Turkish Journal of Electrical Engineering & Computer Sciences

http://journals.tubitak.gov.tr/elektrik/

Research Article

Turk J Elec Eng & Comp Sci
(2016) 24: 4623 – 4636

doi:10.3906/elk-1501-108

A problem approximation surrogate model (PASM) for fitness approximation in optimizing the quantization table for the JPEG baseline algorithm

Vinoth Kumar BALASUBRAMANIAN*, Karpagam MANAVALAN
Department of Computer Science and Engineering, PSG College of Technology, Coimbatore, Tamil Nadu, India

Received: 16.01.2015 • **Accepted/Published Online:** 21.08.2015 • **Final Version:** 06.12.2016

Abstract: The quantization table in the baseline Joint Photographic Experts Group (JPEG) algorithm plays an important role in compression/quality trade-off. Hence the detection of the optimal quantization table is viewed as an optimization problem. The genetic algorithm (GA) is an attractive optimization tool by many researchers for this application due to its ability in dealing with complex problems. In spite of its advantages, the GA requires more computation time to achieve an optimal solution if it has an expensive fitness evaluation.

This paper proposes a problem approximation surrogate model (PASM) for fitness approximation to assist the GA in optimizing the quantization table for a target bits per pixel. This proposal reduces the computational time of the GA without any loss in performance. The PASM uses an image block clustering process and an indirect evaluation method to approximate the fitness value. The number of clusters in the clustering process may influence the performance of the PASM. A performance analysis with different number of clusters has been done and a suitable cluster number is identified with the help of measuring criteria such as mean squared difference, correct selection, potentially correct selection, and rank correlation. In addition, the results acquired from these measuring criteria are confirmed using statistical hypothesis tests such as Friedman's ANOVA and Wilcoxon signed rank. The PASM with suitable cluster number has been tested in a classical genetic algorithm and knowledge based genetic algorithm. Several benchmark images with different complexity levels have been examined in three different target bits per pixel to validate the performance of the PASM. The results proved that the PASM guarantees better results in terms of peak signal-to-noise ratio with a reduction in computational time.

Key words: JPEG, quantization table, optimization, genetic algorithm, surrogate model, fitness approximation, problem approximation, ANOVA, Wilcoxon signed rank test

1. Introduction

Joint Photographic Experts Group (JPEG) is a widely used image compression standard on the web and in multimedia applications. The web statistics report reveals around 68.9% of images on the Internet are in JPEG format [1]. JPEG is a still-frame compression standard developed by CCITT with the collaboration of ISO in 1992. It supports four distinct modes of operation in algorithmic point of view: sequential discrete cosine transform (DCT), progressive DCT, hierarchical, and lossless. In sequential DCT based mode often called a JPEG baseline algorithm, the image is subdivided into 8×8 pixel blocks and they are processed from left to right and top to bottom. The forward DCT is applied to each block and 8×8 DCT coefficients are quantized by a quantization table. Finally the quantized DCT coefficients are entropy encoded and stored along with the

*Correspondence: vinki_ece@yahoo.co.in

quantization table to form the compressed file. During sequential JPEG decompression, the compressed file is decoded and the resultant DCT coefficients are dequantized by the quantization table. Finally, the inverse discrete cosine transform (IDCT) is applied to the dequantized DCT block to get the pixel values. A detailed description of a baseline JPEG encoder/decoder is available in [2].

The quantization table used in the JPEG scheme plays an important role in image quality/compression trade-off. The default quantization tables recommended by the JPEG standard do not provide the best trade-off for all images. Since the generation of the quantization table is viewed as an optimization problem, many researchers use metaheuristic techniques to optimize the quantization table, which guarantees the best trade-off. For optimizing the quantization table, a genetic algorithm (GA) seems to be a very appropriate choice by many researchers. In general, the GA requires a large number of generations to achieve a good solution, and if computationally expensive fitness evaluations are added to it, then it requires more computation time [3]. The solution to this problem is the use of a surrogate model, which can simulate the behavior of the original fitness function in the evolution cycle but can be evaluated much faster [4]. In the literature, fitness approximation is done in three ways [5] for different applications: i) problem approximation [6,7], replaces the original statement of the problem with its approximate one, which is easier to solve, ii) functional approximation [8–11], replaces the original objective function by an alternate and explicit expression, and iii) evolutionary approximation [12,13], estimates the fitness value of an offspring from their parents or from their fellow offspring. Although several surrogate models used in GAs are available in the literature, to the best of our knowledge they have been never used to optimize the quantization table for the JPEG baseline algorithm.

In this paper we propose a problem approximation surrogate model (PASM) to approximate the fitness value in a GA for optimizing the quantization table for the target bits per pixel. Our objective is to reduce the computational time of the GA without any loss in performance. The PASM separates all 8×8 blocks of an image into subgroups by K means clustering method, and one representative from each group is taken to represent an image. The measures that influence the fitness value are bits per pixel and mean squared error. These measure values are evaluated only for these representative blocks and the measure values of remaining blocks are estimated from these representative measure values of each group. This divide-and-conquer approach replaces the number of blocks to be processed with a smaller number of blocks, which in turn reduces the computational time drastically. However, k value may influence the performance of the PASM. Therefore, the performance of the PASM with different k values is examined by measures [14] such as MSD, CS, PCS, and RC; also the results are validated by a statistical hypothesis testing approach. The proposed PASM is introduced in both a classical genetic algorithm (CGA) and knowledge based genetic algorithm (KBGA) [1]; in addition, the results are compared in terms of peak signal-to-noise ratio (PSNR) to that of the same GAs without using the PASM. Furthermore, it has been found that, by using a PASM, CGA and KBGA are able to find an optimal quantization table in a reduced computational time and also it exhibits a better performance.

