JPEG Quantization Table Optimization by Guided Fireworks Algorithm

No Author Given

No Institute Given

Abstract. Digital images are very useful and ubiquitous, however there is a problem with their storage because of their large size and memory requirement. JPEG lossy compression algorithm is prevailing standard that solves that problem. It allows for different levels of compression (and corresponding quality) by quantization using recommended quantization tables. It is possible to optimize these tables for better quality at the same level of compression. This presents a hard combinatorial optimization problem for which stochastic metaheuristics proved to be efficient. In this paper we adjusted recent guided fireworks algorithm from the class of swarm intelligence algorithms for quantization table optimization. We tested the proposed approach on standard benchmark images and compared results with other approaches from literature. By using different image similarity metrics our approach proved to be more successful.

Keywords: image processing, JPEG algorithm, fireworks algorithm, swarm intelligence

1 Introduction

Widespread use of digital images facilitated advances in numerous scientific areas. Medicine, astronomy, biology and many other fields were significantly improved by digital images introduction. Common requirement in all these areas that use digital images is some kind of image processing which reduces to application of different algorithms and mathematical formulas to the matrix of numbers that represent digital image. Simplicity and power of digital image processing is one of the major factors that contributes to ubiquity of digital images.

Benefits of using digital images are numerous, however there are also some problems. One of the biggest problems is the space needed for saving digital images. One high quality digital image typically consists of millions of pixels and accordingly needs tens of megabytes of memory. One solution to that problem is to use compression techniques to record image data in some format that would use less memory. All compression algorithms can be divided into two groups: lossless and lossy algorithms. Lossless algorithms try to find some consistency and redundancy in the data and to rewrite in more compact but reversible way so that decompressed data are identical to the original [10]. Such algorithm can achieve typical compression ratios of 2:1 or 3:1 which is inadequate for huge digital images. Fortunately, lossy compression algorithms by discarding some informations which results in minimal quality loss, achieve but high compression rates of 10:1 or even 100:1.

JPEG is one of the most used lossy compression algorithm for digital images. It is well known that by using JPEG algorithm space needed for image information can be reduced twenty or even fifty times [4]. The main compression is done by quantization step, where based on quantization table less important information is neglected.

In this paper a recent and very successful swarm intelligence algorithm, guided fireworks algorithm, was used to solve combinatorial problem of selecting elements of quantization table for JPEG algorithm in order to optimize decompressed image according to some metrics.

In Section 2 steps of JPEG algorithm are described while in Section 3 use of quantization table in JPEG algorithm is briefly explained. In Section 4 guided fireworks algorithm is presented. Our proposed algorithm for quantization table selection is explained in Section 5. Experimental results are shown in Section 6 and at the end in Section 7 conclusion and suggestion for further work are given.

2 JPEG Algorithm

JPEG algorithm the most used lossy algorithm for digital image compression. It is a very powerful algorithm that can significantly reduce the size of the image file without any significant visible consequences.

JPEG algorithm consists of several steps and the first preprocessing step is based on the fact that human eye is less sensitive to changes in color than in the light intensity. HSV chromaticity components resolution reduction facilitates stronger compression for color images. JPEG algorithm is less suitable for the simple drawings (drawings with sharp edges or text). For such images even moderate compression will noticeably damage them.

The main step in the JPEG algorithm, the one that results in significant compression is quantization which is the subject of research and experiments in this paper. Some of the steps in JPEG algorithm will be just mentioned since they do not have impact on results of our proposed algorithm.

The first step, after preprocessing, is to transform image into frequency domain by applying the two-dimensional discrete cosine transform (DCT) which is performed on non-overlapping blocks of size 8×8 of light intensities.

Result of this step is that the image is transformed to blocks of size 8×8 where each block consists 64 frequency coefficient. The very first coefficient in the block represents average intensity and it is named DC component while the rest 63 coefficients are AC components. DC coefficient contain the most information about the block of the image. Coefficients close to DC represent low frequencies and coefficients closer to the lower-right corner represent high frequencies. Higher frequencies describe sudden changes in intensity values, i.e. edges and noise. These features are less important and usually these coefficients are close to zero.

The next step in JPEG algorithm is quantization where the main compression is done but also some information is lost.

