}0&1&0&0\\0&0&0&1\\1&0&0&0\\0&0&1&0\end{bmatrix}$$

a) M_1 b) M_2 c) M_3 d) M_4

Evaluate the $PSNR$ value of each candidate

$PSNR_1 = 38.353$, $PSNR_2 = 38.3113$, $PSNR_3 = 38.3719$, $and\ PSNR_4 = 38.3797$

Calculate the total value of $PSNR$

$PSNR_T = 153.4159$

$P_1 = \frac{PSNR_1}{PSNR_T}$, $P_2 = \frac{PSNR_2}{PSNR_T}$, $P_3 = \frac{PSNR_3}{PSNR_T}$, $P_4 = \frac{PSNR_4}{PSNR_T}$

Determine the selection probability of each candidate.

Thus,
$P_1 = \frac{38.353}{153.4159} = 0.24999$, $P_2 = \frac{38.3113}{153.4159} = 0.24972$, $P_3 = \frac{38.3719}{153.4159} = 0.25012$, $P_4 = \frac{38.3797}{153.4159} = 0.25017$

Calculate the candidate's cumulative probability.

$P_{1_{cu}} = 0.24999, P_{2_{cu}} = 0.49971, P_{3_{cu}} = 0.74983, P_{4_{cu}} = 1.000$

By using roulette wheel approach, each candidate generates 4 values between 0 and 1 to follow the convincing candidate.

$\{0.4444, 0.8923, 0.5649, 0.1445\}$
Implies
a) M_1 candidate follows M_2 candidate,
b) M_2 candidate follows M_4 candidate,
c) M_3 candidate follows M_3 candidate, and
d) M_4 Candidate follows M_1 candidate.

Every candidate generates a random integer between 1 and 4 which correspond to the row number of the followed candidate.

$\{3, 4, 1, 2\}$

Substitute the row_3 of the candidate M_1 with row_3 of the candidate M_2 , row_4 of the candidate M_2 with row_4 of the candidate M_4, row_1 of the candidate M_3 with row_1 of the candidate M_3 and row_2 of the candidate M_4 with row_2 of the candidate M_1, respectively to adapt the quality of the followed candidate by the follower candidate

Candidate M_1: $\begin{bmatrix}0&0&0&1\\0&0&1&0\\0&1&0&0\\1&0&0&0\end{bmatrix}$ row_3 of the candidate M_1:
$\{0\ 1\ 0\ 0\}$
Candidate M_2: $\begin{bmatrix}1&0&0&0\\0&0&1&0\\0&0&0&1\\0&1&0&0\end{bmatrix}$ row_3 of the candidate M_2:
$\{0\ 0\ 0\ 1\}$
After replacement, the new candidate $M_1{}'$: $\begin{bmatrix}0&0&0&1\\0&0&1&0\\0&0&0&1\\1&0&0&0\end{bmatrix}$

The new 4 candidates in the cohort

$$\begin{bmatrix}0&0&0&1\\0&0&1&0\\0&0&0&1\\1&0&0&0\end{bmatrix}\begin{bmatrix}1&0&0&0\\0&0&1&0\\0&0&0&1\\0&0&1&0\end{bmatrix}\begin{bmatrix}0&1&0&0\\1&0&0&0\\0&0&1&0\\0&0&0&1\end{bmatrix}\begin{bmatrix}0&1&0&0\\0&0&0&1\\1&0&0&0\\0&0&1&0\end{bmatrix}$$

(a) $M_1{}'$ (b) $M_2{}'$ (c) $M_3{}'$ (d) $M_4{}'$

Calculate the $PSNR$ values of the 4 new candidates

$PSNR'_1 = 38.1833$, $PSNR'_2 = 38.1091$, $PSNR'_3 = 38.2017$, $and\ PSNR'_4 = 38.3937$

Apply CC concept in every PSNR values of all the candidates and compare these values from the current to previous iteration.

1. $M_1{}' < M_1$
2. $M_2{}' < M_2$
3. $M_3{}' < M_3$
4. $M_4{}' > M_4$

Accept the high PSNR values for every candidate and consider it as the current iteration value which enables the cohort to reach to its optimal value.

$39.1833, 38.2091, 38.2417, 38.2037$

Continue the process till the saturation condition is reached as mentioned in **Step 6** of Part 2 under Stage 1.

Fig. 3 Illustration of a greyscale image using CICC

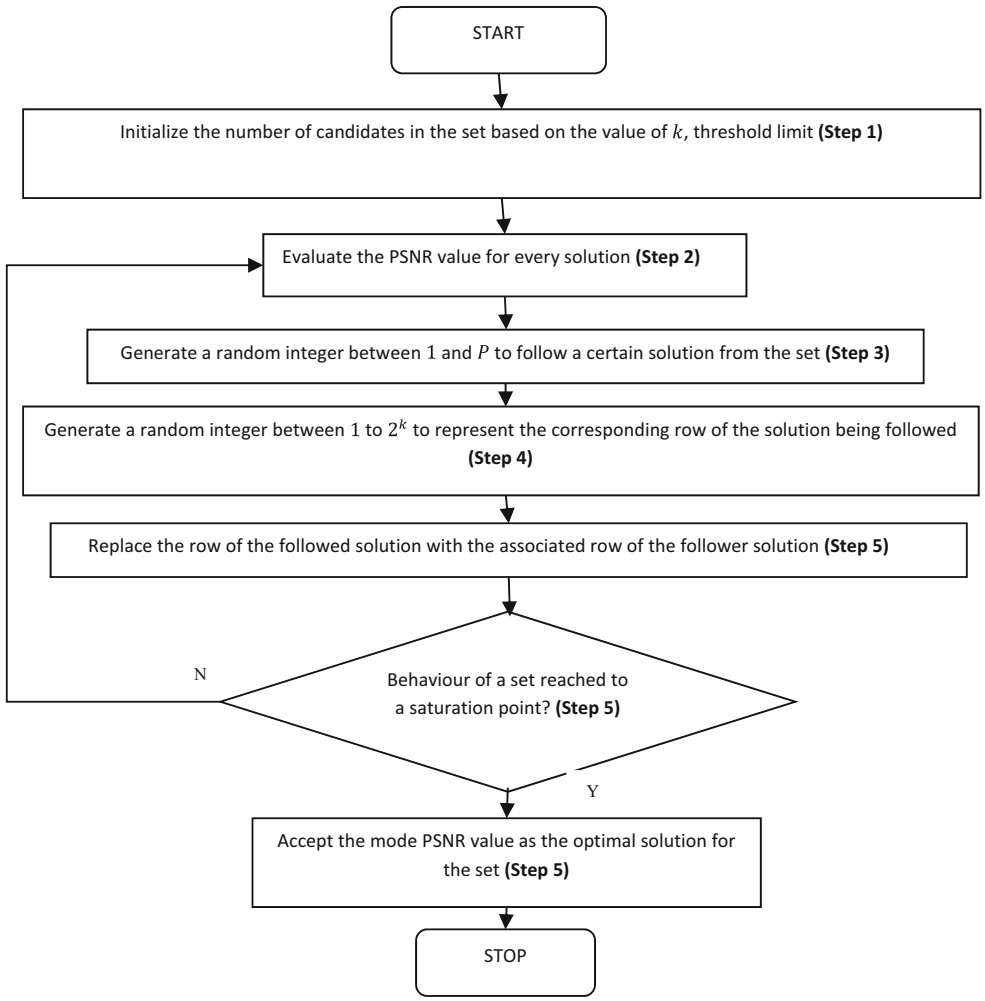


Fig. 4 Modified multi-random start local search (M-MRSLS) flow chart

value is selected as 2, hence there are in total 24 solutions. The total number of selected solutions is 4 in order to get more clarity and easiness. These solutions are picked up randomly. A fitness function, i.e. PSNR as described in Eq. 1, is considered to determine the quality of a stego-image which is dependent upon the behaviour of a set M_p. The complete performance of the set can be determined by the overall performance/behaviour of every solution. The quality of each solution makes its behaviour. In M_p, the row position having value 1 is considered the quality of a solution. The size of each solution which is in the form of substitution matrix is taken 4×4 and implies 4 qualities for every solution in the set. The same number of runs and same iterations under each run are considered for M-MRSLS as discussed in Stage-1 for CICC, i.e. 20 runs and maximum 40 iterations under each run. Steps in detail are discussed as follows.

Step 1: Produce the P random solutions for a set wherein the representation of any solution is $\{M_p, p = \{1, 2, 3 \ldots P)\, where\, P = 2^d!\}$

Step 2: Evaluate the PSNR value for each solution M_p. Equation 1 is used to determine the PSNR value.

