DOI: 10.21917/ijsc.2017.0210

SINGLE VERSUS MULTIPLE TRIAL VECTORS IN CLASSICAL DIFFERENTIAL EVOLUTION FOR OPTIMIZING THE QUANTIZATION TABLE IN JPEG BASELINE ALGORITHM

B. Vinoth Kumar[1] and G.R. Karpagam[2]

Department of Computer Science and Engineering, PSG College of Technology, India

Abstract

Quantization Table is responsible for compression / quality trade-off in baseline Joint Photographic Experts Group (JPEG) algorithm and therefore it is viewed as an optimization problem. In the literature, it has been found that Classical Differential Evolution (CDE) is a promising algorithm to generate the optimal quantization table. However, the searching capability of CDE could be limited due to generation of single trial vector in an iteration which in turn reduces the convergence speed. This paper studies the performance of CDE by employing multiple trial vectors in a single iteration. An extensive performance analysis has been made between CDE and CDE with multiple trial vectors in terms of Optimization process, accuracy, convergence speed and reliability. The analysis report reveals that CDE with multiple trial vectors improves the convergence speed of CDE and the same is confirmed using a statistical hypothesis test (t-test).

Keywords:

Meta-Heuristic Search, Differential Evolution, Trial Vectors, Image Compression, JPEG, Quantization Table, Optimization, Statistical Hypothesis Test and t-Test

1. INTRODUCTION

Joint Photographic Experts Group (JPEG) is a famous still image compression standard and it is dominating the other image file formats. According to w3tech survey [1], around 73.9% of images on the internet are in JPEG format. In a JPEG baseline algorithm [2], there are four major steps, namely (i) Dividing an image into 8 × 8 blocks, (ii) Applying Discrete Cosine Transform for each block, (iii) Performing quantization for each block and (iv) Applying entropy encoding for each block. Among these steps, quantization plays a significant role in image quality / compression trade-off. Quantization is performed by 8 × 8 quantization table which is recommended by the Independent JPEG Group (IJG). Also, this group allows the users to customize the quantization table for their applications.

Many researchers tried to optimize the quantization table using meta-heuristic approaches [3]-[4] such as Simulated Annealing [5], Genetic Algorithm [6]-[12], Chaos Evolutionary Programming [13], Particle Swarm Optimization [14], Firefly algorithm [15], Differential Evolution [16]-[17] and Quantum Genetic Algorithm [18]. Kumar et al. [10] has been proved that Classical Differential Evolution (CDE) is a promising technique to optimize the quantization table for the JPEG baseline algorithm.

Storn and Price [19] introduced a population based optimization algorithm called Differential Evolution (DE). Initialization, Mutation, Crossover and Selection are the important operators in DE algorithm. There are two crossover strategies, namely binomial and exponential, used in the DE algorithm. There are many DE variants such as DE/Rand/1, DE/Rand/2, DE/Best/1, DE/Best/2, [19]-[20] available in the literature by varying the above said operators. Among these variants, “DE/Rand/1/bin” is identified as a Classical Differential Evolution in which the exploration capability is very strong and it is suitable for multimodal problems [21]-[22]. In CDE, only one trial vector will be generated which could limit the convergence speed. Some researchers tried to improve the convergence speed by employing multiple trial vectors in different DE variants [23]-[24]. However, the employing of multiple trial vectors would increase the computation time for a high dimensional combinatorial problem such as quantization table optimization. Therefore, the number of trial vectors plays an important role for this kind of problem. Although employing of multiple trail vectors in the DE algorithm are available in the literature, it has been never used for this application.

This paper studies whether the performance of CDE can be improved by employing multiple trail vectors in a single iteration. The performance of CDE with multiple trial vectors is analyzed by Average Best Unfitness value, Average Best of Generations, Optimization Accuracy, Probability of Convergence, Average number of function Evaluations and Successful Performance. The analysis reports prove that CDE with multiple trial vectors performs better than CDE and the same is confirmed by using a statistical hypothesis test.

The rest of this paper is organized as follows. A brief review of the CDE algorithm is given in section 2. Section 3 illustrates the CDE with multiple trial vectors. The various performance measures are explained in section 4. The experiments and results are discussed in section 5. Final thoughts are concluded in section 6.