3 Quantization Tables

Compression level and quality of compressed image in JPEG algorithm are mainly determined by the quantization table. The idea is to discard DCT coefficient that are less important and to reduce precision level for others. In this step, each DCT coefficient from 8×8 matrix is divided by corresponding element from the quantization table. By arranging elements in quantization table, different compression levels and qualities of compressed image can be determined. If all elements in quantization table are equal to one, then there will be no compression and the quality of the image will not be decreased.

JPEG provides recommendation for quantization table that was determined empirically based on human perception. Quantization table that provides high compression and good decompressed image quality is named Q_{50}. Based on this quantization table, tables Q_1 to Q_{100} can be calculated. The number in index represents the scale for quality of the compressed image. For low quality index higher compression will be achieved but at the expense of image quality. On the other hand, with high quality index better image quality will be achieved but the compression level will be low. For higher quality indices than 50 (less compression), quantization tables are obtained by multiplying the table Q_{50} by $(100 - quality\_level)/(50)$ while for lower quality inices but higher compression quantization tables are defined by multiplying standard Q_{50} by $50/quality\_level$. In both cases values are clipped to be between 1 and 255.

As mentioned before, these Q_i are quantization tables made based on human subjective opinion about image quality, but for many applications some more objective metrics are necessary. For such kinds of applications compressed images are further processed with some specific goal, and that goal achievement represents the quality metrics.

A method for determining customized JPEG quantization table for low bit rate mobile visual search was proposed in [3]. In the proposed method pairwise image matching precision was incorporated into distortion measure and quantization table was optimized to achieve better trade-off between compression level and visual quality.

In [1] an algorithm for finding the optimal quantization table that enables improvement of feature detection performance was proposed. Optimal quantization table was based on the measured impact that scale-space processing has on the DCT.

In [19] a comparison between compression by the traditional quantization matrix and by a set of quantization matrices especially optimized for ultrasound images was performed. Experimental results showed that images compressed by optimized tables have significantly better quality in the sense of the medical information.

Selecting elements of the quantization table represents a combinatorial problem: each of the 64 elements can be any number from some range. Theoretically, that range should be $[1, 1023]$, but in practice table elements are usually in the range $[1, 255]$. The only certain way to find the best quantization table is exhaustive search. However, that is not possible since the computational time is measured in hours even for 5 coefficients and then increased 255 times for each additional coefficient up to 64. For such hard optimization problems during last decades algorithms that imitate some natural processes were successfully used. Very promising branch of such algorithms are swarm intelligence algorithms that simulate simple individuals that collectively produce significant intelligence. Many different swarm intelligence algorithms were proposed so far and used for various purposes [2], [16], [15]. These algorithms were also used for JPEG quantization table optimization. In [6] genetic algorithm was used, while in [9] particle swarm optimization was used for optimizing quantization table. In [13] firefly algorithm was proposed to solve this combinatorial problem. In [14] a brief survey on swarm intelligence algorithms applied to JPEG algorithm was given. In this paper one of the latest swarm intelligence algorithm, fireworks algorithm, will be used for quantization table selection.

4 Guided Fireworks Algorithm

Guided fireworks algorithm (gFWA) is the latest improvement of the fireworks algorithm and it was proposed by Li, Zheng and Tan in 2016 [8]. The original fireworks algorithm (FWA) proposed in 2010 [11] simulates fireworks explosion with two different types of the fireworks. Well manufactured fireworks produce numerous sparks around explosion center which is used to define exploitation, while badly manufactured fireworks produce only a few sparks scattered in the space which represents exploration [11]. Since the initial version of fireworks algorithm was presented, several improved versions were proposed. The first modification was named enhanced fireworks algorithm where five modification of initial fireworks algorithm were introduced [17]. After enhanced FWA, cooperative FWA (CoFWA) was proposed in [18]. CoFWA enhanced the exploitation ability by using independent selection operator and increase the exploration capacity by crowdness-avoiding cooperative strategy among the fireworks. In [7] another two methods for improving exploration were proposed. The mechanism that allows FWA to dynamically adjust the number of sparks based on the fitness function results and the search results. The better diversity of the fireworks population was achieved by sharing the fitness information among the fireworks. This version of FWA is also known as the FWA with dynamic resource allocation (FWA-DRA). The latest version, guided fireworks algorithm will be briefly described.

During last few years fireworks algorithm was used as part of many different applications for solving hard optimization problems. It was used for SVM parameters tuning in [12] and in [5] it was used for parameter tuning of local-concentration model for spam detection.