Step 3: Every solution M_p, $p = 1, 2, \ldots P$ generates a random integer between 1 and p and follows a solution associated with the generated integer. For example, assume that each solution M_p, $p = 1, 2, \ldots P$ generates a random integer 4 implies that each solution of the set follows the solution M_4.

Step 4: In order to follow a certain solution as described in **Step 3**, each solution generates further a random integer from within 1 to 2^d. This integer represents the corresponding row of the solution being followed. To adapt the qualities of the followed solution, the values of this row number replace the values of the associated row of the follower solution.

Solutions

$$\begin{bmatrix}0&1&0&0\\1&0&0&0\\0&0&1&0\\0&0&0&1\end{bmatrix}\begin{bmatrix}1&0&0&0\\0&0&1&0\\0&1&0&0\\0&0&0&1\end{bmatrix}\begin{bmatrix}0&0&0&1\\0&1&0&0\\1&0&0&0\\0&0&1&0\end{bmatrix}\begin{bmatrix}0&0&1&0\\0&0&0&1\\1&0&0&0\\0&1&0&0\end{bmatrix}$$

a) M_1 b) M_2 c) M_3 d) M_4

Evaluate the $PSNR$ value of each solution

$PSNR_1 = 38.3476$, $PSNR_2 = 38.3710$, $PSNR_3 = 38.3638$, $and\ PSNR_4 = 38.3651$

Select a random value of PSNR

38.3710

Set generates a single random value between 1 and 4 to follow the certain solution.

2

Implies

$\{M_1, M_2, M_3, M_4\}$ follows a solution M_2

Every solution generates a random integer between 1 and 4 which correspond to the row number of the followed solution.

$\{3, 4, 1, 2\}$

Replace the row_3 of the solution M_1 with row_3 of the solution M_2 , row_4 of the solution M_2 with row_4 of the solution M_2, row_1 of the solution M_3 with row_1 of the solution M_2 and row_2 of the solution M_4 with row_2 of the solution M_2 in order to adapt the quality of the followed solution by the follower solution

Solution M_1: $\begin{bmatrix}0&1&0&0\\1&0&0&0\\0&0&1&0\\0&0&0&1\end{bmatrix}$ row_3 of the solution M_1: {0 0 1 0}

Solution M_3: $\begin{bmatrix}0&0&0&1\\0&1&0&0\\1&0&0&0\\0&0&1&0\end{bmatrix}$ row_3 of the solution M_3: {1 0 0 0}

After replacement, the new solution M_1' : $\begin{bmatrix}0&1&0&0\\1&0&0&0\\1&0&0&0\\0&0&0&1\end{bmatrix}$

The new 4 solutions in the set

$$\begin{bmatrix}0&1&0&0\\1&0&0&0\\1&0&0&0\\0&0&0&1\end{bmatrix}\begin{bmatrix}0&0&0&1\\0&1&0&0\\1&0&0&0\\0&0&1&0\end{bmatrix}\begin{bmatrix}0&0&0&1\\0&1&0&0\\1&0&0&0\\0&1&0&0\end{bmatrix}\begin{bmatrix}0&0&1&0\\0&1&0&0\\1&0&0&0\\0&1&0&0\end{bmatrix}$$

(a) M_1' (b) M_2' (c) M_3' (d) M_4'

Calculate the $PSNR$ values of the 4 new solutions

$PSNR'_1 = 38.3650$, $PSNR'_2 = 38.1091$, $PSNR'_3 = 38.3697$, $and\ PSNR'_4 = 38.3669$

Select a random value of PSNR

38.3697

Continue the process till the saturation condition is reached as mentioned in **Step 6** of Stage 2.

Selected random PSNR values from each iteration

38.3476, 38.3710, 38.3710, 38.3710, 38.3638, 38.3345, 38.3638, 38.365
38.3690, 38.3690, 38.3690, 38.3707, 38.3707, 38.3621, 38.3621, 38.36
38.3650, 38.3650, 38.3697, 38.3697, 38.3697, 38.3669, 38.3705, 38.367
38.3609, 38.3536, 38.3608, 38.3671, 38.3621, 38.3671, 38.3530, 38.3530
38.3536, 38.3593, 38.3593, 38.3593, 38.3719, 38.3593, 38.3593, 38.3719

Accept the mode PSNR value

38.3593

Fig. 5 Illustration a greyscale image using M-MRSLS

Step 5: The goal of this algorithm is to determine an optimized fitness solution amongst the generated feasible solutions. Since there is a great randomization involved in M-MRSLS algorithm, an idea of statistical mode is incorporated in the algorithm. The mode is the number that occurs most frequently within a set of numbers. Mode helps identify the most common or frequent occurrence of a characteristic. The saturation point is considered as the maximum number of attempts, i.e. 40.

Step 6: All the steps are repeated until the saturation point is reached.

Flowchart and illustration as shown in Figs. 4 and 5 bring more clarity for this algorithm.

After the implementation of either Stage-1 or Stage-2, we are able to achieve an optimization matrix and thus a transformed secret message which is hidden into the frequency components of every 16×16 block of greyscale image. The embedding procedure of the transformed secret message in to the image blocks is described in Phase 3 following the illustration of M-MRSLS algorithm.

(d) *Stage-2: a sample illustration of a greyscale image using M-MRSLS*
The total number of solutions in a set $M_p, p = 1, 2, \ldots, 2^d$ for $d = 2$ is 4. A single iteration is shown considering a greyscale image Woman having size 256×256.

(iii) *Phase 3 (Embedding procedure of transformed secret text using* 16×16 *quantization table)*
As described earlier, the generation of an optimal substitution matrix by applying either CICC algorithm or M-MRSLS algorithm, which in turn generates the transformed secret text, is used to hide into the low–middle frequency components of the quantized DCT coefficients. Any image is firstly divided into blocks, and once the DCT is applied to each block, DCT coefficients are generated (refer "Appendix A"). Each block of an image is fragmented in to three frequency bands, i.e. low, middle and high. The top most left value of any image block is considered as the DC coefficient, and the remaining coefficients are referred to as AC coefficients. AC coefficients are used to hide the secret text as there could be a visual distortion of the image quality if the DC coefficients are altered [49]. Low–middle frequency zones of the coefficients are considered to be the safer one since the higher frequency zone is easily targeted by an attacker [50], and there is a probable chance of revealing the secret information by applying compression techniques and noise attacks. As described in "Appendix C" (the overall procedure of JPEG), hiding of transformed secret text is done after quantization. As described in the previous section, quantization table plays a very major role for quantization as we would be able to select the more number of quantized coefficients as per the size and the values of the quantization table. The embedding procedure of transformed secret text is shown in Fig. 6.

(iv) *Phase 4 (generation of stego-image)*

This phase includes five subphases RLE, DPCM, EC, HC and FB which has been applied to the embedded output (refer Fig. 6) to generate the stego-image. These subphases are explained in detail in "Appendix C". RLE and DPCM are applied only on AC components and DC components of the embedded output, respectively. The produced output from the previous phase is used for EC in which DC and AC components are encoded. Then, HC and FB are applied to these generated encoded values to produce a stego-image. All the applied processes on embedded output under this phase are used for encoding purpose. The flowchart for this phase is shown through Fig. 7.

2.1.2 Retrieval Algorithm

This algorithm describes the extraction procedure of secret text and cover image at the receiver end. The reverse process of embedding algorithm is considered in this section which is represented in Fig. 8.

The whole procedure is divided into five steps as follows.