2. CLASSICAL DIFFERENTIAL EVOLUTION (CDE)

From the current population G, the mutant vector $v_{i,G}$ is calculated as shown in Eq.(1) by using the randomly selected three chromosomes $x_{r1,G}$, $x_{r2,G}$ and $x_{r3,G}$. Here the scaling factor F is chosen between 0 and 1 to control the evolution. A binomial uniform crossover based on crossover probability C_r is performed between mutant vector $v_{i,G}$ and target vector $x_{i,G}$ to form trial vector $u_{i,G}$. It is shown in Eq.(2). A better vector is selected for next generation G+1 as shown in Eq.(3). The algorithm 1 shows the pseudo code of CDE, which is adopted from Kumar et al. [16].

$$v_{i,G} = x_{r1,G} + F(x_{r2,G} - x_{r3,G}) \tag{1}$$

$$u_{i,G} = u_{j,i,G} = \begin{cases} v_{j,i,G} & \text{if } \left(rand_j(0,1) \le C_r \text{ or } j = j_{rand}\right) \\ x_{j,i,G} & \text{otherwise} \end{cases} \tag{2}$$

$$x_{i,G+1} = \begin{cases} x_{i,G} & \text{if } \left(fitness\left(x_{i,G}\right) \le fitness\left(u_{i,G}\right)\right) \\ u_{i,G} & \text{otherwise} \end{cases} \quad (3)$$

Algorithm 1: Classical Differential Evolution-Pseudo Code

Initialize population of vectors randomly;

Evaluate the vectors;

While Maximum Generation not reached do

For all vectors do

Select the target vector;

Choose 3 vectors in the population randomly;

Compute the mutant vector;

Perform crossover between the target and mutant vectors to form trial vector;

Evaluate the trial vector;

Replace target vector by trial vector if unfitness value of trial vector is smaller than target vector;

End for

End while

Return best vector;

3. CDE WITH MULTIPLE TRIAL VECTORS

The mutant vector in CDE is computed by taking the difference between two random vectors. But all difference vectors have a negative counterpart and an equal chance of being chosen [25].

$$\begin{aligned} v_{i,G} &= x_{r1,G} + F\left(x_{r2,G} - x_{r3,G}\right) \\ v_{\hat{i},G} &= x_{r1,G} + F\left(x_{r3,G} - x_{r2,G}\right) \end{aligned} \quad (4)$$

$$u_{i,G} = u_{j,i,G} = \begin{cases} v_{j,i,G} & \text{if } \left(rand_j\left(0,1\right) \le C_r \text{ or } j = j_{rand}\right) \\ x_{j,i,G} & \text{otherwise} \end{cases} \quad (5)$$

$$u_{\hat{i},G} = u_{j,\hat{i},G} = \begin{cases} v_{j,\hat{i},G} & \text{if } \left(rand_j\left(0,1\right) \le C_r \text{ or } j = j_{rand}\right) \\ x_{j,i,G} & \text{otherwise} \end{cases} \quad (6)$$

$$x_{i,G+1} = \begin{cases} x_{i,G} & \text{if } \left(\begin{array}{l} fitness\left(x_{i,G}\right) \le \\ \min\left(fitness\left(u_{i,G}\right), fitness\left(u_{\hat{i},G}\right)\right) \end{array} \right) \\ u_{i,G} & \text{if } \left(\begin{array}{l} fitness\left(u_{i,G}\right) \le \\ \min\left(fitness\left(x_{i,G}\right), fitness\left(u_{\hat{i},G}\right)\right) \end{array} \right) \\ u_{\hat{i},G} & \text{otherwise} \end{cases} \quad (7)$$

Here, the difference between two random vectors and its negative counterpart is taken; accordingly two separate mutant vectors are obtained. The CDE with multiple trial vectors is represented as DE/Rand/1*/bin. The computation of mutant vector in the DE/Rand/1*/bin is shown in Eq.(4). In each generation two separate trial vectors are computed which is shown in Eq.(5) and Eq.(6). Both the trial vectors are evaluated by fitness function and the best among them is considered to compare with the target vector, shown in Eq.(7).

