ISSN 1846-6168 (Print), ISSN 1848-5588 (Online)
https://doi.org/10.31803/tg-20220329112403

Original scientific paper
Received: 2022-03-29, Accepted: 2022-04-01

# A Lévy Flight Based BAT Optimization Algorithm for Block-based Image Compression

Ilker Kilic

**Abstract:** Many metaheuristics have been adopted to solve the codebook generation problem in image processing. In this paper, the Bat algorithm is combined by the Lévy flight distribution to find out the global optimum codebook. The Lévy flight distribution is combined by the local search procedure. Therefore most of the time the bat concentrate on the local area for specific food while it rarely flies to the different parts of the field for better food opportunities. This process strongly guides the bat on the global minimum way and offers better food, then the bat flies to that direction. Consequently, if a bat is captured by a local minimum point accidentally, the Lévy flight step provides a chance to escape from it easily. Numerical results suggest that the proposed Lévy flight based Bat algorithm is better than the classical ones and provides the global optimum codebook for image compression.

**Keywords:** Bat algorithm; image compression; Lévy distribution; metaheuristic algorithm; vector quantization

## 1 INTRODUCTION

Vector quantization (VQ) is a block-based lossy image compression technique combined with a codebook generation algorithm [1-4]. A codebook contains vector blocks that represent the image best. The LBG [5] technique is a well-known codebook generation algorithm that satisfies local minimum solutions.

On the other hand, different metaheuristic algorithms have been developed and combined by VQ to overcome the optimum codebook generation problem. The optimum codebook is produced by the Genetic algorithm (GA) based on Principle Component Analysis (PCA) conversion [6]. The Particle Swarm Optimization (PSO) technique is improved by the advantages of the adaptive fuzzy inference method for codebook generation [7]. Honey Bee Mating Optimization (HBO) [8] algorithm is adapted to obtain the near-global optimal codebook. Improved Differential Evolution (IDE) [9] technique, which is an improved version of the traditional DE with modifications in the scaling factor and the boundary control mechanism, is used for codebook design. Ant Colony Optimization is redesigned for VQ by decreasing both computation number and the time required [10]. Firefly algorithm (FRA) is adapted to LBG and achieved faster and higher quality reconstructed images than PSO, HBO and LBG [11]. Recently Cuckoo search algorithm (CSA) is generated for optimizing the LBG codebook [12]. The Cuckoo search algorithm spends 25% of convergence time for local search while it spends 75% of convergence time for the global optimum solution. The Bat algorithm (BAT) is a quite new optimization technique that is used for optimum solutions [13-18]. Lévy flight distribution is combined with many optimization algorithms for global optimization [19-35].

In this paper, the classical BAT metaheuristic technique is improved by Lévy flight distribution to solve the codebook generation problem. When the bats are searching for food by locally near to the global solution, the Lévy flight mechanism helps the bat both focus on the local and random search at the same time. Lévy flight food search deals with the local search mostly and random search rarely.

## 2 RELATED ALGORITHMS

In this section, LBG algorithm of VQ and the Lévy algorithm is presented briefly.

### 2.1 LBG Algorithm and Vector Quantization

Vector quantization (VQ) is a block-based lossy image compression technique. The codebook production is the fundamental process of VQ. Let input image $Y=\left\{x_{ij}\right\}$ be $N\times N$ pixels size. The original image is formed by sub-blocks with the size of m×m pixels. The dimension of the sub-blocks are defined by $N_b=\left(\frac{N}{m}\times\frac{N}{m}\right)$ and represented by a collection of original image vectors expressed as $X=\left\{x_i,\ i=1,2,...,N_b\right\}$. Let the variable $L$ be assumed as $m\times m$ pixel size. Then the original sub-blocks $x_i$ are determined as $L$ dimensional Euclidean space $x_i\in\Re_L$. The number of $N_c$ codewords belong to codebook named $C=\left\{C_1,C_2,...,C_{N_c}\right\}$, $j=1,2,...,N_c$. The original image vectors are determined by row vectors of $x_i=\left(x_{i1},x_{i2},...,x_{iL}\right)$ and the $i^{\text{th}}$ code word is defined as $c_i=\left(c_{i1},c_{i2},...,c_{iL}\right)$.. The codeword which has the minimum error is assigned to each original image block. So, the block number from the codebook list is used instead of the real image block to obtain a lossy image compression. Composing the vectors of $C$ is achieved by minimizing the mean square error (MSE) formulation Eq. (1) to Eq. (3).