The remainder of this paper is organized as follows. In the next section, the optimization problem considered in this paper is explained. A brief review of CGA and KBGA is given in Section 3. Section 4 illustrates the proposed PASM. The validation measures are given in section 5. The experimental results are discussed in section 6. Concluding remarks are summarized in section 7.

2. The optimization problem

The optimization problem considered here is to find the quantization table that generates better image quality for the given bits per pixel. For any optimization problem, the fitness function is one that determines the

goodness of each chromosome. In this paper, an unfitness function is used instead of a fitness function to evaluate the survival probability of each chromosome. The unfitness function is shown in Eq. (1).

$$\xi = a\left(\frac{8}{B_r} - \lambda\right)^2 + \varepsilon, \tag{1}$$

where a = 10, B_r = bit rate, λ = desired compression ratio, and ε = mean squared error (MSE).

Here the constraint is incorporated into the unfitness function and the unfitness value depends on the difference between the actual and the desired compression ratio, in addition to the MSE of the decoded image.

3. Genetic algorithm

A genetic algorithm is a directed random search technique often applied to optimization in complex multidimensional search spaces. It belongs to the class of search methods that operate on a population of solutions for a problem and make it evolve by iteratively applying a set of stochastic operators. These operators, inspired by the natural evolution process, manipulate the individuals in a population to improve the solution. The modifications in these operators to suit a particular application may lead to the variants of GA. In this paper, two variants of GA, CGA and KBGA, are taken into consideration.

The main steps involved in the CGA process are initialization, evaluation, crossover, and mutation. Initialization is a first step where a population of candidate solutions is randomly generated for the given problem. Each candidate solution is called a chromosome and made up of genes. Here every quantization table, which is an 8 × 8 matrix, denotes a chromosome. Therefore, each chromosome has 64 genes. Each chromosome is evaluated by the unfitness function shown in Eq. (1), which is a measure of performance toward an objective. $(mu\ +\lambda)$-selection is used to select the chromosomes as parents for crossover and mutation. Crossover is an operation used to create two new offspring from two randomly selected parents. Mutation is an operation that changes a gene value in a chromosome randomly.

KBGA follows the same procedure as of CGA with modifications in genetic operators. KBGA uses knowledge based operators such as knowledge based initialization (KBI), knowledge based selection (KBS), knowledge based crossover (KBC), and knowledge based selective mutation (KBSM) to accelerate the search of optimal solution. KBGA incorporates the domain knowledge in genetic search to address the problems like uncertainty and low convergence speed of the GA in constrained optimization. A detailed description of KBGA is given in [1]. The algorithms for CGA and KBGA used in our work are shown in algorithms 1 and 2, respectively.

4. Problem approximation surrogate model (PASM)

As discussed in the introduction, the computational time of the GA process has become prohibitive due to expensive fitness value calculation. The unfitness function considered here includes the computation of measures like bits per pixel (B_r) and MSE (ε) for the given image. For each unfitness value evaluation, the whole JPEG compression and decompression process has to be executed to calculate the above said measures. Furthermore, the computational time taken for these measures depends on the number of 8 × 8 blocks present in an image. Thus the time taken for evaluating the unfitness value is directly proportional to the number of 8 × 8 blocks present in an image. In this paper, the PASM is used for fitness approximation, which reduces the total number of 8 × 8 blocks of an image to lower number to produce approximately the same unfitness value as the actual number of 8 × 8 blocks but in a shorter amount of time. It is well accepted that the natural images are generally

statistically self-similar, that is blocks of an image can be approximated using other blocks of the same image [15]. The PASM exploits the similarities between the 8 × 8 blocks of an image to reduce the number of 8 × 8 blocks for processing.

Algorithm 1: Classical Genetic Algorithm

Input: a gray scale image and desired compression ratio

Output: a best chromosome (Quantization Table)

Generate population of n chromosomes randomly;
Evaluate the chromosomes;
Select superior chromosomes based on low unfitness value;
While maximum generation not reached *do*
    Perform single point crossover on selected chromosomes;
    Evaluate the offspring;
    Select superior chromosomes from both parents and offspring;
    Perform mutation on selected chromosomes;
    Evaluate the offspring;
    Select superior chromosomes from both parents and offspring;
End while
Return best Chromosome;

Algorithm 2: Knowledge Based Genetic Algorithm

Input: a gray scale image, desired compression ratio and number of clusters

Output: a best chromosome (Quantization Table)

Generate population of n chromosomes randomly using KBI;
Evaluate the chromosomes;
Select superior chromosomes based on low unfitness value;
While maximum generation not reached *do*
    Select the chromosomes using KBS;
    Perform KBC on selected chromosomes;
    Evaluate the offspring;
    Select superior chromosomes from both parents and offspring;
    Perform KBSM on selected chromosomes;
    Evaluate the offspring;
    Select superior chromosomes from both parents and offspring;
End while
Return best Chromosome;

The construction of the PASM can be divided into two stages: (i) clustering of image blocks and (ii) evaluation of unfitness value. An image block clustering process adopted from [1] is used to cluster the image blocks. Clustering of image blocks is done in transform domain using the K means algorithm, which is chosen

for its simplicity and low computational cost. The image block clustering process uses a deterministic centroid initialization method (DCIM), which made the K means algorithm produce unique clustering results every time. The procedures of the image block clustering process and DCIM are given in algorithms 3 and 4, respectively.