Guided fireworks algorithm uses n fireworks and for each of them some number of sparks is generated. Fireworks and sparks represent points in d-dimensional space, where d is the dimension of the problem. The number of the sparks for each firework x_i is calculated as:

$$\lambda_i = \hat{\lambda} \frac{\max\limits_j (f(x_j)) - f(x_i)}{\sum_{j=1}^{n} (\max\limits_k (f(x_k)) - f(x_i))}, \tag{1}$$

where $\hat{\lambda}$ represents parameter that controls the overall number of sparks generated by the n fireworks, $y_{max} = \max(f(x_i)$ $(i = 1, 2, ..., n)$ represents the worst solution in the population and η is a small constant used to avoid division-by-zero error.

For each firework, explosion amplitude is defined by next equation:

$$A_i = \hat{A} \cdot \frac{f(x_i) - y_{min} + \eta}{\sum_{i=1}^{n} (f(x_i) - y_{min}) + \eta}, \tag{2}$$

where $\hat{A}$ defines the highest value of the explosion amplitude and $y_{min} = \min(f(x_i),$ $(i = 1, 2, ..., n)$ represents the best solution in the population of n fireworks.

In each generation, the firework with the best fitness is named core firework (CF). For CF, explosion amplitude was adjusted according to the following equation [8]:

$$A_{CF}(t) = \begin{cases} A_{CF}(1) & \text{if } f(X_{CF}(t)) = f(X_{CF}(t-1)), \\ C_r A_{CF}(t-1) & \text{if } f(X_{CF}(t)) = f(X_{CF}(t-1)), \\ C_a A_{CF}(t-1) & \text{if } f(X_{CF}(t)) < f(X_{CF}(t-1)) \end{cases} \tag{3}$$

where t represents the number of the current generation, while $C_a > 1$ and $C_r < 1$ are constants.

In each generation, a guiding spark (GS) is generated for each firework. The GS is generated by adding to the firework's position a guiding vector (GV). The position of the GS, G_i for firework X_i is determined by the next algorithm [8]:

Algorithm 1 Generating the Guiding Spark for X_i [8]

Require: X_i, s_{ij}, λ_i and σ
  Sort the sparks by their fitness values $f(s_{ij})$ in ascending order.
  $\Delta_i \leftarrow \frac{1}{\sigma\lambda_i}(\sum_{j=1}^{\sigma\lambda_i}) s_{ij} - \sum_{j=\lambda_i-\sigma\lambda_i+1}^{\lambda_i} s_{ij}$
  $G_i \leftarrow X_i + \Delta_i$
  return G_i

The guiding vector Δ_i is the mean of $\sigma\lambda_i$ vectors which is defined by the next equation:

$$\Delta_i = \frac{1}{\sigma\lambda_i}\sum_{j=1}^{\sigma\lambda_i}(s_{ij} - s_{i,\lambda_i-j+1}) \tag{4}$$

The complete guided fireworks algorithm is shown in next algorithm:

Algorithm 2 Guided fireworks algorithm [8]

```
Randomly initialize μ fireworks in the potential space.
Evaluate the fireworks' fitness.
repeat
   Calculate λ_i according to the Eq. 1
   Calculate A_i according to the Eq. 2 and Eq. 3
   For each firework, generate λ_i sparks within the amplitude A_i
   For each firework, generate guiding sparks according to previous algorithm.
   Evaluate all the sparks' fitness.
   Keep the best individual as a firework.
   Randomly choose other μ − 1 fireworks among the rest of individuals.
until termination criteria is met.
return  the position and the fitness of the best individual.
```

In this paper gFWA will be used for selecting coefficients in the quantization table.

5 The Proposed Algorithm

Compression level and obtained quality are mainly determined by the quantization table. In this paper the goal is to find equivalent compression level to some compression level achieved by using recommended quantization tables Q_i, but with the aim that image compressed by new quantization table has better quality. As mentioned, usually the quality is measured by human perception, but in many different applications some objective measurements are necessary, and human opinion has no value. We will use two different metrics that are used in literature to measure similarity of two images. The compressed image has better quality if it is more similar to the original one. For our proposed algorithm the goal is to get the best quality of the compressed image with some constraints, so the objective function for the optimization algorithm, gFWA, will be appropriate image similarity metrics.