4. PERFORMANCE MEASURES

In order to compare the performance of CDE and CDE with multiple trial vectors (CDE-MTV), the measures given in Table.1 are taken from the paper [16] [17] to validate the efficiency of both algorithms.

Table.1. Performance Measures to Evaluate the algorithms

Measure	Description	Evaluation Criteria
Average Best Unfitness Value $f_a(k)$	Calculates the best unfitness value after particular computation time k, averaged over the total number of independent runs n. $$f_a\left(k\right) = \frac{\sum_{runs=1}^{n} \text{Best Unfitness value}\left(k\right)}{n}$$	Entire optimization process
Average Best of generation $\overline{BOG}$	Calculates the best-of-generation unfitness over all particular computation time k, and over the total number of runs n. $$\overline{BOG} = \frac{1}{n}\frac{1}{k}\sum_{r=1}^{n}\sum_{g=1}^{k} f\left(BOG_{rg}\right)$$ where f (BOG_{rg}) expresses the unfitness value of the best solution at generation g of run r (among n independent runs).	Entire optimization process
Optimization Accuracy Acc_k	Determines the location of the best found solution between the lower (worst known solution) and upper bound (best known solution). It may vary from 0 (worst) to 1 (best). $$Acc_k = \frac{f_a\left(k\right) - Min_s}{Max_s - Min_s}$$ where, $f_a(k)$ = Average best Unfitness value at a particular generation k. Min_s = Worst known solution, Max_s= Best known solution	Accuracy
Probability of Convergence, P	Calculates the number of successful trials (s) in the total number of independent runs (n) $$P = \frac{s}{n}$$ It may vary from 0 (worst) to 1 (best).	Reliability
Average number of Function Evaluations, $AFES$	Calculates the average number of evaluations required to reach the vicinity of the best known value in each successful trial. A lower value is preferred. $$AFES = \frac{1}{s}\sum_{i=1}^{s} EVAL_i$$ where, $EVAL_i$ = number of function evaluation in the successful run i	Convergence speed
Successful Performance, SP	Calculates the ratio of average number of function evaluations to the probability of convergence	Convergence speed

	$SP = \frac{AFES}{P}$ A lower value is preferred.	and Reliability

5. EXPERIMENTAL RESULTS AND DISCUSSIONS

The focus of this paper is to optimize the quantization table using CDE by employing multiple trial vectors. Here every Quantization table which is an 8×8 vector which has 64 elements. An unfitness function used in this study to evaluate the quantization table is shown in Eq.(8).

$$\xi = a\ (8/B_r - \lambda)^2 + \varepsilon \quad (8)$$

where, a =10, B_r = Bit rate, λ = desired compression ratio = 8/(target bits per pixel), ε =Mean squared error.

For a basic understanding of CDE with multiple trial vectors, a simple example is shown in Table.2. In this example the initial population is 4, scaling factor F is a 0.3 and crossover probability is 0.8. The unfitness values of the initial population are {164.34, 476.45, 157.49, 193.48}.

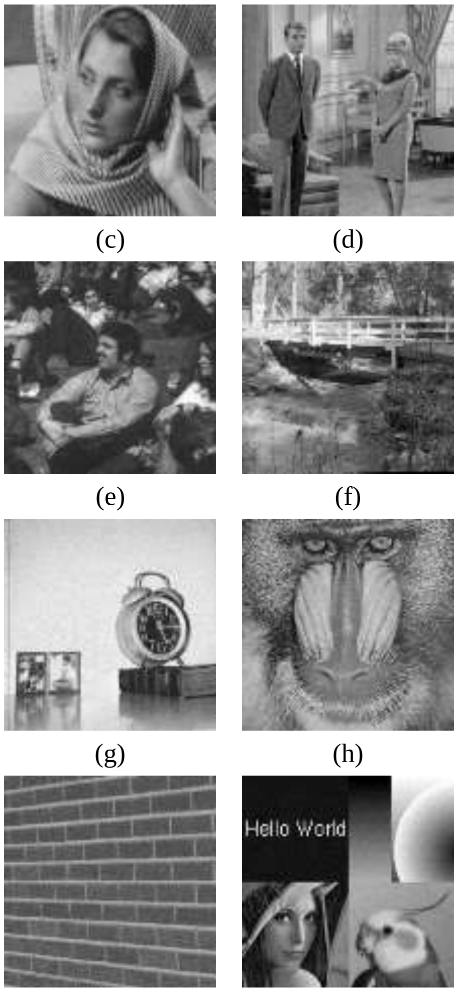
For each generation the CDE with multiple trial vectors generates two trial vectors. The unfitness values of these trial vectors are compared with target vector and the one with least unfitness value is selected for next population. The unfitness values after two generations are {115.30, 146.98, 157.49, 136.79}.