Algorithm 3: Image Block Clustering Procedure

Input: a gray scale image
M (Number of Main Clusters)
N (Number of Subclusters)

Output: Cluster Representatives
Number of Image blocks in each cluster

Split an image into 8 × 8 nonoverlapping blocks.
Transform each 8 × 8 block of the image using discrete cosine transform (DCT)
Extract DC coefficients and Standard deviation of AC coefficients as features from each DCT block. *Store* them together as a vector.
Group the DC coefficients into M clusters (Main Clusters) using K Means algorithm.
Choose the initial centroids using DCIM.
Group the corresponding image blocks into M clusters based on DC coefficient clusters.
Group the Standard deviation of AC coefficients in the corresponding main clusters into N clusters (Subclusters) using K Means algorithm. *Choose* the initial centroids using DCIM.
Group the corresponding image blocks into N clusters based on Standard deviation of AC coefficients clusters.
Choose a block that is closest to the cluster center as cluster representative.
Return Cluster representatives and number of blocks in each cluster.

Algorithm 4: Deterministic Centroid Initialization Method

Input: a set of N data vectors (Data set)
K (Number of Clusters)

Output: Initial Cluster Centroids

Sort the data vectors in ascending order.
Split the ordered vectors into K bins randomly.
Calculate the mode for each bin and *consider* as initial centroids
Return initial centroids

A block that is closest to the cluster center is taken as a representative of that cluster; also MSE and B_r values can be calculated only for the representative blocks. Since these measures satisfy the additive property subject to suitable normalization [16,17], the MSE and B_r values calculated for the representative block are assumed for the remaining blocks in that cluster and they are added to form total MSE and B_r values of an image, which will be approximately equal to the actual MSE and B_r values. The algorithm for evaluation of unfitness value using the PASM is shown in algorithm 5.

Algorithm 5: Unfitness value evaluation using PASM

Input: Representative blocks;

8×8 Quantization tables;

Number of blocks in each cluster;

a = 10;

λ = desired compression ratio = $\left(\frac{8}{\text{target bits per pixel}}\right)$;

Output: Unfitness values

For all chromosomes *do*

Calculate MSE and B_r for each representative block using quantization table;

Assume same MSE and B_r for the remaining blocks in the cluster;

5. Performance measures

Analyzing the quality of the model is found essentially in surrogate modeling. Different performance measures considered here are adopted from [14] with small modifications in expectation parameter due to use of the (mu $+\lambda$)-selection, where mu represents the number of parents and λ represents the number of offspring.

5.1. Mean squared difference (MSD)

This is a commonly used measure that calculates the mean squared difference between the actual fitness value (ϕ^{actual}) and the model based approximated fitness value ($\phi^{approximate}$) of each chromosome. This measure is averaged over n different chromosomes taken into account in one generation, which is shown in Eq. (2).

$$MSD = \frac{1}{n}\sum_{j=1}^{n}\left(\phi_j^{(approximate)} - \phi_j^{(actual)}\right)^2 \tag{2}$$

5.2. Correct selection (CS)

The above measure mainly focuses on the accuracy of the surrogate model. However, the selection of right individuals for the next generation is also important in view of the evolutionary process. This measure is based on the number of chromosomes that have been correctly selected using the surrogate model. It is given in Eq. (3).

$$CS = \frac{\xi - \langle\xi\rangle}{\mu - \langle\xi\rangle}, \tag{3}$$

where ξ $(0 \leq \xi \leq \mu)$ represents the number of chromosomes that would also be selected if the fitness value evaluations were done without using a model. The expected value of $\langle\xi\rangle$ is given by $\left\lfloor\frac{\mu}{2}\right\rfloor$. This measure value can be positive or negative, where positive values indicate that more than $\left\lfloor\frac{\mu}{2}\right\rfloor$ chromosomes are selected correctly and negative values indicate that only less than $\left\lfloor\frac{\mu}{2}\right\rfloor$ chromosomes are selected correctly.

5.3. Potentially correct selection (PCS)

This measure indicates whether good or bad chromosomes have been selected. It includes the rank of the selected chromosomes, calculated based on the fitness function without using a model. It is given in Eq. (4).

$$PCS = \frac{\pi - \langle \pi \rangle}{\pi^{(max)} - \langle \pi \rangle}, \tag{4}$$

where

$$\pi^{(\max)} = \sum_{m=1}^{\mu} \lambda m, \langle \pi \rangle = \sum_{m=1}^{\lfloor \frac{\mu}{2} \rfloor} \lambda m$$

A grade λ – m is assigned to each chromosome, if it is an mth best individual selected based on the real fitness function. Then the grades of all selected chromosomes are summed to form π. The maximum π value indicates that all μ chromosomes are selected correctly.