Step 1: The overall encoding procedure by different processes is shown in Fig. 7. The reverse procedure of

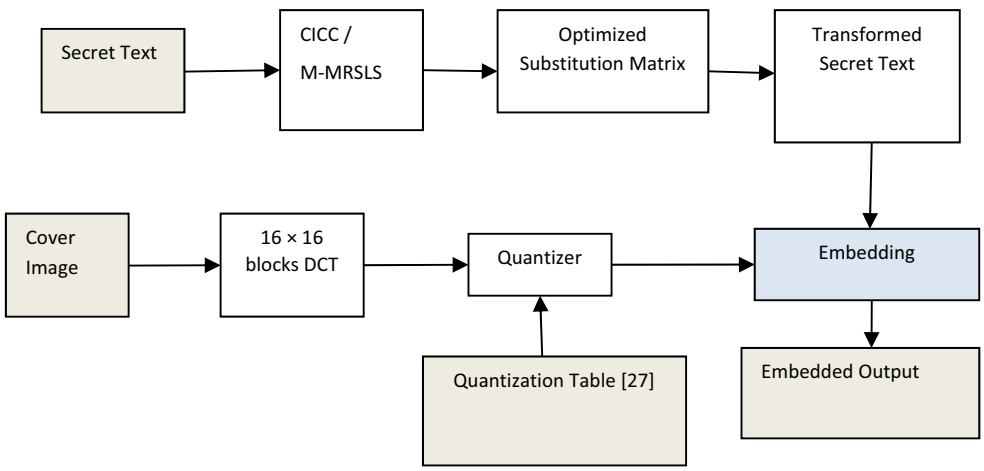


Fig. 6 Embedding procedure of CICC/M-MRSLS

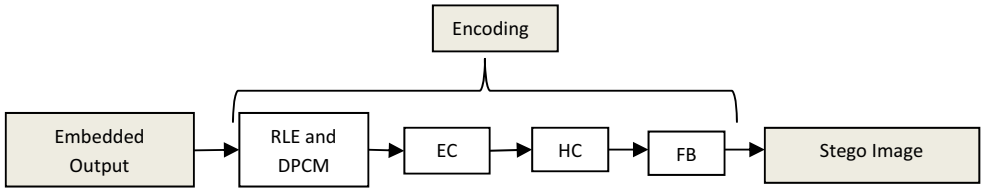


Fig. 7 Generation of stego-image

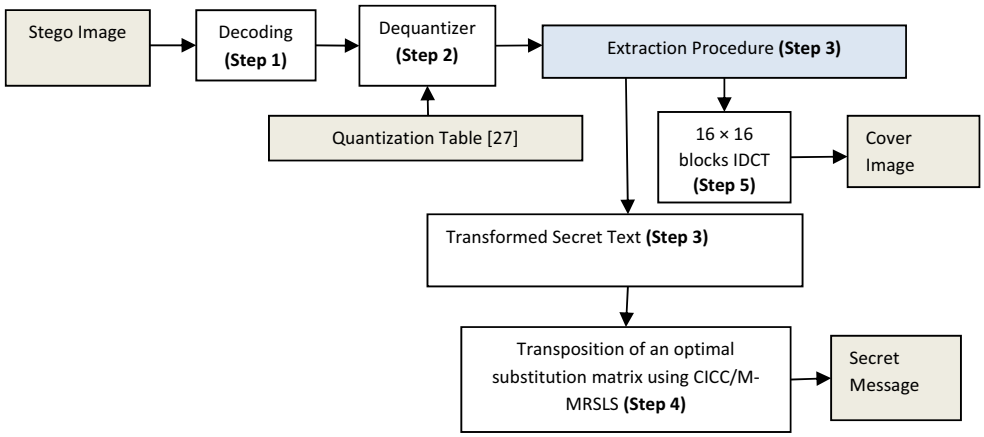


Fig. 8 Extraction procedure

encoding referred to as decoding is applied to stego-image which helps to retrieve the decoded blocks of stego-image of size 16 × 16.

Step 2: Dequantization is applied into the produced output from **Step 1**. The same quantization table as proposed by Almohammad et al. [27] is used for dequantization.

Step 3: The output received after dequantization is used to extract the transformed secret bits. The order used for extraction is same as the embedding order of transformed secret text. The embedding order table is discussed and presented in Table 2.

Step 4: Transposition of an optimal substitution matrix is done either using CICC or M-MRSLS. In order to get the original secret text, the transformed bits of secret text (output of Step 3) are passed to the transpose of the optimal substitution matrix.

Step 5: Inverse DCT (refer "Appendix C") is applied to the 16 × 16 blocks of dequantized coefficients to extract the block of cover image. The same procedure is repeated until all the blocks of the cover image are retrieved.

3 Results and Discussion

This section shows the results of the proposed CICC and MRSLS. All the algorithms were coded in MatlabR12 and executed on a personal computer with Intel i5 processor having 2 GB RAM Under the Windows 10 operating system. In order to implement the proposed methods, total six greyscale images are considered for testing purpose. These six images are all 8 bit images having 256 grey levels and the size of these images are taken as 256 × 256. These images considered for experimentation are Lena, Baboon, Boat, Gold hill, Girl and Women. The analysis of these results is done in terms of evaluation parameters such as PSNR, computational time, secret text capacity and number of function evaluations. Figure 9 presents the computed PSNR with respect to each image for both the proposed methods, i.e. CICC and M-MRSLS in which the cover image as well as stego-image is shown.

Steganography implemented on various images

Also, a comparative analysis is carried out between CICC (16 × 16), M-MRSLS (16 × 16), CI (16 × 16), a non-optimal substitution method of size 16 × 16, a non-optimal substitution method of size 8 × 8 and JQTM. The non-optimal substitution method is just the naming convention given by Li et al. [13] as there is no optimization methodology involved to transform the secret bits. If the image is divided into 16 × 16 blocks and the 16 × 16 quantization table is used, then this method is named as non-optimal substitution method 16 × 16. On the other side, if the image is divided into 8 × 8 blocks and the quantization table of size 8 × 8 is used as proposed for JQTM, the method is named as non-optimal substitution method 8 × 8. All the methods are implemented and tested on the same images, and its results as PSNR and standard

- Baboon Image 256×256 *Pixels*

(a) PSNR = 27.6428 (CICC) (a) PSNR = 27.6438 (M-MRSLS)

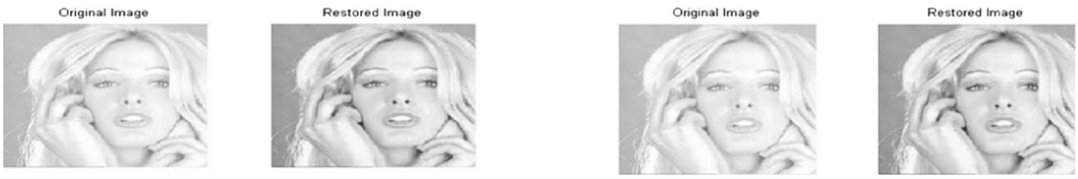
- Lena Image 256×256 Pixels

(a) PSNR=34.2691 (CICC) (a) PSNR=34.2691 (M-MRSLS)

- Boat Image 256×256 *Pixels*

(b) PSNR = 33.7763 (CICC) (b) PSNR = 33.7763(M-MRSLS)

- Woman Image 256×256 *Pixels*



(c) PSNR= 38.3797 (CICC) (c) PSNR = 38.3797 (M-MRSLS)

- Gold Hill Image 256×256 *Pixels*

(d) PSNR = 34.2106 (CICC) (d) PSNR=34.211 (M-MRSLS)

- Girl Image 256×256 *Pixels*

(e) PSNR = 28.0225 (CICC) (e) PSNR = 28.0231 (M-MRSLS)

Fig. 9 Original and stego-images of the CICC and M-MRSLS method

Table 3 Image quality [PSNR in decibel (DB)]

Method	Images					
	Lena	Baboon	Woman	Boat	Gold hill	Girl
CI (16 × 16)	38.2017	29.9342	39.126	38.5897	41.5983	30.1151
CICC (16 × 16)	34.2691	27.6428	38.3797	33.7763	34.2106	28.0225
M-MRSLS (16 × 16)	34.2691	27.6438	38.3797	33.7763	34.211	28.0231
Non-optimal substitution method (16 × 16)	37.7849	29.1646	38.3546	33.7541	41.041	28.0174
Non-optimal substitution method (8 × 8)	37.784	29.1646	42.1203	38.5884	41.04	29.5103

Table 4 Standard deviation of PSNR

Method	Images					
	Lena	Baboon	Woman	Boat	Gold hill	Girl
CICC (16 × 16)	0.01599	0.002196	0.003442	0.005093	0.010823	0.00181
M-MRSLS (16 × 16)	0.02550	0.001706	0.02533	0.014351	0.010914	0.00554

Table 5 Comparison of function evaluations (FE)

Method	Images					
	Lena	Baboon	Woman	Boat	Gold hill	Girl
CICC (16 × 16)	27	32	31	32	24	24
M-MRSLS (16 × 16)	25	36	22	39	35	18

Table 6 Comparison of computational time (time in seconds)

Method	Images					
	Lena	Baboon	Woman	Boat	Gold hill	Girl
CICC (16 × 16)	71.7879	144.6961	63.2178	115.796	108.9902	59.8623
M-MRSLS (16 × 16)	81.6882	121.876	92.5372	102.5778	77.5181	84.2666
Non-optimal substitution (16 × 16)	7.0765	8.8272	3.9194	4.3911	7.7045	3.442
CI (16 × 16)	551.0261	631.758	550.9619	641.1347	633.831	644.2497

Table 7 Comparison: best, median and worst image quality (PSNR in DB)