The first standard metric for image similarity is mean square error (MSE) defined by:

$$MSE = \frac{1}{NM}\sum_{i=1}^{N}\sum_{j=1}^{M}(x_{i,j} - x'_{i,j})^2 \tag{5}$$

where $x_{i,j}$ represents the intensity value of the pixel (i, j) in the original image, and $x'_{i,j}$ represents the intensity value of the corresponding pixel in the compressed image. For two identical images, MSE is equal to zero since all the differences in the sum are zero which means that in the case when MSE is used as an objective function, the goal is to minimize it. Standard metrics based on MSE also used for image similarity is peak signal to noise ratio (PSNR). This metrics is defined by:

$$PSNR = 10 \log \frac{255^2}{MSE + \epsilon} \tag{6}$$

where ϵ is a small constant to prevent dividing by zero. PSNR is larger for more similar images.

The second metrics that will be used as objective function is normalized cross correlation (NK) that is defined as:

$$NK = \frac{\sum_{i=1}^{N} \sum_{j=1}^{M} x_{i,j} x'_{i,j}}{\sum_{i=1}^{N} \sum_{j=1}^{M} x_{i,j}^2} \tag{7}$$

For identical images NK is equal to 1 which is also a maximum possible value. Therefore, the objective is to minimize -NK.

When the metrics are established, gFWA should find elements of the quantization table so that the best quality of the image with some predetermined compression level is achieved. Since the elements of the quantization table need to be determined, they will represent input vector for the gFWA. Problem dimension is 64. Even though theoretically elements can be in range $[1, 1023]$, in practice, usually it is enough to set the range to $[1, 255]$ because DCT coefficients are rarely larger then 255.

Condition that some compression level should be achieved can be described in different ways. One measure of compression level could be the sum of all bits that are needed by all quantized non-zero DCT coefficients i.e. coefficient between 2^{k-1} and $2^k - 1$ requires k bits. This measure is not convenient since the number of required bits changes for each block. Element from the quantization table determines the maximum number of bits that is necessary for saving the quantized coefficient, but not the exact number. However, that is rectified by later Huffman so the sum of all the elements in the quantization table can appropriate measure of the level of coding. Larger sum corresponds to larger elements in the quantization table and consequently higher compression. In this paper this measurement was used as requirement that the sum of elements in the optimized quantization table be equal to the sum of elements in corresponding Q_i table.

In order to incorporate this condition, problem of selecting elements in the quantization table becomes a constrained optimization problem. The most difficult constraints are equality constraints where all feasible solutions are in one hyperplane since the search space is extremely reduced and it is difficult to find any feasible solutions. Equality constraints are usually relaxed by allowing some tolerance, larger in the beginning in order to find some feasible solutions and later dynamically reduced to zero. Deb's rules are used for comparison of feasible and non-feasible solutions.

In our case, equality constraint can be changed to inequality constraint that the sum elements in the optimized quantization table need to be larger or equal than the sum of elements in the corresponding Q_i table. This is possible because objective function is to have the best possible quality of the image, and the image will have better quality if more information is saved, i.e. DCT coefficients were divided by smaller numbers so the sum will be as small as possible. In this way objective function will force the solutions towards equality constraint and play the role of dynamic tolerance for equality relaxation.

In constraint optimization problems not all generated solution will be feasible. In this example, solutions where the sum of quantization table elements is less then given number are not acceptable. In order to guide optimization algorithm, Deb's rules are usually used. Between feasible and non-feasible solution, feasible solution is better regardless of the value of the objective function. Between two non-feasible solution, better is the one that has less constraint violation. For two feasible solution, value of the objective function is used to determine the better one. In this paper, gFWA was modified so that non-feasible solutions were discarded immediately after they were generated, without computing objective function value which is computationally very expensive operation (compression and decompression of the whole image). This was possible since the constraint was not on objective function but on the property of input vector (bound-constrained optimization).

Another adjustment in the proposed algorithm deals with the fact that elements of the quantization table are integers while gFWA works with real numbers. We performed optimization with the standard gFWA that generates real number solutions and rounded them to the nearest integer. That is possible since intensities as well as DCT coefficients are originally real numbers, artificially converted to integers because of our conventions for storing digital images.

6 Experimental Results

Our proposed algorithm was implemented in Matlab version R2016b. All experiments were performed on Intel ® Core^{TM} i7-3770K CPU at 4GHz, 8GB RAM computer with Windows 10 Professional OS.