The Algorithms are implemented in Matlab R2008b and Dell workstation of Intel® Xenon® CPU E3-1240 V3 @ 3.40 GHz processor with 16 GB of RAM. CDE and CDE with multiple trial vectors have been run for the standard benchmark images shown in Fig.1. All the images are grey scale with size of 256 × 256. The parameter settings for both the algorithms are shown in Table.3.

The programs executed 20 times for each image against each of the target bits/pixel: 0.75 and 1.0 and 1.5. The quality of quantization Table.is evaluated by employing two performance measures; Mean Squared Error (MSE) and Peak Signal to Noise Ratio (PSNR). The mean result among 20 runs for CDE-MTV based quantization table, CDE based quantization table and default JPEG quantization tables is presented in Table.4.

From the Table.4, it is clearly shown that CDE-MTV based quantization table yields better results than other two with less MSE and high PSNR. CDE-MTV quantization tables reduces the MSE on an average by 15.6% and 23.14% over the CDE based quantization tables and default JPEG quantization tables respectively.

(a) (b)



(c) (d) (e) (f) (g) (h) (i) (j)

Fig.1. Standard benchmark images (a) lena (b) camera man (c) barbara (d) couple (e) crowd (f) bridge (g) clock (h) baboon (i) pattern (j) montage

To analyze the performance of CDE and CDE with multiple trial vectors in detail, the measures given in Table.1 have been taken into consideration. These measures are calculated for ten different images shown in Fig.1 with three different target bits/pixel 0.75, 1.0 and 1.5 in 20 independent runs. The summary results of the performance measures are reported in Table.5-Table.10, and measure by measure analyzation is given in the subsequent paragraphs.

Table.5 shows the average best unfitness value at various computation times for different target bits/pixel. From the Table.5, it is clear that CDE with multiple trial vectors has better unfitness value over CDE after each particular computation time. In addition, CDE with multiple trial vectors achieves the better $f_a(k)$ than CDE in 2000 seconds where the same is achieved by CDE within 4000 seconds.

Average best-of-generation for the periods 1 to 1000 seconds, 1001 to 2000 seconds, 2001 to 3000 seconds and 3001 to 4000 seconds are summarized in Table.6 for different target bits/pixel. From the Table.6, it is clearly seen that the $\overline{BOG}$ of CDE with multiple trial vectors in each period is lesser than CDE, which confirms that CDE with multiple trial vector is better than CDE for the entire optimization process.

Table.2. An example for CDE with Multiple Trial Vectors progress

First Generation Second Generation				
Initial Population	**Trial Vectors**	**Selection/ Population for next generation**	**Trial Vectors**	**Selection/ Population for next generation**
164.34	226.86 153.22	153.22	115.30 167.90	115.30
476.45	363.43 195.02	195.02	272.23 146.98	146.98
157.49	169.56 259.08	157.49	233.52 237.65	157.49
193.48	144.33 203.89	144.33	152.92 136.79	136.79

Table.3. Parameter settings for CDE and CDE with multiple trial vectors (CDE-MTV)

Parameter	CDE	CDE-MTV
Population Size	32	32
Scaling Factor	0.3	0.3
Crossover Probability	0.8	0.8
Computation time	4000 sec	4000 sec

Table.7 summarizes the optimization accuracy value after 1000, 2000, 3000 and 4000 seconds for different target bits/pixel. From the Table.7, it has been noted that the Acc_k of CDE with multiple trial vectors in 2000 seconds is same as CDE in 4000 seconds, which shows CDE with multiple trial vectors is very close to an optimal solution in a lesser time.