5.4. Rank correlation (RC)

This measure gives the relation between the ranks of all chromosomes with and without a model. It is given in Eq. (5).

$$RC = 1 - \frac{6 \sum_{l=0}^{\lambda} d_l^2}{\lambda(\lambda^2 - 1)} \tag{5}$$

d_l is the difference between the ranks of the *lth* chromosome based on the original fitness function and on the approximate model. If the RC value is higher, then the relation between the ranks of all chromosomes with and without a model is stronger with a positive slope.

6. Experimental results

Although GA is a well-known effective approach for the generation of the quantization table, more effort is needed to reduce the computational time. The focus of this paper is to reduce the computational time of the GA using the PASM. The image block clustering process plays a vital role in PASM. The performance of the PASM is based on the number of clusters k. Generally the approximation error is taken into account to validate the performance of the model; at the same time, it could not be concluded that the model with low approximation error may do the optimization process correctly [14]. Here the approximation error is inversely proportional to the k value. Therefore, there is also a need to find the suitable k value based on not only approximation error but also on the evolutionary perspective. In this study, the k value is chosen as 75, 100, 125, and 150 based on the computational time taken to evaluate the unfitness function. In order to find the suitable k value in the PASM, it is integrated in the CGA process and the performances of the PASM with different k values are validated by four quality measures explained in section 5. These measures are calculated between the PASM assisted CGA and its counterpart CGA without the PASM. The experiments are carried out by considering $(mu + \lambda)$-selection with $mu = \lambda$ and $mu + \lambda$ chromosomes as the initial population in both cases.

The proposed PASM assisted CGA is realized using MATLAB R2008b and it is implemented on a Dell workstation of Intel Xenon CPU E3-1240 V3 @ 3.40 GHz processor with 16 GB of RAM. Benchmark images shown in Figure 1 are taken from the USC-SIPI image database and are of size 256 $\times$ 256 and digitized to

256 gray levels. As the model performance can be changed from one generation to the next, it is necessary to validate the performance of the PASM with different k values at multiple generations.

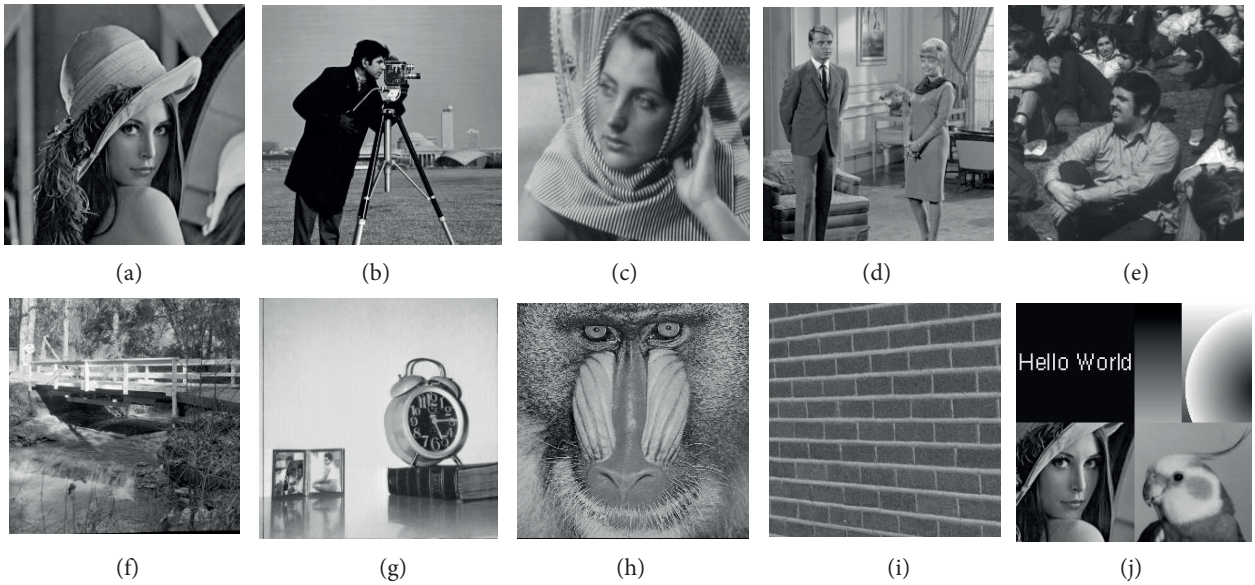

Figure 1. Benchmark test images: (a) Lena, (b) camera man, (c) Barbara, (d) couple, (e) crowd, (f) bridge, (g) clock, (h) baboon, (i) pattern, (j) montage.