Method	Images								
	Lena			Baboon			Woman		
	Best	Median	Worst	Best	Median	Worst	Best	Median	Worst
CICC (16 × 16)	34.2691	34.2585	34.2083	27.6438	27.6398	27.6371	38.3797	38.3719	38.3708
M-MRSLS (16 × 16)	34.2691	34.2543	34.2083	27.6428	27.6398	27.6371	38.3797	38.35335	38.3113

Table 8 Comparison: best, median and worst image quality (PSNR in DB)

Method	Images								
	Boat			Gold hill			Girl		
	Best	Median	Worst	Best	Median	Worst	Best	Median	Worst
CICC (16 × 16)	33.7763	33.7652	33.7652	34.2106	34.1877	34.1754	28.0231	28.0186	28.018
M-MRSLS (16 × 16)	33.7763	33.75595	33.7361	34.2106	34.1877	34.1754	28.0225	28.0182	28.0087

Table 9 Comparison: computational time for best, median and worst case

Method	Images								
	Lena			Baboon			Woman		
	Best	Median	Worst	Best	Median	Worst	Best	Median	Worst
CICC (16 × 16)	207.4928	369.5112	484.7685	193.0027	454.4547	716.9827	252.8656	479.92825	533.0646
M-MRSLS (16 × 16)	78.9396	154.6788	316.5477	136.44023	289.1447015	521.5855	46.109	237.93195	389.6469

Table 10 Comparison: computational time for best, median and worst case

Method	Images								
	Boat			Gold hill			Girl		
	Best	Median	Worst	Best	Median	Worst	Best	Median	Worst
CICC (16 × 16)	222.3972	373.12095	527.4827	273.6345	479.6342	525.104	272.3617	489.69175	564.7522
M-MRSLS (16 × 16)	50.5418	204.15725	439.7497	49.8291	170.97835	309.8737	221.9242	298.2659	512.2002

Table 11 Comparison of capacity (bits)

Method	Capacity (bits)				
	Selected DCT coefficients for hiding	Number of bits to be hidden per DCT coefficient	Hiding capacity per block	Total blocks for 256 × 256 image	Total hiding capacity
CICC (16 × 16)	121	2	242	256	61952
M-MRSLS (16 × 16)	121	2	242	256	61952
CI (16 × 16)	121	2	242	256	61952
Non-optimal substitution method (8 × 8)	26	2	52	1024	53248

deviation of PSNR are shown in Tables 3 and 4. We find that the image qualities of the proposed methods are comparable with the other mentioned methods. We observed that the image quality found for all the six images by CI (16 × 16) is improved than the proposed methods, i.e. CICC (16 × 16) and M-MRSLS (16 × 16).

Total 20 runs are considered for implementing the proposed methods. Each run is having 40 iterations. Function evaluations are calculated for the proposed methods and presented in Table 5. Table 6 presents the elapsed/computational time for the proposed methods, CI (16 × 16) and the non-optimal substitution method (16 × 16). Since there is no optimality included in non-optimal substitution method (16 × 16) and is executed only one time, the evaluated computational time is less than the proposed methods. Though the image quality for CI (16 × 16) is improved than the proposed methods, we found that this method does not converge the solution and increases the computational cost as shown in Table 6.

Best, median and worst case PSNR values are evaluated for all the images of the proposed methods and shown in Tables 7 and 8. Tables 9 and 10 present the computational time for the proposed methods for best, median and the worst case. This computational time is evaluated for all the test images. Capacity of embedded secret text in terms of number of bits is also calculated for the proposed methods, CI (16 × 16), and is compared with non-optimal substitution method (8 × 8). The comparative analysis for the capacity is shown in Table 11. Since the proposed methods and CI (16 × 16) use the (16 × 16) quantization table, there are total 121 DCT coefficients selected in each block and 2 bits are used per coefficient to hide the secret text. Thus, the total number of secret bits which can be embedded in each block is $121 \times 2 = 242$. The size of the cover image is considered 256 × 256. Therefore, the total numbers of image blocks are $(256 \times 256) \div (16 \times 16) = 256$ which enables to calculate the total embedding capacity of secret text in the image, i.e. $256 \times 242 = 61952\, bits$. However, if we compare this analysis with non-optimal substitution method (8 × 8) which is implemented on the same images and the (8 × 8), quantization table is used as proposed in JQTM where in total 26 DCT coefficients are selected and 2 bits are used per coefficient to hide the secret bits. Thus, total $26 \times 2 = 52$ coefficients are selected in each block which further calculates the total number of bits/capacity for all the blocks/entire image, i.e. $Total\ number\ of\ blocks \times 52$. In this case, the

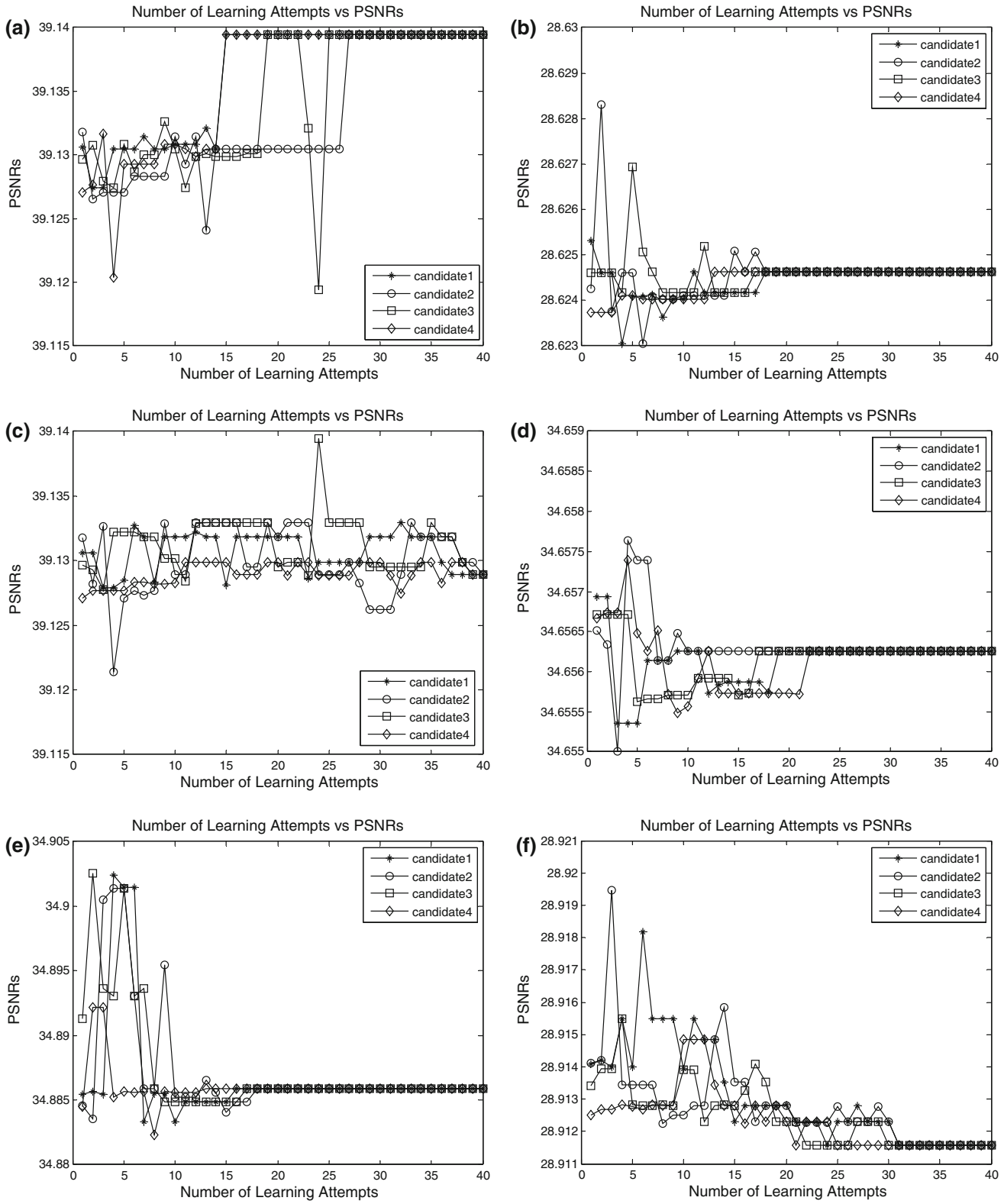


Fig. 10 CICC (16 × 16). **a** Lena. **b** Baboon. **c** Woman. **d** Boat. **e** Gold hill. **f** Girl