Table.4. Comparison of image quality measures for CDE and CDE with Multiple Trial vectors (CDE-MTV)

Target Bits/Pixel		0.75		1		1.5	
Image	**Quantization Table**	**MSE**	**PSNR in dB**	**MSE**	**PSNR in dB**	**MSE**	**PSNR in dB**
Lena	**JPEG**	51.96	31.01	34.29	32.81	19.26	35.31
	CDE	46.83	31.46	35.61	32.65	19.87	35.18
	CDE-MTV	43.58	31.77	27.39	33.79	14.69	36.5
Camera Man	**JPEG**	66.24	29.95	44.25	31.71	22.29	34.68
	CDE	54.73	30.78	35.34	32.68	20.06	35.15
	CDE-MTV	51.81	31.02	30.41	33.34	12.98	37.03
Barbara	**JPEG**	61.71	30.26	41.93	31.94	16.92	35.88
	CDE	51.05	31.09	33.38	32.39	16.39	36.02
	CDE-MTV	43.33	31.79	29.56	33.46	11.67	37.49
Clock	**JPEG**	24.28	34.31	14.64	36.51	7.21	39.58
	CDE	24.73	34.23	17.91	35.63	9.95	38.19
	CDE-MTV	21.97	34.75	13.56	36.84	5.93	40.43
Bridge	**JPEG**	157.67	26.19	120.02	27.37	75.77	29.37
	CDE	150.09	26.4	113.16	27.63	81.22	29.07
	CDE-MTV	145.64	26.53	98.41	28.23	62.76	30.19
Couple	**JPEG**	49.57	31.21	34.31	32.81	19.49	35.27
	CDE	48.15	31.34	36.94	32.49	22.44	35.06
	CDE-MTV	45.03	31.63	29.90	33.4	14.94	36.42
Crowd	**JPEG**	40.51	32.09	27.25	33.81	14.77	36.47
	CDE	43.95	31.74	27.59	33.75	14.98	36.41
	CDE-MTV	39.18	32.23	25.44	34.11	11.89	37.41
Baboon	**JPEG**	404.18	22.1	330.00	22.98	223.28	24.68
	CDE	372.19	22.46	314.8	13.19	191.35	25.35
	CDE-MTV	353.71	22.68	254.93	24.10	161.78	26.08
Pattern	**JPEG**	58.63	30.48	48.10	31.34	35.91	32.61
	CDE	45.05	31.63	53.64	30.86	29.07	33.53
	CDE-MTV	41.56	31.98	42.89	31.84	25.50	34.10
Montage	**JPEG**	25.20	34.15	13.55	36.85	5.51	40.76
	CDE	23.50	34.45	17.14	35.82	9.49	38.39
	CDE-MTV	20.21	35.11	11.18	37.68	4.29	41.84

Table.5. Summary of Average Unfitness value for various bits/pixel

bpp	CDE				CDE with Multiple Trial Vectors			
	Computation time in seconds				**Computation time in seconds**			
	After 1000	**After 2000**	**After 3000**	**After 4000**	**After 1000**	**After 2000**	**After 3000**	**After 4000**
0.75	116.99	97.68	90.88	88.22	99.07	85.66	79.97	76.90
1	95.71	78.31	71.45	68.86	86.99	72.44	66.41	63.31
1.5	68.53	50.74	44.16	41.80	53.36	40.81	37.06	35.40
Avg	**93.75**	**75.58**	**68.84**	**66.30**	**79.81**	**66.31**	**61.15**	**58.54**

Table.6. Summary of Average best-of-Generations for various bits/pixel

bpp	CDE				CDE with Multiple Trial Vectors			
	Computation time in seconds				**Computation time in seconds**			
	1 to 1000	**1001 to 2000**	**2001 to 3000**	**3001 to 4000**	**1 to 1000**	**1001 to 2000**	**2001 to 3000**	**3001 to 4000**
0.75	139.66	105.19	93.70	89.31	119.54	91.25	82.28	78.21
1	118.50	85.22	74.25	69.98	109.27	77.60	68.85	64.67
1.5	100.02	57.88	46.74	42.77	87.01	45.66	38.60	36.07
Avg	**119.39**	**82.76**	**71.57**	**67.36**	**105.27**	**71.50**	**63.24**	**59.65**