The performance measures for the PASM with different k values have been calculated at each generation for each image. The programs are executed for 20 runs and the comparison graph has been drawn based on the average value of each measure at every five generations as shown in Figures 2–5. From the graphs, no clear conclusions can be drawn about which k value is better. This leads to the need for statistical analysis. Here the difference between the performances of the PASM with different k values needs to be analyzed at multiple time points. Thus Friedman's ANOVA test [18] is suited to this study, where it analyzes whether there is a significant difference in performance of different k values across multiple generations. As a result, a null hypothesis is made that there are no significant differences among different k values. Table 1 shows the P-value obtained from Friedman's ANOVA test with 0.05 at level of significance (α) for the above-mentioned performance measures. When a P-value is greater than the significance level, then the null hypothesis is not rejected; otherwise it is rejected. The P-value of all the measures is less than 0.05, which shows the rejection of the null hypothesis. Therefore, it can be observed that there is a significant difference among the PASM with different k values.

Table 1. Friedman's ANOVA test results for different performance measures.

Performance measures	SS	df	P-value	Significance level
Mean squared deviation	1000	799	1.22062e–079	0.05
Correct selection	1000	799	5.58068e–056	
Potentially correct selection	1000	799	1.03053e–058	
Rank correlation	1000	799	5.10723e–065	

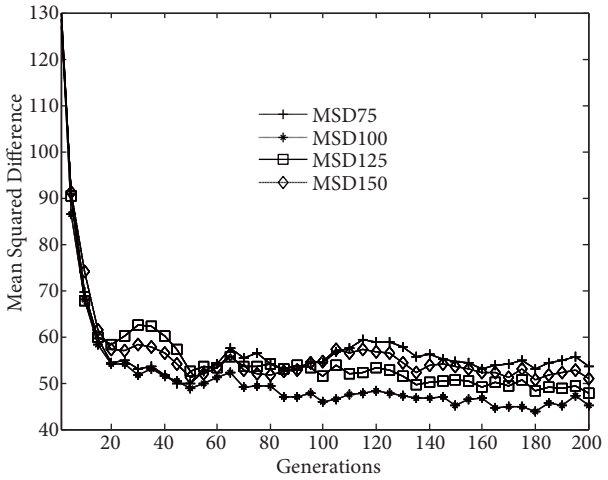

Figure 2. Mean squared difference criterion for different k values.

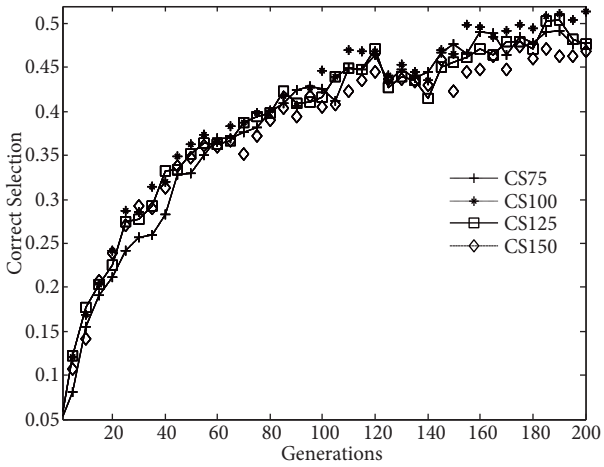

Figure 3. Correct selection criterion for different k values.

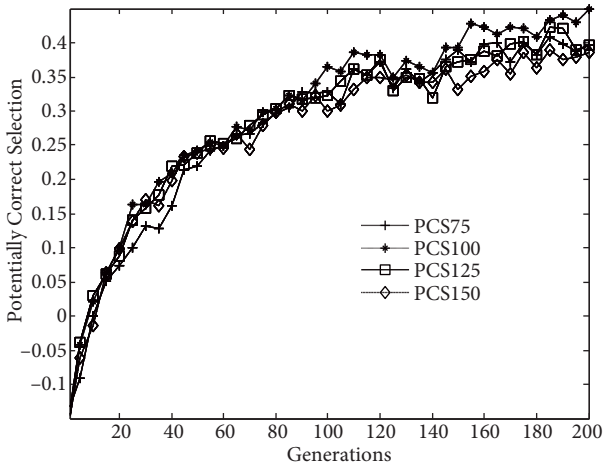

Figure 4. Potentially correct selection criterion for different k values.

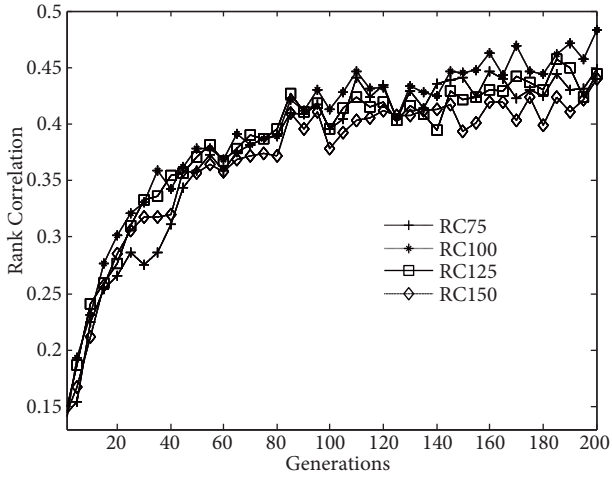

Figure 5. Rank correlation criterion for different k values.