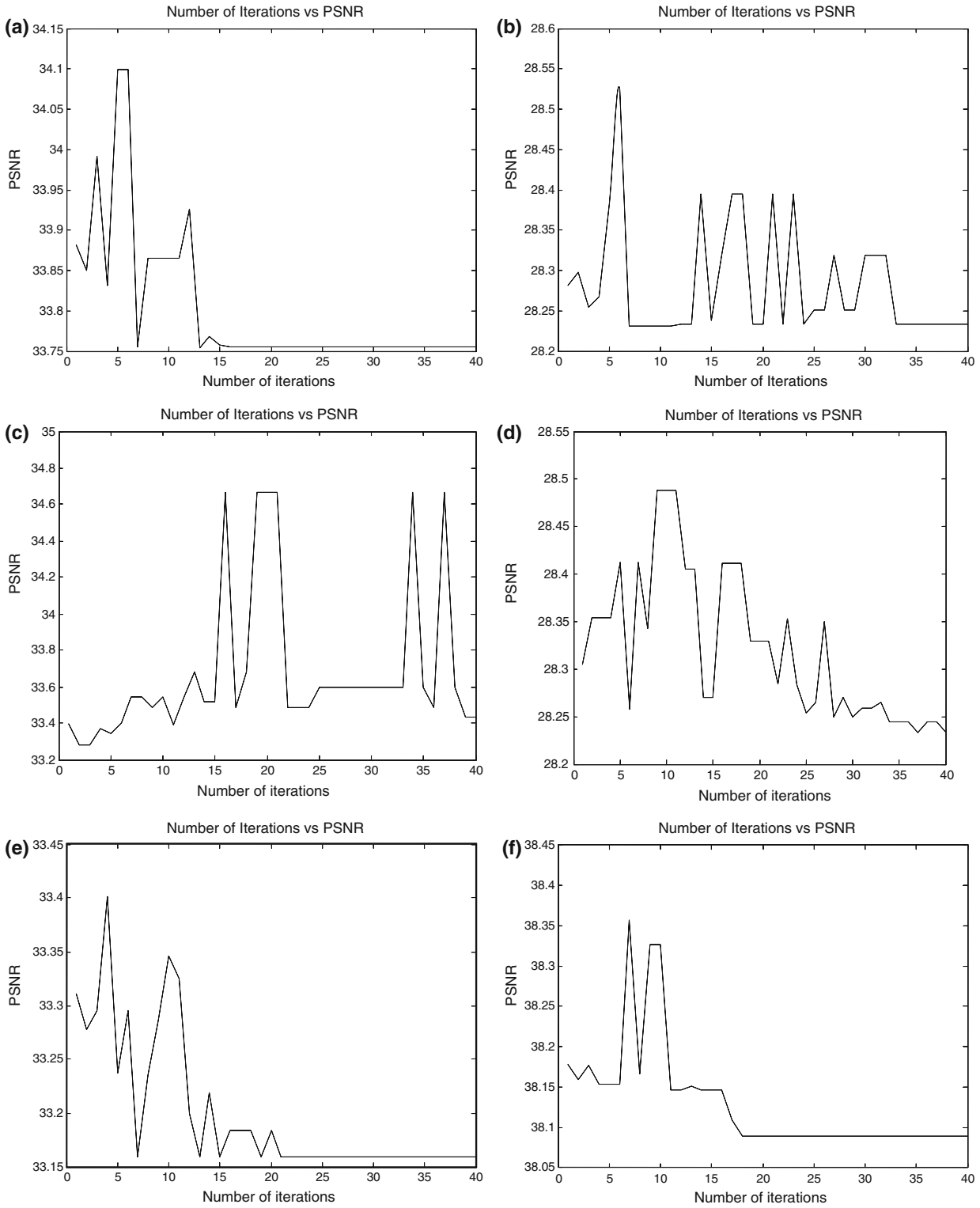


Fig. 11 M-MRSLS (16 × 16). **a** Lena. **b** Baboon. **c** Boat. **d** Girl. **e** Gold hill. **f** Woman

total number of blocks is $(256 \times 256) \div (8 \times 8) = 1024$; hence, the calculated capacity is $1024 \times 52 = 53248$ which less is than the capacity of the proposed methods. The increased percentage of capacity improvement is $((61952 - 53248) \div 53248) \times 100 = 16.35\%$, and so we can conclude that the secret text embedding capacity increases of the proposed methods with the comparable quality of the other methods.

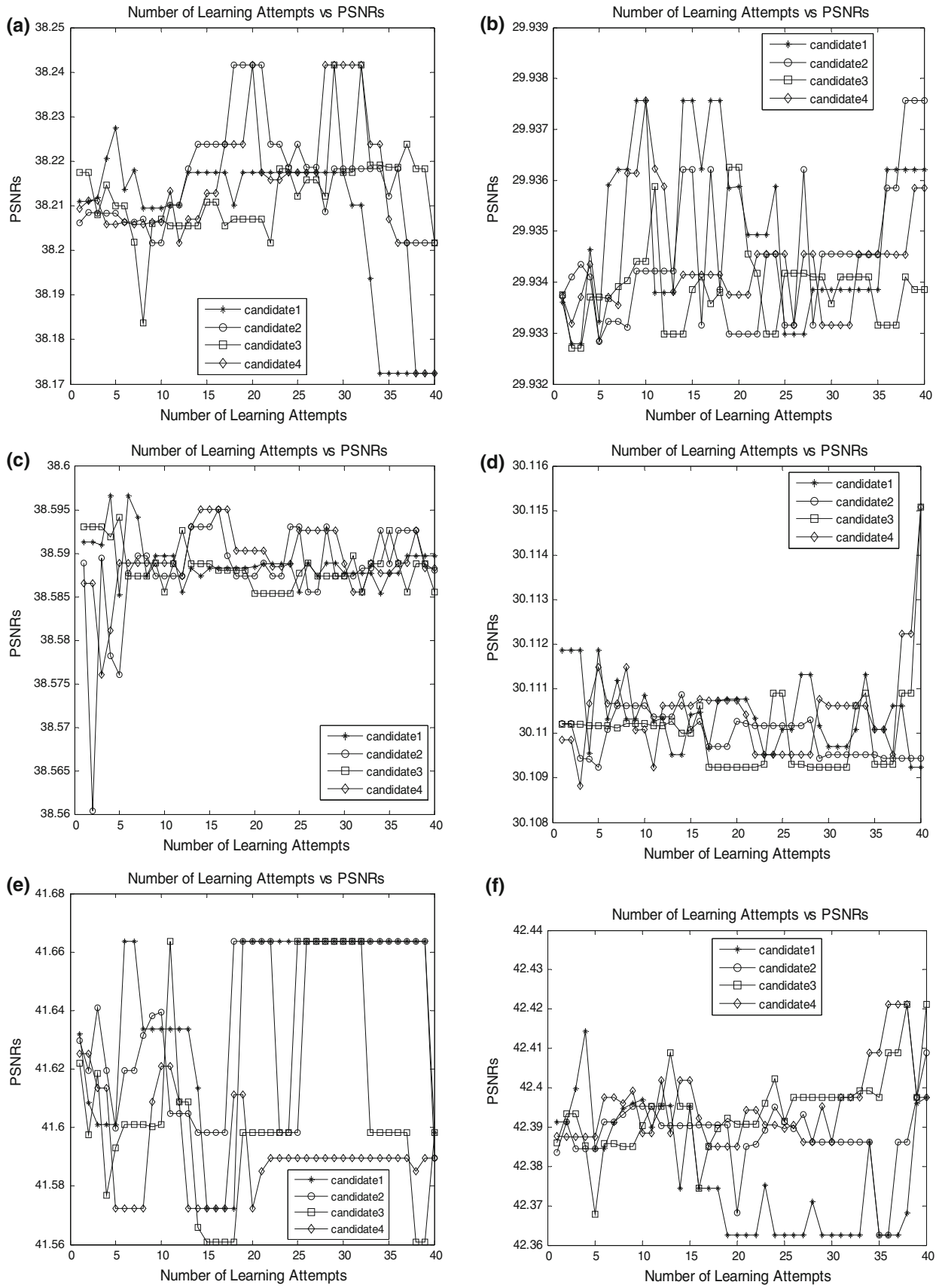


Fig. 12 CI (16 × 16). **a** Lena. **b** Baboon. **c** Boat. **d** Girl. **e** Gold hill. **f** Woman

Snapshots of the proposed methods and CI (16 × 16) are also presented to show the converged PSNR value, number of learning attempts and elapsed value for each sample image. Figures 10 and 11 show the graph and the converged PSNR value tested for all the images for CICC and M-MRSLS, respectively. A single run is showcased in these tables. Four different representations of candidates are shown in the graph of Fig. 10 for Lena image. These representations are 'Asterisk', 'Circle', 'Square' and 'Diamond' for the Canidate 1, Candidate 2, Candidate 3 and Candidate 4, respectively. The dimensions of the graph in x and y directions are represented as the number of learning attempts and PSNR, respectively. Also, if we analyse the other method, i.e. M-MRSLS as shown in Fig. 11, a mode value is calculated, wherein for each iteration a random PSNR is selected. Thus, there will be 40 PSNRs for all the iterations under each run on which the mode value is considered. We refer the Lena image again from Fig. 11. Since a single random value of PSNR is selected for every iteration, the representation of the candidate is shown in one shape, i.e. 'Circle' in the graph. Figure 12 is a graph representation for CI (16 × 16) method, wherein each candidate is depicted as similar as represented for CICC (16 × 16); however, we could observe that the PSNR values are not being converged for all the six images. Thus, this method picked up the latest PSNR value computed at iteration number 40.