$$MSE(C)=\frac{1}{N_b}\sum_{j=1}^{N_c}\sum_{i=1}^{N_b}\mu_{ij}\ ||\ x_i-c_j\ ||^2 \tag{1}$$

$$\sum_{i=1}^{N_C}\mu_{ij}=1\ ,\ i\in\left\{1,2,...,N_b\right\} \tag{2}$$

$$\mu_{ij} = \begin{cases} 1, \text{ if } x_i \text{ is in the } j^{\text{th}} \text{ cluster} \\ 0, \text{ otherwise} \end{cases} \tag{3}$$

The Euclidean distance between the $x_i$ image block and codeword $c_j$ is defined by $\| x_i - x_j \|$. The LBG algorithm executes the two rules given by Eq. (4) and Eq. (5). The group of image vectors $(R_j, j = 1, 2, ..., N_c)$ must satisfy

$$R_j \supset \left\{ x \in X : d\left(x, c_j\right) < d\left(x, c_k\right), \forall k \neq j \right\} \tag{4}$$

The center of the named as $c_j$ is calculated as

$$c_j = \frac{1}{N_j} \sum_{i=1}^{N_j} x_i, \ \ x_i \in R_j \tag{5}$$

Here, $N_j$ is the total number of elements that belong to $R_j$. Assume that the image vectors are $x_i$, $i = 1, 2, ..., N_b$ Euclidean distance is $d$, and initial codewords are $c_j(0)$, $j = 1, 2, ..., N_c$. Then the LBG technique achieves the following three steps to obtain the local optimal codebook;

Divide the original image blocks into several clusters using Euclidean distance equation. The center of clusters is saved in an indicator matrix named $\boldsymbol{U}$ size of $N_b \times N_c$ pixels.

$$\mu_{ij} = \begin{cases} 1, \text{ if } d\left(x_i, c_j(k)\right) = \min\left(x_i, c_j(k)\right) \\ 0, \text{ otherwise} \end{cases} \tag{6}$$

Determine the new cluster centers by Eq. (7).

$$c_j(k+1) = \frac{\sum_{i=1}^{N_b} \mu_{ij} \, x_i}{\sum_{i=1}^{N_b} \mu_{ij}}, \ j = 1, 2, ..., N_c \tag{7}$$

Until there is no change of centroid $c_j$ value execute the Eq. (6) and Eq. (7) sequentially.

### 2.2 BAT Algorithm

The bat-inspired metaheuristic algorithm is produced by Xin-She Yang [13] based on imitation of the echolocation feature of microbats. The microbats use echolocation between 25 kHz to 125 kHz, which means 2 mm to 14 mm wavelengths in the air. Microbats searches prey using this echolocation feature. In the search phase microbats emit short pulses, but when they approach a potential prey their pulse emitting frequencies increase, wavelengths of echolocation become shorter. Therefore, the microbats increase the accuracy of the prey location. The standard bat algorithm uses three rules; i. All microbats use echolocation and detect the distance to the potential food. ii. Microbats fly randomly with the parameters of $V_i$ velocity, $X_i$ position, $F_i$ frequency, $\lambda_i$ wavelength and $A_i$ loudness. They can adjust the rate of pulse emission $r \in [0, 1]$ according to the distance to the food. iii. The loudness varies from $A_{\min}$ to $A_0$ a microbat has two flying types which are local search or generation of or random search.

Generation of new solutions;

When a bat begins to search for prey it flies randomly following the three steps defined by Eq. (8) to Eq. (10). Here the frequency, velocity and position of the bats are updated.

$$F_i^t = F_{\min} + (F_{\max} - F_{\min}) u_i \tag{8}$$

$$V_i^t = V_i^{t-1} + (X_i^{t-1} - X_g) F_i^t \tag{9}$$

$$X_i^t = X_i^{t-1} + V_i^t \tag{10}$$

In the equations above, the frequency boundaries are defined as ($F_{\min}$, $F_{\max}$) = (0 Hz, 1 Hz), the uniform random variable $u_i \in [0, 1]$. $F_{\min}$ and $F_{\max}$ are the user-defined parameters and can be changed according to optimization problem type. $X_i$ and $V_i$ are the positions and the velocities of a bat, where the $X_g$ denotes the global best position. The frequencies of the microbats are determined randomly at the beginning of the algorithm.