After finding the significant differences between the k values, it is important to find the best k value. A nonparametric test (Wilcoxon signed rank test) [18] is performed to find the best k value among the considered k values. As a null hypothesis, it is assumed that there is no significant difference between paired k values, whereas the alternate hypothesis is that there is a significant difference between the paired k values at the 5% significance level. Based on the rankings, one of two signs ('<' and '>') is assigned for the comparison of the performance of any two k values in Table 2, where '<' shows the row k value is significantly worse than column k value and '>' shows the row k value is significantly better than the column k value. From Table 2, it can be clearly seen that $k = 100$ is better than other k values in all criteria. In the MSD criterion, $k = 75$ is significantly worse than $k = 150$; however, $k = 75$ performs significantly better than $k = 150$ in all other criteria. This confirms the statement given in [14] that the large approximation errors must not mislead the evolution process. From the above study, it can be concluded that the PASM with 100 clusters performs better than the PASM with other numbers of clusters.

Table 2. Wilcoxon signed rank test results for different performance criteria.

Metric	k value	100	125	150
MSD	75	$<$	$<$	$<$
	100	-	$>$	$>$
	125	-	-	$>$
CS	75	$<$	$<$	$>$
	100	-	$>$	$>$
	125	-	-	$>$
PCS	75	$<$	$<$	$>$
	100	-	$>$	$>$
	125	-	-	$>$
RC	75	$<$	$<$	$>$
	100	-	$>$	$>$
	125	-	-	$>$

A model management procedure is required to integrate the PASM into the GA process. Direct replacement with no evolution control is used as a model management procedure, where it directly replaces the exact unfitness value with its approximate value. An exact unfitness value is not at all used in the optimization process. The main difference between the GA and the PASM based GA is an evaluation of the unfitness value. In GA, an unfitness value is evaluated by the original unfitness function, whereas in PASM-GA, it is evaluated by the PASM based unfitness function. In addition, there is no difference in implementation style between the algorithms. As per the above study, the number of clusters in the PASM is set as 100 and the PASM-CGA is evaluated using a set of benchmark images shown in Figure 1. A performance comparison is made between the JPEG based, CGA based, and PASM-CGA based quantization tables using a performance measure: peak signal-to-noise ratio (PSNR). The parameters for both GA simulations are given in Table 3. Both simulations are terminated when they achieve a better result than a standard JPEG result of the corresponding bits/pixel. Random gene values of initial chromosomes are in the range of 1 to 256. The programs are executed for 20 runs for each image against each of the target bits per pixel: 0.75 and 1.0 and 1.5. Table 4 compares the performance of the PASM-CGA based quantization table with CGA based and default JPEG quantization tables for different bits per pixel. It displays the average results of both CGA and PASM-CGA from 20 independent runs. From Table 4, it is clearly shown that the PASM-CGA based quantization table yields similar results as a CGA and JPEG in terms of PSNR. Table 4 also shows that the CGA and PASM-CGA take 34,134.90 s and 19,209.92 s on average, respectively, to generate the optimal quantization table. It confirms that the PASM reduces the computational time of the CGA by 43.7%.

Table 3. CGA and PASM-CGA parameter settings.

Parameter	CGA	PASM-CGA
Population size	64	64
Crossover probability	0.9	0.9
Mutation probability	0.09	0.09
Number of independent runs	20	20

In order to validate the performance of the PASM with our previous results for the optimization of the quantization table, KBGA is taken into consideration. PASM based KBGA is implemented in the same programming environment with the same simulation parameter settings and evaluated using the same set of

Table 4. Comparison of JPEG, CGA, and PASM-CGA for different target bits/pixels.