4 Conclusions and Future Directions

In this paper, we have proposed two secure JPEG steganography algorithms based on the CICC and M-MRSLS optimization technique and have applied to six greyscale test images. Also, an effort has been put to explore the said methods using 16 × 16 quantization table. Since we have used JPEG steganography where the DCT calculation is required for 8 × 8 pixels block and if the number of blocks gets increased to 16 × 16, this calculation may increase the running/computational cost and so the complexity [50]. However, as per the results discussed in Sect. 3, the proposed methods, which are benefited from the optimization algorithms and the quantization table, achieve a good balance between the security, image quality and secret text capacity. Perhaps, the results of the proposed methods are comparable or sometimes even improved than CI (16 × 16), JQTM, non-optimal substitution method (16 × 16) and non-optimal substitution method (8 × 8). The two objectives (i) to increase PSNR value and (ii) to increase the embedded text capacity are considered in our proposed work, and these objectives are conflicted with each other because enhancement of embedded text capacity force to decrease the PSNR, i.e. image quality. Thus, in our future work we could use and apply a multi-objective function to solve the same problem. Also, we could focus on to improve robustness and efficiency in terms of maintaining a trade-off between optimality and speed of the proposed algorithms. Furthermore, in the line of very recent developments different variations of CI methodology [51] could be tested solving the steganography capacity improvement problem. In the near future, the CI algorithm could be hybridized with the approaches such as natural flocking [52], Cuckoo Search algorithm [53] as well as Ideology Algorithm [54] for solving digital steganography problems [55].

Acknowledgements Authors would like to thank the anonymous receivers. Their comments helped in much improvement in the quality of the manuscript.

Appendix A: Joint Photographic Expert Group (JPEG)

JPEG is a lossy compression technique where the exact replica of the original data is not possible. This compression technique is based on DCT and applied to any image either greyscale or colour image. The following operations are required for JPEG compression (Fig. 13).

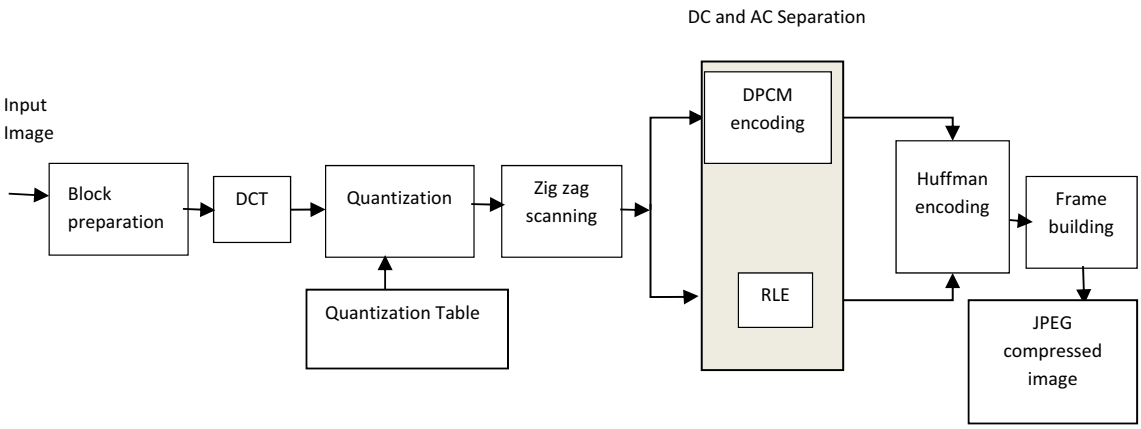


Fig. 13 Block diagram of encoder of JPEG compression

1. Block preparation.
2. Discrete cosine transform (DCT).
3. Quantization.
4. Zigzag scanning.
5. Digital pulse code modulation (DPCM) encoding.
6. Run length encoding (RLE).
7. Huffman encoding technique.
8. Frame building.

1. Block Preparation:
   In this step, a greyscale image is divided in to 8×8 pixel blocks to get the optimal results. If in case image dimensions are not a multiple of 8, extra pixels can be padded to the right part of the image to make it a complete 8×8 pixel blocks. Thus, there will be 64 pixels in each block for processing (Fig. 14).

2. DCT:
   Each value in the block represents the intensity value. The idea of applying DCT to each block is to transform the block into frequency domain. Due to this, each intensity value will be represented as amplitude of a unique cosine function, which enables to keep the cosine functions separated and helps to remove the information having smallest involvement to the image. DCT can be applied on one-dimensional data. Since the image is divided in to two dimensions, DCT is used first in the x direction and then in the y direction to process the image data. The two-dimensional DCT equation is given as under in Eq. 5 where $C(x) = \frac{1}{\sqrt{2}}$ if $x = 0$ and $C(x) = 1$ for other cases. $f(x, y)$ is the original image value at (x, y) position, and $F(u, v)$ is the new calculated value in the frequency domain.

$$F(u,v) = \frac{1}{4}.C(u)\,C(v) \left[\sum_{x=0}^{7}\sum_{y=0}^{7} f(x,y)\cos\frac{(2x+1).u\pi}{16}\cos\frac{(2y+1).v\pi}{16}\right] \quad (5)$$

   This step takes most of the time to compute DCT coefficients (Fig. 15).
3. Quantization:
   This step helps to remove the least important part of the image by dividing each DCT coefficients from two-dimensional matrixes with its corresponding prearranged integer value which in turn produces an integer/float value depending upon the value of the constant. The rounded off operation is applied to the results after division. The predetermined table used for this step is referred to as quantization table, and if the values of this table increase, there is more chance to introduce quantization error and to make highest frequency coefficients zero; however,

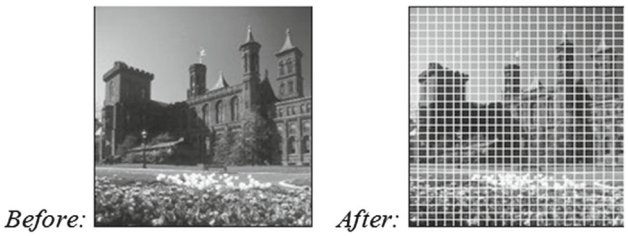


Fig. 14 Block preparation

-412	-30	-61	27	56	-20	-2	0
4	-22	-61	10	13	-7	-9	5
-47	7	77	-25	-29	10	5	-6
-49	12	34	-15	-10	6	2	2
12	-7	-13	-4	-2	2	-3	3
-8	3	2	-6	-2	1	4	2
-1	0	0	-2	-1	-3	4	-1
0	0	-1	-4	-1	0	1	2

÷

16	11	10	16	24	40	51	61
12	12	14	19	26	58	60	55
14	13	16	24	40	57	69	56
14	17	22	29	51	87	80	62
18	22	37	56	68	109	103	77
24	35	55	64	81	104	113	92
49	64	78	87	103	121	120	101
72	92	95	98	112	100	103	99

Round Value

-26	-3	-6	2	2	-1	0	0
0	-2	-4	1	1	0	0	0
-3	1	5	-1	-1	0	0	0
-4	1	2	-1	0	0	0	0
1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Fig. 15 Quantized matrix

Table 12 Standard JPEG quantization matrix

16	11	10	16	24	40	51	61
12	12	14	19	26	58	60	55
14	13	16	24	40	57	69	56
14	17	22	29	51	87	80	62
18	22	37	56	68	109	103	77
24	35	55	64	81	104	113	92
49	64	78	87	103	121	120	101
72	92	95	98	112	100	103	99

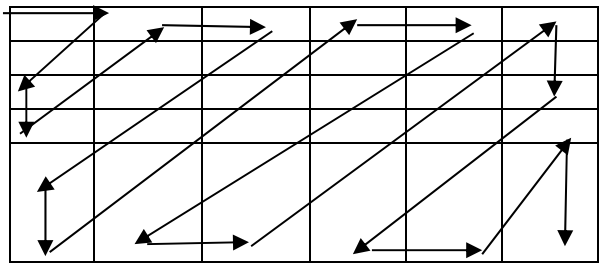


Fig. 16 Zigzag scanning pattern

this has some other effects also. JPEG standard quantization table is shown in Table 12.