Performance of our proposed algorithm was tested on several standard test images. Experimental results are shown for image "Lena" (Fig. 1), gray version, resolution 512×512. Experiment is shown for the level of compression where the degradation of image quality is easily visible. For that purpose we selected recommended Q_{10} table and by using gFWA we computed optimized quantization table Q_{10_opt} that achieves the same compression level. JPEG recommended table Q_{10} and our optimized Q_{10_opt} are shown in Table 1.

It can be seen that these tables are rather different even though their level of compression is very similar. Elements under and on anti-diagonal are all 255 in both cases. The difference is in elements above anti-diagonal. Sum of the elements in Q_{10} in 12,610 while the sum of elements in our Q_{10_opt} is larger, 12,647. This means that the compression level is slightly larger then by Q_{10}

Table 1. Quantization table Q_{10} (left) and Q_{10_opt} optimized by gFWA (right)

$$\begin{vmatrix} 80 & 55 & 50 & 80 & 120 & 200 & 255 & 255 \\ 60 & 60 & 70 & 95 & 130 & 255 & 255 & 255 \\ 70 & 65 & 80 & 120 & 200 & 255 & 255 & 255 \\ 70 & 85 & 110 & 145 & 255 & 255 & 255 & 255 \\ 90 & 110 & 185 & 255 & 255 & 255 & 255 & 255 \\ 120 & 175 & 255 & 255 & 255 & 255 & 255 & 255 \\ 245 & 255 & 255 & 255 & 255 & 255 & 255 & 255 \\ 255 & 255 & 255 & 255 & 255 & 255 & 255 & 255 \end{vmatrix} \qquad \begin{vmatrix} 7 & 5 & 9 & 2 & 18 & 226 & 231 & 255 \\ 26 & 17 & 35 & 68 & 177 & 254 & 255 & 255 \\ 8 & 35 & 15 & 84 & 252 & 255 & 255 & 255 \\ 118 & 172 & 244 & 247 & 255 & 255 & 255 & 255 \\ 243 & 250 & 252 & 255 & 255 & 255 & 255 & 255 \\ 138 & 201 & 255 & 255 & 255 & 255 & 255 & 255 \\ 133 & 255 & 255 & 255 & 255 & 255 & 255 & 255 \\ 255 & 255 & 255 & 255 & 255 & 255 & 255 & 255 \end{vmatrix}$$

table. This is due to the constraint that the sum has to be larger or equal to the sum of elements in Q_{10} table. Even with larger sum in quantization table, i.e. higher compression level, better quality of decompressed image was achieved because elements of quantization table are more appropriate. Resulting images are shown in Fig. 2. By visual inspection it is easy to see that the quality image in Fig. 2(b) is much better. Image is smoother and block edges are not visible like in the image compressed by JPEG recommended Q_{10} table (Fig. 2(a)).

Fig. 1. Original image

Besides perceptual results, numerical results are also better. When Q_{10} was used for JPEG compression of "Lena" the value for MSE was 59.3049 while PSNR was 30.3999 dB. The resulting image is shown in Fig. 2(a). When optimized Q_{10_opt} quantization table was used MSE dropped to 32.1913 which is significantly less then in the previous case. Corresponding PSNR raised to 33.0534 dB which is again better. Fig. 2(b) shows the resulting image when optimized quantization table was used.

(a) (b)

Fig. 2. Decompressed image by (a) Q_{10} and (b) quantization table Q_{10_opt} obtained by gFWA

In Table 2 are shown obtained metrics for three different test images. As it can be seen, our proposed algorithm successfully selected elements of quantization table so that the quality of the images was significantly improved for the same compression level.

Table 2. Experimental results obtained by quantization tables optimized by gFWA

Image	MCE		PSNR		NK	
	Q_{10}	Q_{gFWA}	Q_{10}	Q_{gFWA}	Q_{10}	Q_{gFWA}
Lena	59.3049	32.1913	30.3999	33.0534	0.9999	1.0000
Barbara	175.1430	129.2543	25.6969	27.0164	0.9997	0.9999
Boat	99.9588	61.7657	28.1326	30.2233	0.9979	0.9996

In [13] average pixel intensity distance between the original and compressed image was used as similarity measure. For images that were using Q_{10} table average pixel intensity distance was 5.886, while for optimized quantization table it was 5.1. For quantization table obtained by our proposed algorithm, average pixel intensity level was 4.085 which is better.