Target bits/pixel		0.75			1			1.5		
Image	Quantization table	Bits/ pixel	PSNR in dB	CPU running time in seconds	Bits/ pixel	PSNR in dB	CPU running time in seconds	Bits/ pixel	PSNR in dB	CPU running time in seconds
Lena	JPEG	0.76	31.01	NA	1.03	32.81	NA	1.50	35.31	NA
	CGA	0.77	31.03	9316.40	1.00	32.95	26,447.66	1.51	35.35	85,489.58
	PASM-CGA	0.77	31.12	5328.60	1.02	33.01	13,646.94	1.52	35.57	45,229.50
Camera man	JPEG	0.75	29.95	NA	1.02	31.71	NA	1.51	34.68	NA
	CGA	0.76	29.96	9578.14	1.02	31.79	18,189.00	1.52	34.78	80,575.78
	PASM-CGA	0.77	30.05	5344.20	1.01	31.91	10,679.77	1.52	34.94	41,947.50
Barbara	JPEG	0.77	30.26	NA	1.01	31.94	NA	1.50	35.88	NA
	CGA	0.76	30.43	9297.76	1.02	32.28	9353.44	1.50	36.05	66,836.80
	PASM-CGA	0.77	30.54	5794.53	1.03	32.38	5920.53	1.52	36.19	33,031.71
Clock	JPEG	0.75	34.31	NA	1.00	36.51	NA	1.51	39.58	NA
	CGA	0.75	34.47	30,675.34	0.99	36.62	33,545.04	1.53	39.87	77,890.26
	PASM-CGA	0.76	34.54	15,972.62	1.01	36.85	18,199.11	1.53	39.95	44,718.75
Bridge	JPEG	0.75	26.19	NA	1.03	27.37	NA	1.61	29.37	NA
	CGA	0.77	26.35	9862.15	1.06	27.46	9331.68	1.58	29.96	82,678.04
	PASM-CGA	0.74	26.47	6663.12	1.07	27.55	5930.19	1.57	30.05	40,647.85
Couple	JPEG	0.75	31.21	NA	1.01	32.81	NA	1.50	35.27	NA
	CGA	0.75	31.28	23,146.17	1.02	32.96	25,642.00	1.52	35.35	71,574.03
	PASM-CGA	0.77	31.39	12,971.81	1.01	33.05	14,788.72	1.51	35.37	48,314.00
Crowd	JPEG	0.76	32.09	NA	1.01	33.81	NA	1.52	36.47	NA
	CGA	0.76	32.24	34,629.73	0.99	33.89	36,856.12	1.52	36.71	55,934.48
	PASM-CGA	0.75	32.56	17,427.20	1.02	34.05	20,178.93	1.52	36.92	30,869.89
Baboon	JPEG	0.75	22.10	NA	1.04	22.98	NA	1.54	24.68	NA
	CGA	0.77	22.14	9349.60	1.05	23.32	9434.88	1.49	24.82	9477.76
	PASM-CGA	0.77	22.22	5295.68	1.04	23.41	5445.97	1.47	24.85	5782.20
Pattern	JPEG	0.75	30.48	NA	1.01	31.34	NA	1.51	32.61	NA
	CGA	0.74	30.73	9158.56	1.00	31.64	9168.00	1.53	32.62	9281.60
	PASM-CGA	0.74	30.77	5289.41	0.97	31.71	5506.77	1.47	32.73	5665.00
Montage	JPEG	0.75	34.15	NA	1.00	36.85	NA	1.50	40.76	NA
	CGA	0.74	34.32	26,457.42	1.01	36.92	51,652.23	1.52	40.86	83,217.48
	PASM-CGA	0.75	34.55	16,389.61	1.02	37.10	29,164.18	1.52	41.01	54,279.31

NA: Not applicable

benchmark images as given in [1]. The simulation parameters for PASM-KBGA are shown in Table 5 and the simulation is terminated when it achieves a better result than the KBGA result of the corresponding bits/pixel. The programs are executed 20 times for each image against each of the bits per pixel 0.75, 1.0, and 1.5. Table 6 compares the performance of both KBGA and PASM-KBGA for different bits per pixel. The CPU running

times shown in Table 6 are the average results of 20 runs. It is observed that KBGA and PASM-KBGA take 18,080.52 s and 11,253.66 s on average, respectively, to generate the optimal quantization table. It confirms that the PASM reduces the computational time of the KBGA by 37.7%. From the above results it can be

Table 5. KBGA and PASM-KBGA parameter settings

Parameters	KBGA	PASM-KBGA
Initial population	100	100
Crossover probability	0.6	0.6
Mutation probability	0.015 to 0.093 for ranks 1 to 6 respectively	0.015 to 0.093 for ranks 1 to 6 respectively
Number of independent runs	20	20

Table 6. Comparison of KBGA and PASM-KBGA for different target bits/pixels.

Target bits/pixel		0.75			1.0			1.5		
Image	Quantization table	Bits/ pixel	PSNR in dB	CPU running time in seconds	Bits/ pixel	PSNR in dB	CPU running time in seconds	Bits/ pixel	PSNR in dB	CPU running time in seconds
Lena	KBGA	0.75	31.80	17,965.80	1.03	33.96	18,050.00	1.55	37.25	18,136.30
	PASM-KBGA	0.76	31.93	11,133.12	1.04	34.09	11,220.00	1.52	37.33	11,356.68
Camera man	KBGA	0.76	30.87	18,081.80	1.01	33.21	18,167.00	1.51	36.26	18,239.20
	PASM-KBGA	0.77	30.92	11,152.56	1.05	33.26	11,251.20	1.52	36.31	11,362.92
Barbara	KBGA	0.77	31.30	17,822.60	1.01	33.57	17,933.00	1.52	37.35	18,001.30
	PASM-KBGA	0.76	31.36	11,120.64	1.05	33.65	11,187.60	1.48	37.51	11,294.28
Clock	KBGA	0.75	35.45	18,011.30	1.00	37.35	18,093.00	1.51	41.43	18,179.80
	PASM-KBGA	0.78	35.55	11,053.32	0.99	37.42	11,174.40	1.53	41.53	11,279.76
Bridge	KBGA	0.75	26.58	18,088.50	1.04	28.03	18,132.00	1.60	30.27	18,118.90
	PASM-KBGA	0.76	26.62	11,181.84	1.02	28.15	11,277.60	1.60	30.43	11,349.96
Couple	KBGA	0.75	31.70	18,013.40	1.00	33.46	18,088.00	1.50	36.40	18,174.90
	PASM-KBGA	0.76	31.88	11,129.88	0.99	33.67	11,242.80	1.53	36.47	11,329.20
Crowd	KBGA	0.75	32.38	18,053.20	1.02	34.36	18,178.00	1.53	37.62	18,265.60
	PASM-KBGA	0.77	32.56	11,251.56	1.04	34.42	11,353.20	1.56	37.67	11,426.28
Baboon	KBGA	0.76	22.50	17,943.70	1.04	23.66	18,072.00	1.53	25.52	18,135.10
	PASM-KBGA	0.74	22.55	11,141.52	1.02	23.71	11,236.80	1.55	25.58	11,305.44
Pattern	KBGA	0.75	31.10	17,937.40	1.01	32.21	18,065.00	1.51	34.17	18,121.20
	PASM-KBGA	0.74	31.27	11,161.32	1.03	32.27	11,265.60	1.53	34.31	11,371.32
Montage	KBGA	0.75	35.76	18,037.50	1.01	37.96	18,115.00	1.50	43.42	18,195.20
	PASM-KBGA	0.77	35.98	11,262.48	1.05	38.12	11,322.00	1.54	43.52	11,414.52