Local flying;

When a bat detects a prey specifically then the local flying strategy is used. This procedure is determined by Eq. (11) and Eq. (12).

$$\text{if } (U_k \geq R_k), \text{ then } X_i^t = X_i^{t-1} + \varepsilon_k A_{ort}^{t-1} \tag{11}$$

$$\text{if } (U_k < R_k), \text{ then } X_i^t = X_i^{t-1} \tag{12}$$

Here $U_k \in [0, 1]$ and $\varepsilon_k \in [-1, 1]$ are uniform random variables, $A_{ort}$ is the average loudness of all the bats and $R_k$ is the pulse rate of the $k^{\text{th}}$ bat. Once a bat found its prey, the loudness usually decreases and the rate of pulse emission increases in order to determine true location of the prey. Therefore Eq. (12) is used more frequently than Eq. (11) when the bat is close to its prey.

Parameter updating;

After a local or random search flying is completed the pulse rate $R_k$ and the loudness $A_i$ parameters of the bats are updated using Eq. (13) and Eq. (14).

$$A_i^{t+1} = \alpha A_i^t \tag{13}$$

$$R_i^{t+1} = R_{\max} (1 - 0.5 \gamma^t) \tag{14}$$

Here α is similar to the cooling factor of the cooling schedule in the simulated annealing [16]. Therefore both alpha and gamma parameters are changed slowly depending on the iteration numbering which is shown in Fig. 1. Therefore both of them are determined as 0.997 for all iterations of the codebook generation method. The loudness reaches zero and the pulse rate variable reaches the final value of one by the end of the iteration number as shown in Fig. 1. The procedure of updating velocities and positions of the BAT algorithm is similar to the PSO algorithm. The frequency fi controls the step and range of the bat movement. The local search is controlled by the loudness and the pulse rate variables.

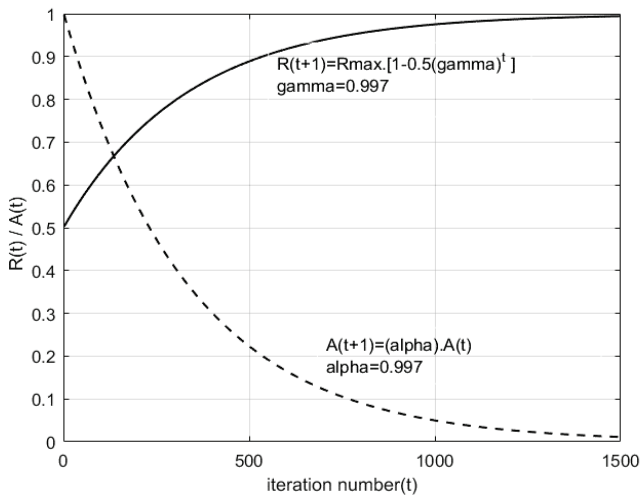



**Figure 1** Loudness and pulse rate distribution according to the iteration number *t*

## 3 LEVY FLIGHT BASED BAT ALGORITHM

The BAT optimization contains two search mechanisms. The first one is the generation of the solution step. The other one is the local search focusing on the pray. If the bat is not near to the global optimum solution when it is on the random search, the equations of Eq. (9) and Eq. (10) may cause the bat captured by a local minimum. The local search performed by Eq. (11) may not provide an approximation step in the correct scale [17]. Therefore the standard BAT formulations are reorganized and simplified [18]. The first modification is a simplification of the bat location. The Eq. (15) is generated instead of Eq. (9) and Eq. (10). In Eq. (10) the velocity of the bat is disregarded.

$$X_i^{t+1} = X_i^t + (X_g - X_i^t)F_i^t \tag{15}$$

Adoption of Eq. (15) for updating the bat location instead of Eq. (9) and Eq. (10) allows the bats to fly closer to the prey. On the other hand Eq. (15) decreases the ability of the bat population to visit the around effectively. Therefore a Lévy flight distribution is added to the local search procedure which is given by Eq. (16).