Table.7. Summary of Optimization Accuracy for various bits/pixel

bpp	CDE				CDE with Multiple Trial Vectors			
	Computation time in seconds				Computation time in seconds			
	After 1000	After 2000	After 3000	After 4000	After 1000	After 2000	After 3000	After 4000
0.75	0.61	0.79	0.86	0.88	0.80	0.92	0.97	1.00
1	0.68	0.85	0.92	0.94	0.78	0.91	0.97	1.00
1.5	0.76	0.89	0.94	0.95	0.87	0.96	0.99	1.00
Avg	**0.68**	**0.84**	**0.90**	**0.93**	**0.82**	**0.93**	**0.98**	**1.00**

The Table.8 shows the *P* measure of both the algorithms for different target bits/pixel. The *P* measure value of CDE shows that it does not able to reach the optimal solution at all runs for all images within the preset maximum computation time, whereas CDE with multiple trial vectors is able to reach the optimal solution at all runs for all images. The Table.9 and Table.10 shows the AFES measure and SP measure of both the algorithms for different target bits/pixel. Both the measures prefer the lower values. From the Table.9 and Table.10, it is clear that CDE with multiple trial vectors is able to reach the optimal solution consistently within a lesser computation time. *AFES* and *SP* measures could not be calculated for CDE for some images because they do not produce any optimal solution over the preset maximum computation time.

Table.8. Probability of Convergence for various bits/pixel

Algorithm	CDE			CDE with Multiple Trial Vectors		
bpp	**0.75**	**1**	**1.5**	**0.75**	**1**	**1.5**
Lena	1	0	0	1	1	1
Camera man	1	1	0.8	1	1	1
Barbara	1	1	1	1	1	1
Clock	0.2	0	0	1	1	1
Bridge	1	0.8	0	1	1	1
Couple	0.8	0	0	1	1	1
Crowd	0	0	0.2	1	1	1
Baboon	1	1	1	1	1	1
Pattern	1	0	1	1	1	1
Montage	1	0	0	1	1	1

Table.9. Average Number of Function Evaluations for various bits/pixel

Algorithm	CDE			CDE with Multiple Trial Vectors		
bpp	**0.75**	**1**	**1.5**	**0.75**	**1**	**1.5**
Lena	74	-	-	62	90	98
Camera man	73	94	122	32	84	66
Barbara	58	66	112	50	54	72
Clock	123	-	-	58	80	132
Bridge	72	85	-	66	68	76
Couple	105	-	-	34	96	98
Crowd	-	-	129	72	132	112
Baboon	86	61	62	66	54	46
Pattern	45	-	60	36	108	56
Montage	103	-	-	36	106	108

The above analysis confirms that CDE with multiple trial vectors performs better than CDE; however, it is necessary to confirm the results statistically. Hence, one tailed t-test (hypothesis testing) is used to compare the performance of both the algorithms. As a null hypothesis, H_0 is assumed that there is no significant difference between the CDE and CDE with multiple trial vectors, whereas the alternative hypothesis H_1 is that CDE with multiple trial vectors is more efficient than CDE at the 5% significance level.

Table.10. Successful Performance for various bits/pixel

Algorithm	CDE			CDE with Multiple Trial Vectors		
Bpp	**0.75**	**1**	**1.5**	**0.75**	**1**	**1.5**
Lena	74	-	-	62	90	98
Camera man	73	94	152.5	32	84	66
Barbara	58	66	112	50	54	72
Clock	615	-	-	58	80	132
Bridge	72	106.25	-	66	68	76
Couple	131.25	-	-	34	96	98
Crowd	-	-	645	72	132	112
Baboon	86	61	62	66	54	46
Pattern	45	-	60	36	108	56
Montage	103	-	-	36	106	108