4. Zigzag Scanning:
   This step is a part of entropy coding. The main purpose of zigzag scanning is to cluster all the zero's and no zero's value from 64 coefficients in each block. Since there are many zero's available, the finest method is to collect all the zeroes together for compression. Thus, the zigzag scanning is used. This scanning converts the 8×8 matrix into one-dimensional array, referred to as vector and the dimension of this vector would be 1×64. Zigzag scanning pattern is shown in Fig. 16 where all the AC values are encoded in each block.

5. DPCM Encoding:
   DPCM encoding is used to encode all the DC values for each block. The DC coefficient is the first and the mean value of all the 64 values of that block. The value of the DC coefficient is quiet large and close to the DC coefficient of the previous block. Thus, DPCM encodes the difference between the actual sample value and its predicted value after quantization. The predicted value is based on previous samples.
6. Run Length Encoding (RLE):
   RLE is applied to all the AC coefficients. This encoding is very useful on data that has many runs of the related sequence. This technique enables the data stream to save the data as a single data value. A special symbol is used to denote the meaning of the representation. For example, if the data string is *mmmmkkk* and if we apply RLE to the data string, then the encoded string will be in the form of $@m4k3$, wherein the character which immediately follows the special symbol @ is a character and the next immediate symbol is the frequency of the character.
7. Huffman Encoding:
   Huffman encoding is a technique that can be applied on symbols which can be in the form of bytes, DCT coefficients, etc. This method encodes the symbol in a variable length code where the frequently occurring symbols are encoded with less number of bits and rarely used symbols are encoded with more number of bits which facilitate to compress the input file. This technique can be applied to differentially encoded DC coefficients as well as AC coefficients of a block.
8. Frame Building:
   The frame building process is the last step of JPEG compression which accumulates all the data. Checking and verification of an error is also done in the last stage before sending to the output.

Appendix B: Cohort Intelligence (CI)

CI is an evolving optimization algorithm developed by Kulkarni et al. [40]. This algorithm is inspired from natural and social tendency of learning with each other. CI is based on artificial intelligence in which group of candidates form a cohort and competes with each other to achieve a shared goal. In order to solve a combinatorial problem, we formulate this methodology mathematically. Consider total number of candidates as C, wherein an individual candidate is referred as c and its value lies from $\{1, 2, \ldots C\}$. Each candidate observes the behaviour of any other candidate as well as itself and tries to follow either its own behaviour or the behaviour of the other. Here, the term behaviour refers to an objective function. So let us assume the objective function is $Minimize\ f(x) = f(x_1, x_2, x_3, \ldots, x_N)$. The behaviour of any candidate depends upon its existing qualities; the quality of a candidate is represented mathematically as $x^c = \left(x_1^c, x_2^c, x_3^c, \ldots, x_N^c\right)$. In order to improve the behaviour of a cohort, it is required that the other candidates follow a better behaviour. Since CI is a probabilistic approach, thus, to follow a better behaviour is totally based on its probability. Each candidate puts its efforts to make a better or improved cohort behaviour until the saturation condition reaches. The following steps are required for this algorithm.

Step 1: Number of candidates C is initialized; a candidate is represented as $c = 1, 2, \ldots, C$, learning attempt counter starts with $n = 1$, and the convergence fac-

tor is assumed as r. The value of C and r can be chosen based on the given problem.

Step 2: In order to select the behaviour of other candidates, the probability of each candidate is calculated. Individual candidate behaviour is represented mathematically as $f^*(x_c)$.

$$p^c = \frac{\frac{1}{f^*(x^c)}}{\sum_{c=1}^{C} \frac{1}{f^*(x^c)}} \tag{6}$$

Step 3: A random number $r \in (0, 1)$ is generated, and roulette wheel selection approach is used to enable each candidate to select behaviour of other candidates from within the existing choices.

Step 4: Check whether the saturation condition is reached for consecutive substantial number of learning attempts or whether the maximum number of attempts exceeds.

Step 5: In order to evaluate the saturation condition, verify the difference between the individual behaviours. Assume l is a learning attempt and $l_{\max}$ is the max number of learning attempts, and then the difference of the individual behaviour should not be exceeded by a value ε. The equations are given as under:

$$\max\left(f\left(x^c\right)^l\right) - \max\left(f\left(x^c\right)^{l-1}\right) \leq \varepsilon, \text{ and}$$
$$\min\left(f\left(x^c\right)^l\right) - \min\left(f\left(x^c\right)^{l-1}\right) \leq \varepsilon, \text{ and}$$
$$\max\left(f\left(x^c\right)^l\right) - \min\left(f\left(x^c\right)^l\right) \leq \varepsilon.$$

Step 6: Accept the candidate's behaviour as the final solution, if either of these conditions mentioned in step 4 is valid, else continue to step 2.

Appendix C: Multi-Random Start Local Search (MRSLS)

MRSLS is developed by Kulkarni et al. [42] and is a methodology which follows a duo-swapping approach. In this approach, the neighbouring solutions interchange itself. The steps for this algorithm are as under:

Step 1: Number of solutions are initialized and interpreted $p = \{1, 2, 3 \ldots P)$ where in P is considered the total number of solutions in a set.

Step 2: Accept a random generated solution as a starting solution from the set.

Step 3: Apply duo-swapping approach to interchange the solutions fall into the adjacent positions.

Step 4: If the new solution is found better than the previous one, it gets updated, else remains the previous one.

Step 5: Continue with all the steps unless the stopping criterion is met.

A different variety of problems has been considered to test and validate this method which proved the method is comparable with CI.

References

1. Coron, J.S.: What is cryptography? IEEE Secur. Priv. **4**(1), 70–73 (2006)
2. Mishra, R.; Bhanodiya, P.: A review on steganography and cryptography. Int. Conf. Adv. Comput. Eng. Appl. (2015). doi:10.1109/ICACEA.2015.7164679
3. Cheddad, A.; Condell, J.; Curran, K.; Kevitt, P.: Digital image steganography: survey and analysis of current methods. Signal Process. **90**, 727–752 (2010)
4. Rabah, K.: Steganography-the art of hiding data. Inf. Technol. J. **3**(3), 245–269 (2004). ISSN 1682-6027
5. Chandramouli, R.; Kharrazi, M.; Memon, N.: Image steganography and steganalysis: concepts and practice. LNCS **2939**, 35–49 (2004)
6. Westfeld, A.; Pfitzmann, A.: Attacks on steganographic systems breaking the steganographic utilities EzStego, Jsteg, Steganos, and S-Tools-and some lessons learned. In: Information Hiding, Third International Workshop, IH'99, Dresden, Germany, 1 Lecture Notes in Computer Science, pp. 61–76. Springer (1999)
7. Chan, C.K.; Cheng, L.M.: Hiding data in images by simple LSB substitution. Pattern Recognit. **37**(3), 469–474 (2004)
8. Chang, C.C.; Chen, T.S.; Chung, L.Z.: A steganographic method based upon JPEG and quantization table modification. Inf. Sci. **141**, 123–138 (2002)
9. Noda, H.; Furuta, T.; Niimi, M.; Kawaguchi, E.: Application of BPCS steganography to wavelet compressed video. Proc. Int. Conf. Image Process. **4**, 2147–2150 (2004)
10. Wang, R.Z.; Lin, C.F.; Lin, J.C.: Image hiding by optimal LSB substitution and genetic algorithm. Pattern Recognit. **34**(3), 671–683 (2001)
11. Zhang, W.; Wang, S.; Zhang, X.: Improving embedding efficiency of covering codes for applications in steganography. IEEE Commun. Lett. **11**(8), 680–682 (2007)
12. Petitcolas, F.A.P.; Anderson, R.J.; Kuhn, M.G.: Information hiding-a survey. Proc. IEEE **87**(7), 1062–1078 (1999)
13. Li, X.; Wang, J.: A steganographic method based upon JPEG and particle swarm optimization algorithm. Inf. Sci. **177**, 3099–3109 (2007)
14. Raid, A.M.; Khedr, W.M.; El-dosuky, M.A.; Ahmed, W.: JPEG image compression using discrete cosine transform: a survey. Int. J. Comput. Sci. Eng. Surv. (IJCSES) **5**(2), 60–70 (2014)
15. Klapetek, P.: Februrary 2002. http://klapetek.cz/wdwt.html. Accessed 23 May 2017
16. Bhattacharyya, D.; Kim, T.: Image data hiding technique using discrete fourier transformation. In: Ubiquitous Computing and Multimedia Applications Volume 151 of the Series Communications in Computer and Information Science, April 13–15, pp. 315–323 (2011)
17. Wang, H.; Wang, S.: Cyber warfare: steganography vs steganalysis. Commun. ACM Voting Syst. **47**(10), 76–82 (2004)
18. Venkatraman, S.; Abraham, A.; Paprzycki, M.: Significance of steganography on data security. In: IEEE (2004)
19. Fridrich, J.: Methods for tamper detection in digital images. In: Workshop at multimedia and security at ACM Multimedia, Florida (1999)