observed that the introduction of the PASM in the GA leads to better results in a shorter time compared to its counterpart GA without the PASM.

7. Conclusion

In this paper, a PASM has been proposed to assist GAs for optimizing the quantization table in the JPEG baseline algorithm. The PASM used an image block clustering process and an indirect evaluation method to approximate the fitness value. The performance of the PASM depended on the number of clusters in the clustering process. An experimental analysis has been done on different numbers of clusters and a suitable cluster number is identified based on approximation error and evolutionary perspective. In addition, verification has been performed using Friedman's ANOVA and Wilcoxon signed rank tests. The proposed PASM is integrated in CGA and KBGA and their results are compared with its counterpart in terms of PSNR. The integrated PASM in GAs guarantees better results with a reduction in computational time. Furthermore, the proposed PASM can be applied to other population-based metaheuristics for quantization table optimization.

References

[1] Vinothkumar B, Karpagam M. Knowledge based genetic algorithm approach to quantization table generation for the jpeg baseline algorithm. Turk J Elec Eng & Comp Sci 2016; 24: 1615–1635.

[2] Wallace G. The JPEG still picture compression standard. IEEE T Consum Electr 1992; 38: 18-34.

[3] Bo L, Qingfu Z, Georges GEG. A gaussian process surrogate model assisted evolutionary algorithm for medium scale expensive optimization problems. IEEE T Evolut Comput 2014; 18: 180-192.

[4] Loshchilov IG. Surrogate-assisted evolutionary algorithms. PhD, Paris-Sud University, Paris, France, 2013.

[5] Shi L, Rasheed K. A survey of fitness approximation methods applied in evolutionary algorithms. In: Tenne Y, Goh CK, editors. Computational Intelligence in Expensive Optimization Problems. Berlin, Germany: Springer-Verlag, 2010. pp. 251-258.

[6] Laura AA, Goldberg DE. Efficient discretization scheduling in multiple dimensions. In: Genetic and Evolutionary Computation Conference; 9–13 July 2002; New York. NY, USA: Morgan Kaufmann. pp. 271-278.

[7] Bo Y, Xin Y, Bin L, Thomas W. A new memetic algorithm with fitness approximation for the defect-tolerant logic mapping in crossbar-based nano-architectures. IEEE T Evolut Comput 2013; 18: 846-859.

[8] Achille M, Anoop AM. A computationally efficient metamodeling approach for expensive multi objective optimization. Optim Eng 2008; 9: 37-67.

[9] Horng SC, Lin SY. Evolutionary algorithm assisted by surrogate model in the framework of ordinal optimization and optimal computing budget allocation. Inform Sciences 2013; 233: 214-229.

[10] Tawatchai K, Sujin B. Surrogate-assisted multi objective evolutionary algorithms for structural shape and sizing optimization. Math Probl Eng 2013; 2013: 1-13.

[11] Wang C, Duan Q, Gong W, Ye A, Di Z, Miao C. An evaluation of adaptive surrogate modeling based optimization with two benchmark problems. Environ Modell Softw 2014; 60: 167-179.

[12] Yoon JW, Cho SB. An efficient genetic algorithm with fuzzy c-means clustering for traveling salesman problem. In: IEEE 2011 Evolutionary Computation Congress; 5–8 June 2011, New Orleans, LA. New York, NY, USA: IEEE. pp. 1452-1456.

[13] Sun X, Gong D, Jin Y. A new surrogate-assisted interactive genetic algorithm with weighted semi supervised learning. IEEE T Syst Man Cy B 2013; 43: 685-698.

[14] Jin Y, Hüsken M, Sendhoff B. Quality measures for approximate models in evolutionary computation. In: Springer 2003 Genetic and Evolutionary Computation Conference; 9–11 July 2003; Chicago. Berlin Heidelberg, Germany: Springer. pp. 170-173.

[15] Ebrahimi M, Vrscay ER. Self-similarity in imaging 20 years after fractals everywhere. In: TICSP 2008 Local and Non-Local Approximation in Image Processing Workshop; 23–24 August 2008; Lausanne, Switzerland. Finland: TICSP. pp. 165-172.

[16] Taubman D. High performance scalable image compression with ebcot. IEEE T Image Process 2000; 9: 1158-1170.

[17] Zhou W, Bovik AC. Mean squared error - love it or leave it. IEEE Signal Proc Mag 2009; 26: 98-117.

[18] Smith JE, Clark AR, Staggemeier AT, Serpell MC. A genetic approach to statistical disclosure control. IEEE T Evolut Comput 2011; 16: 431-441.