7 Conclusion

In this paper an algorithm for JPEG quantization table selection was proposed. For selecting elements in quantization table novel swarm intelligence algorithm, guided fireworks algorithm, was used. Quantization table elements were optimized so that desired compression level was achieved while the quality of the image was maximized according to selected metrics. For quality measurement two standard metrics were used, mean square error and normalized cross correlation, while peak signal to noise ratio was also used. Our proposed algorithm significantly improved quality of the compressed image. In further work more similarity metrics that are adjusted for specific applications can be used.

References

1. Chao, J., Chen, H., Steinbach, E.: On the design of a novel JPEG quantization table for improved feature detection performance. In: IEEE International Conference on Image Processing. pp. 1675–1679 (Sept 2013)
2. Dorigo, M., Gambardella, L.M.: Ant colonies for the travelling salesman problem. Biosystems 43(2), 73–81 (July 1997)
3. Duan, L.Y., Liu, X., Chen, J., Huang, T., Gao, W.: Optimizing JPEG quantization table for low bit rate mobile visual search. In: Visual Communications and Image Processing. pp. 1–6 (Nov 2012)
4. Gupta, M., Garg, A.K.: Analysis of image compression algorithm using DCT. International Journal of Engineering Research and Applications (IJERA) 2(1), 515–521 (2012)
5. He, W., Mi, G., Tan, Y.: Parameter optimization of local-concentration model for spam detection by using fireworks algorithm. Advances in Swarm Intelligence, LNCS 7928, 439–450 (2013)
6. Kumar, B.V., Karpagam, M.: Differential evolution versus genetic algorithm in optimising the quantisation table for JPEG baseline algorithm. International Journal of Advanced Intelligence Paradigms 7(2), 111–135 (2015)
7. Li, J., Tan, Y.: Enhancing interaction in the fireworks algorithm by dynamic resource allocation and fitness-based crowdedness-avoiding strategy. In: IEEE Congress on Evolutionary Computation (CEC). pp. 4015–4021 (July 2016)
8. Li, J., Zheng, S., Tan, Y.: The effect of information utilization: Introducing a novel guiding spark in the fireworks algorithm. IEEE Transactions on Evolutionary Computation 21(1), 153–166 (Feb 2017)
9. Ma, H., Zhang, Q.: Research on cultural-based multi-objective particle swarm optimization in image compression quality assessment. Optik-International Journal for Light and Electron Optics 124(10), 957–961 (2013)
10. Starosolski, R.: New simple and efficient color space transformations for lossless image compression. Journal of Visual Communication and Image Representation 25(5), 1056–1063 (2014)
11. Tan, Y., Zhu, Y.: Fireworks algorithm for optimization. Advances in Swarm Intelligence, LNCS 6145, 355–364 (June 2010)
12. Tuba, E., Tuba, M., Beko, M.: Support vector machine parameters optimization by enhanced fireworks algorithm. In: International Conference in Swarm Intelligence, pp. 526–534. Springer (2016)
13. Tuba, M., Bacanin, N.: JPEG quantization tables selection by the firefly algorithm. In: International Conference on Multimedia Computing and Systems (ICMCS). pp. 153–158. IEEE (2014)

14. Viswajaa, S., Kumar, V., Karpagam, G.R.: A survey on nature inspired metaheuristics algorithm in optimizing the quantization table for JPEG baseline algorithm. International Advanced Research Journal in Science, Engineering and Technology 2(4), 114–123 (2015)
15. Yang, X.S.: Firefly algorithms for multimodal optimization. Stochastic Algorithms: Foundations and Applications, LNCS 5792, 169–178 (2009)
16. Yang, X.S., Deb, S.: Cuckoo search via Levy flights. In: World Congress on Nature Biologically Inspired Computing. pp. 210–214 (Dec 2009)
17. Zheng, S., Janecek, A., Tan, Y.: Enhanced fireworks algorithm. In: 2013 IEEE Congress on Evolutionary Computation. pp. 2069–2077 (June 2013)
18. Zheng, S., Li, J., Janecek, A., Tan, Y.: A cooperative framework for fireworks algorithm. IEEE/ACM Transactions on Computational Biology and Bioinformatics PP(99), 1–1 (2016)
19. Zimbico, A., Schneider, F., Maia, J.: Comparative study of the performance of the JPEG algorithm using optimized quantization matrices for ultrasound image compression. In: 5th ISSNIP-IEEE Biosignals and Biorobotics Conference: Biosignals and Robotics for Better and Safer Living (BRC). pp. 1–6 (May 2014)