$$X_i^{t+1} = X_i^t + (X_g - X_i^t)F_L(X) \tag{16}$$

Here $F_L(X)$ represents the Levy function, whose values are small at big $X$ values. On the contrary the magnitude of $F_L(X)$ is relatively big at small $X$ values. Therefore the bats search is close to global solution in most of the local search iterations. On the other hand the bats rarely fly to far away with the help of Levy flight in order to escape from possible local minimum location. Let $F(X)$ be a normal distribution.

$$F(X) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \tag{17}$$

Let $\mu = 0$ and standard deviation $\sigma$ be a function of $x$.

$$\sigma = \sqrt{\frac{x^3}{c}} \tag{18}$$

Replacing Eq. (18) into Eq. (17), normal distribution is converted to the Lévy distribution which is given by Eq. (19).

$$F_L(X) = \sqrt{\frac{c}{2\pi}} \cdot \left( \frac{e^{-\frac{c}{2x}}}{x^{3/2}} \right) \tag{19}$$

Exponential term decreases to one by increasing $x$, and the Lévy distribution becomes like a power law function with long tail, which has an infinite variance with an infinite mean.

$$F_L(X) \cong k \cdot x^{-\lambda} \tag{20}$$

The approximated Lévy function which is expressed by Eq. (20) is plotted in Fig. 2. The approximated Lévy distributions are shown for the values of $\lambda_1 = -1.0$, $\lambda_2 = -1.25$, $\lambda_3 = -1.5$, $\lambda_4 = -2.0$. The maximum value of $F_L(X)$ is determined by $k = 2.0$ value.

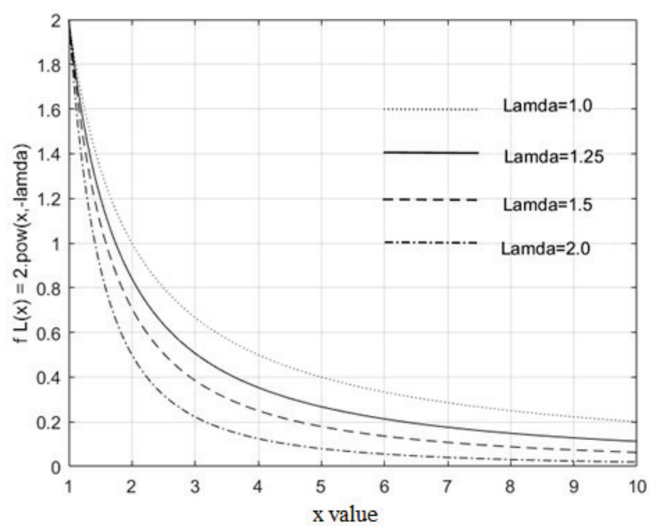



**Figure 2** Approximated Lévy functions for different $\lambda$ values

Lévy Flight based Bat algorithm (LBAT) can be obtained by using Lévy distribution function $F_L(X)$ in the bat random search procedure which is defined by Eq. (21). Therefore, if Eq. (20) is embedded into Eq. (16) the bat movement is determined as

$$X_i^t = X_i^{t-1} + k(X_g - X_i^{t-1})(random\ x)^{-\lambda} \tag{21}$$

where the *random x* is uniformly distributed in [0, 1], $k$ and $\lambda$ are the user defined coefficients. Since the term $k(random\ x)^{-\lambda}$ is power law function, in most of the iterations its value is small whereas the value is rarely relatively big. Consequently, the second term of Eq. (21) is called Lévy Flight step guides to the bat escape from local minimum points. The variable lambda is quite sensitive in Eq. (21) and must be properly tuned. The lambda is selected as $\lambda = 2.0$ for searching the best codebook. The pseudo code of the proposed LBAT is given below.

Other modifications of LBAT are about the pulse rate $R_i$ and loudness $A_i$. The original pulse rate equation of Eq. (14) and loudness equation of Eq. (13) are redesigned as in Eq. (22) and Eq. (23) respectively [18]. Instead of exponential function the $R_i^t$ and $A_i^t$ variables are changed into linear ones depending on the iteration number.

$$R_i^t = (R_0 - R_\infty)\left[\frac{t - t_{\max}}{1 - t_{\max}}\right] + R_\infty \tag{22}$$

$$A_i^t = \left(A_0 - A_\infty\right)\left[\frac{t - t_{\max}}{1 - t_{\max}}\right] + A_\infty \tag{23}$$

Here, $t_{\max}$ is the maximum iteration number while the notations 0 and $\infty$ corresponds the initial and the last values respectively. As previously mentioned, the resulting algorithm is named as LBAT after all these modifications. The pseudo code of the proposed LBAT is given below.