20. Bohme, R.: Principles of modern steganography and steganalysis. Adv. Stat. Steganal. Inf. Secur. Cryptogr. **0**, 11–77 (2010)
21. Pannebaker, W.B.; Mitchell, J.L.: JPEG: Still Image Data Compression Standard. Van Nostrand Reinhold, New York (1993)
22. Huang, J.; Shi, Y.Q.; Shi, Y.: Embedding image watermarks in DC components. IEEE Trans. Circuits Syst. Video Technol. **10**(6), 974–979 (2000)
23. Miano, J.: Compressed Image File Formats: JPEG, PNG, GIF, XBM, BMP. Addison-Wesley, Boston (1999)
24. Tseng, H.W.; Chang, C.C.: Steganography using JPEG-compressed images. In: The Fourth International Conference on Computer and Information Technology, pp. 12–17. IEEE Computer Society Press, Wuhan (2004)
25. Yu, Y.H.; Chang, C.C.; Hu, Y.C.: Hiding secret data in images via predictive coding. Pattern Recognit. **38**, 691–705 (2005)
26. Jiang, C.; Pang, Y.; Guo, L.; Jing, B.; Gong, X.: A High Capacity Steganographic Method Based on Quantization Table Modification, vol. 16, 3rd edn, pp. 223–227. Springer, Berlin (2011)
27. Almohammad, A.; Ghinea, G.; Hierons, R.M.: JPEG steganography: a performance evaluation of quantization tables. In: International Conference on Advanced Information Networking and Applications (2009)
28. Rao, K.; Yip, P.: Discrete Cosine Transform: Algorithms, Advantages, Applications. Academic Press, Boston (1990). ISBN 0-12-580203-X
29. Kobayashi, H.; Noguchi, Y.; Kiya, H.: A method of embedding binary data into JPEG bitstreams. IEICE Trans. Inf. Syst. **J83–D–II**, 1469–1476 (1999)
30. Kulkarni, M.: An information hiding system using 16*16 quantization table. In: International Conference on Advances in Communication and Computing Technologies (ICACACT 2014), pp. 1–4 (2014)
31. Roy, R.; Laha, S.: Optimization of stego image retaining secret information using genetic algorithm with 8-connected PSNR. In: Proceedings of the 19th International Conference on Knowledge Based and Intelligent Information and Engineering Systems, Procedia Computer Science, vol. 60, pp. 468–477 (2015)
32. Khamrui, A.; Mandal, J.K.: A genetic algorithm based steganography using discrete cosine transformation (GASDCT). In: International Conference on Computational Intelligence: Modeling Techniques and Applications (CIMTA) 2013, Procedia Technology, vol. 10, pp. 105–111 (2013)
33. Prema, G.; Natarajan, S.: Steganography using genetic algorithm along with visual cryptography for wireless network application. In: International Conference on Information Communication and Embedded Systems (ICICES), IEEE (2013). doi:10.1109/ICICES.2013.6508373
34. Ghasemi, E.; Shanbehzadeh, J.; Fassihi, N.: High capacity image steganography based on genetic algorithm and wavelet transform. In: Intelligent Control and Innovative Computing, Lecture Notes in Electrical Engineering, vol. 110. Springer (2012)
35. Wang, S.; Yang, B.; Niu, X.: A secure steganography method based on genetic algorithm. J. Inf. Hiding Multimed. Signal Process. **1**(1), 28–35 (2010)
36. Hemanth, D.J.; Umamaheswari, S.; Popescu, D.E.; Naaji, A.: Application of genetic algorithm and particle swarm optimization techniques for improved image steganography systems. Open Phys. **14**(1), 452–462 (2016)
37. El-Emam, N.N.: New data-hiding algorithm based on adaptive neural networks with modified particle swarm optimization. Comput. Secur. **55**, 21–45 (2015)
38. Nickfarjam, A.M.; Azimifar, Z.: Image steganography based on pixel ranking and particle swarm optimization. In: The 16th CSI International Symposium on Artificial Intelligence and Signal Processing (AISP 2012), IEEE (2012). doi:10.1109/AISP.2012.6313773
39. Edward, J.S.; Palaniappan, R.; Ramakrishnan, S.: Imperceptibility–robustness tradeoff studies for ECG steganography using continuous ant colony optimization. Expert Syst. Appl. **49**(1), 123–135 (2016)
40. Wasan, S.A.: Information hiding using ant colony optimization algorithm. Int. J. Technol. Diffus. **2**(1), 16–28 (2011)
41. Kulkarni, A.J.; Durugkar, I.P.; Kumar, M.: Cohort intelligence: a self supervised learning behaviour. In: IEEE International Conference on Systems, Man and Cybernetics, IEEE Computer Society, pp. 1396–1400 (2013)
42. Kulkarni, A.J.; Baki, M.F.; Chaouch, B.A.: Application of the cohort intelligence optimization method to three selected combinatorial optimization problems. Eur. J. Oper. Res. **250**(2), 11–25 (2016)
43. Kulkarni, A.J.; Shabir, H.: Solving 0–1 knapsack problem using cohort intelligence algorithm. Int. J. Mach. Learn. Cybern. **7**(3), 427–441 (2016)
44. Krishnasamy, G.; Kulkarni, A.J.; Paramesran, R.: A hybrid approach for data clustering based on modified cohort intelligence and K-means. Expert Syst. Appl. **41**, 6009–6016 (2014)
45. Dhavle, S.V.; Kulkarni, A.J.; Shastri, A.; Kale, I.R.: Design and economic optimization of shell-and-tube heat exchanger using cohort intelligence algorithm. In: Proceedings of the Natural Computing Applications Forum 2017. Springer (2016)
46. Kulkarni, O.; Kulkarni, N.; Kulkarni, A.J.; Kakandikar, G.: Constrained cohort intelligence using static and dynamic penalty function approach for mechanical components design. Int. J. Parallel Emerg. Distrib. Syst. (2016). doi:10.1080/17445760.2016.1242728
47. Kale, I.R.; Kulkarni, A.J.: Cohort intelligence algorithm for discrete and mixed variable engineering problems. Int. J. Parallel Emerg. Distrib. Syst. (2017). doi:10.1080/17445760.2017.1331439
48. Shah, P.; Agashe, S.; Kulkarni, A.J.: Design of fractional PID controller using cohort intelligence method. Front. Inf. Technol. Electron. Eng. (2017) (**in press**)
49. Zhu, F.: Blocking artifacts reduction in compressed data. In: Proceedings of the 2009 International Conference on Computer Engineering and Applications, IPCSIT, vol. 2. IACSIT Press, Singapore (2011)
50. Langelaar, G.C.; Setyawan, I.; Lagendijk, R.L.: Watermarking digital image and video data. A state-of-the-art overview. In: IEEE Signal Processing Society, pp. 20–46 (2002)
51. Patankar, N.S.; Kulkarni, A.J.: Variations of cohort intelligence. In Press: Soft Computing (2017). doi:10.1007/s00500-017-2647-y
52. Aote, S.S.; Raghuwanshi, M.M.; Malik, L.G.: Improved particle swarm optimization based on natural flocking behavior. Arab. J. Sci. Eng. **41**(3), 1067–1076 (2016)
53. Rakhshani, H.; Rahati, A.: Intelligent multiple search strategy cuckoo algorithm for numerical and engineering optimization problems. Arab. J. Sci. Eng. **42**, 567–593 (2017)
54. Teo, T.H.; Kulkarni, A.J.; Kanesan, J.; Chuah, J.H.; Abraham, A.: Ideology algorithm: a socio-inspired optimization methodology. Neural Comput. Appl. **2016**, 1–32 (2016). doi:10.1007/s00521-016-2379-4
55. Rafat, K.F.; Sher, M.: Secure digital steganography for ASCII text documents. Arab. J. Sci. Eng. **38**(8), 2079–2094 (2013)