Table.11. One tailed t-test results for different performance Measures

Measures	P value in t test				Significance Level
	After 1000 sec	After 2000 sec	After 3000 sec	After 4000 sec	
Average Best Unfitness Value	0.018	0.017	0.023	0.024	0.05
Average Best of Generations	0.023	0.013	0.020	0.023	
Optimization Accuracy	0.021	0.029	0.036	0.036	

The statistical test is performed only for Average Best Unfitness value, Average Best of Generations, Optimization Accuracy, because for other measures CDE could not able to reach the vicinity of optimal solutions. One tailed t-test is performed on above said measures with 0.05 as the level of significance (α) and their p-values are shown in Table.11. The null hypothesis is rejected, when the obtained p-value is less than α, otherwise it is not rejected. From the Table.11, it is observed that p-value of all performance measures is less than 0.05 which indicates the rejection of the null hypothesis H_0. Therefore, the statistical results confirm that CDE with multiple trial vectors is more efficient than CDE with a confidence level of 95%.

6. CONCLUSIONS

In this paper, a Classical Differential Evolution with multiple trial vectors has been proposed to search the optimal quantization table for the JPEG baseline algorithm. CDE with multiple trial vectors based quantization tables reduces the MSE on an average by 15.6% and 23.14% over the CDE based quantization tables and default JPEG quantization tables respectively. Employing multiple trial vectors in a single iteration accelerate the search which in turn improves the convergence speed. Also an extensive comparative analysis has been made between CDE and CDE with multiple trial vectors in terms of their optimization process, accuracy, convergence speed and reliability. The analysis report shows that CDE with multiple trial vectors guarantees an optimal solution in a lesser time. Also the empirical results have been confirmed by statistical hypothesis test (t-test). Possible direction for the future work includes the employing of different multiple trial vector generation strategies for this application by considering the time taken for computation.

REFERENCES

[1] W3Tech, "Usage of JPEG for websites", Available at:www.w3techs.com/technologies/details/im-jpeg/all/all. Accessed on 2017.

[2] Gregory K. Wallace, "The JPEG still Picture Compression Standard", *Communications of the ACM-Special Issue on Digital Multimedia Systems*, Vol. 34, No. 4, pp.30-44, 1991.

[3] S. Viswajaa, B. Vinoth Kumar and G.R. Karpagam, "A Survey on Nature Inspired Meta-Heuristics Algorithms in Optimizing the Quantization Table for the JPEG Baseline Algorithm", *International Advanced Research Journal in Science, Engineering and Technology*, Vol. 2, No. 4.pp. 114-123, 2015.

[4] S.P. Naresh, B. Vinoth Kumar and G.R. Karpagam, "A Literature Review on Quantization Table Design for the JPEG Baseline Algorithm", *International Journal of Engineering and Computer Science*, Vol. 4, No. 10, pp. 14686-14691, 2015.

[5] B.G. Sherlock, A. Nagpal and D.M. Monro, "A Model for JPEG Quantization", *Proceedings of International Symposium on Speech, Image Processing and Neural Networks*, pp. 176-179, 1994.

[6] Yung-Gi Wu, "GA-based DCT Quantization Table Design Procedure for Medical Images", *IEE Proceedings-Vision, Image and Signal Processing*, Vol. 151, No. 5, pp. 353-359, 2004.

[7] M.R. Boyandi, E. Dehghani and M.E. Moghaddam, "A Non-Uniform Image Compression using Genetic Algorithm", *Proceedings of 15th International Conference on Systems, Signals and Image Processing*, pp. 315-318, 2008.

[8] Shuwang Chen, Tao An and Litao Hao, "Discrete Cosine Transform Image Compression Based on Genetic Algorithm", *International Conference on Information Engineering and Computer Science*, pp. 1-3, 2009.

[9] Beatrice Lazzerini, Francesco Marcelloni and Massimo Vecchio, "A Multi-Objective Evolutionary Approach to Image Quality/Compression Trade-Off in JPEG Baseline Algorithm", *Applied Soft Computing*, Vol. 10, No. 1, pp. 548-561, 2010.

[10] B. Vinoth Kumar, G.R. Karpagam and N. Vijaya Rekha, "Performance Analysis of Deterministic Centroid Initialization Method for Partitional Algorithms in Image Block Clustering", *Indian Journal of Science and Technology*, Vol. 8, No. 7, pp. 63-73, 2015.

[11] B Vinoth Kumar and Karpagam Manavalan, "Knowledge based Genetic Algorithm Approach to Quantization Table Generation for the JPEG Baseline Algorithm", *Turkish Journal of Electrical Engineering and Computer Sciences*, Vol. 24. No. 3, pp. 1615-1635, 2016.

[12] B Vinoth Kumar and Karpagam Manavalan, "A Problem Approximation Surrogate Model (PASM) for Fitness Approximation in Optimizing the Quantization Table for the JPEG Baseline Algorithm", *Turkish Journal of Electrical Engineering and Computer Sciences*, Vol. 24, No. 6, pp. 4623-4636, 2016.

[13] S.M. Guo, J.S.H Tsai, W.H. Chang, B.H. Wu and S.J. Guo, "Chaos Evolutionary Programming based JPEG Quantization Table Generation Scheme", *Proceedings of 5th International Conference on Informatics and Systems*, pp. 1-7, 2007.

[14] Huizhu Ma and Qiuju Zhang, "Research on Cultural-based Multi-Objective Particle Swarm Optimization in Image Compression Quality Assessment", *Optik-International Journal for Light and Electron Optics*, Vol. 124, No. 10, pp. 957-961, 2012.

[15] Milan Tuba and Nebojsa Bacanin, "JPEG Quantization Tables Selection by the Firefly Algorithm", *Proceedings of International Conference on Multimedia Computing and Systems*, pp. 1-6, 2014.

[16] B. Vinoth Kumar and M. Karpagam, "Differential Evolution versus Genetic Algorithm in Optimising the Quantisation Table for JPEG Baseline Algorithm", *International Journal of Advanced Intelligence Paradigms*, Vol. 7, No. 2, pp. 111-135, 2015.

[17] B. Vinoth Kumar and Karpagam Manavalan, "Knowledge based Differential Evolution Approach to Quantization Table Generation for the JPEG Baseline Algorithm", *International Journal of Advanced Intelligence Paradigms*, Vol. 8, No. 1, pp. 20-41, 2016.

[18] B. Vinoth Kumar, G.R. Karpagam and S.P. Naresh, "Generation of JPEG Quantization Table using Real Coded Quantum Genetic Algorithm", *Proceedings of IEEE International Conference on Communication and Signal Processing*, pp. 1705-1709, 2016.

[19] Rainer Storn and Kenneth Price, "Differential Evolution: A Simple and Efficient Adaptive Scheme for Global Optimization over Continuous Spaces", *Journal of Global Optimization*, Vol. 11, No. 4, pp. 341-359, 1997.

[20] Swagatam Das and Ponnuthurai Nagaratnam Suganthan, "Differential Evolution: A survey of the State-of-the-Art",

IEEE Transactions on Evolutionary Computation, Vol. 15, No. 1, pp. 4-31, 2011.

[21] Bernabe Dorronsoro and Pascal Bouvry, "Improving Classical and Decentralized Differential Evolution with new Mutation Operator and Population Topologies", *IEEE Transactions on Evolutionary Computing*, Vol. 15, No. 1, pp. 67-98, 2011.

[22] Ivan Zelinka, Vaclav Snasael and Ajith Abraham, "*Handbook of Optimization: From Classical to Modern Approach*", Springer, 2012.

[23] Yong Wang, Zixing Cai and Qingfu Zhang, "Differential Evolution with Composite Trial Vector Generation Strategies and Control Parameters", *IEEE Transactions on Evolutionary Computation*, Vol. 15, No. 1, pp. 55-66, 2011.

[24] Z. Hui-Fang, H. Wei and W. Jin-Song, "Clustering-Based Differential Evolution with Composite Trial Vector Generation Strategies and Control Parameters", *Proceedings of Joint International Conference on Service Science, Management and Engineering*. pp. 1-6, 2016.

[25] Kenneth Price, Rainer M. Storn and Jouni A Lampinen, "*Differential Evolution: A Practical Approach to Global Optimization*", Springer, 2006.