**begin**
Objective function *MSE*(*C*) by Eq. (1)
Generate initial bat population $C = \left\{C_1, C_2, ..., C_{N_C}\right\}$, $C_j \in \Re$, $j = 1, 2, ..., N_C$
Define the Lévy Flight variables $\lambda$, $k$
Define the initial and the end values of pulse rate $R_i$ and loudness $A_i$
Define iteration number $n$
**while** ($n$ < Max Iteration Number)
Define uniform random numbers $U_i \in [0, 1]$
If ($U_i < R_i$) Local search by Levy Flight distribution as in Eq. (21)
else Randomly search around by Eq. (11)
Update the parameter of pulse rate $R_i^t$ by Eq. (22)
Update the parameter of loudness $A_i^t$ by Eq. (23)
**end**

### 3.1 Parameters

In this paper, the LBG algorithm is adapted to the LBAT then each bat is represented by a codebook defined as $C = \left\{C_1, C_2, ..., C_{N_C}\right\}$, $C_j \in \Re$, $j = 1, 2, ..., N_C$. The number of codewords are selected as 8, 16, 32, 64 and 128 for the 256 gray level and 256×256 pixel sized standard images. C-Means variable of $\varepsilon$ is defined as 0.001 for VQ. The user defined parameters $k_w$, $k_p$, and $k_g$ of PSO are defined as 0.9, 0.9 and 0.9 respectively when the $random_1$ and $random_2$ are the uniform randoms change in [0, 1]. The $\alpha$ and $\gamma$ parameters of BAT technique defined as 0.997 and 0.996 for image codebook generation. Objective functions of BAT and LBAT are directly proportional to inverse codebook error, which is determined as $MSE^{-1}(C)$. The varibles $\alpha$, $\gamma$ and $\beta_0$ of FRA are determined as 0.5, 0.01 and 1.0 respectively. The coefficients of LBAT are selected as $\lambda = 1.5$ when $\gamma = 0.01$ and $k = 2.0$ for optimization. The iteration number of LBAT is determined as 1500 which is sufficient for LBAT algorithm to reach the global minimum point.

## 4 SIMULATIONS AND RESULTS

The codebook producing is achieved by using 256×256 pixel size of standard images, which have 256 gray levels. The standard images are divided into 4×4 pixel sized of 4096 subvectors. The standard FRA, BAT, PSO, VQ and LBAT algorithms are used for codebook generation. The variables of $k$ and $\lambda$ are important parameters of approximated Lévy Flight function $F_L(X)$. They are responsible for the convergence speed. Therefore they are fine-tuned as $k = 2.0$, $\lambda = 2.0$. The $k$ determines the maximum Lévy Flight step value whereas the $\lambda$ defines the percentile of the long Lévy Flight step rate as seen in Fig. 2. Small values of $k$ and $\lambda$ cause to be caught by a local minimum whereas the high values cause delay in the convergence time duration. When the bat is focusing on the local search the Lévy flight random step provides an opportunity of food search at the different fields of the search area. Therefore most of the time the bat consantrate on the local area for specific food while it rarely flies to the different part of the field for better food opportunity. The x variable interval is determined as $x \in [1,7]$. Consequently, Lévy Flight steps change from $F_L(1) = 2.0$ to $F_L(7) = 0.04$. The bat steps of LBAT algorithm change generally smaller than 0.5 but rarely bigger than 0.5. The long steps provide the bat easily escaping from possible local minimum whereas small steps help bat focusing to the food. The mean square error *MSE*(*C*) performance values for the codebooks of different and proposed techniques are given in Tabs. 1-3. The number of 8, 16, 32, 64 and 128 codewords are used in the simulation. In addition to these results the error convergence graphics of the algorithms according to MSE versus iteration number are shown in Fig. 3 and Fig. 5 for the high contrast 256×256 pixels size Barbara and low contrast 256×256 pixel size Clock standard images.

**Table 1** MSE performances of codebooks of different techniques for Barbara image

| | Codebook Size | VQ | PSO | FRA | BAT | LBAT |
|---|---|---|---|---|---|---|
| Barbara | 8 | 357.9 | 343.3 | 341.7 | 351.0 | 338.6 |
| | 16 | 259.9 | 250.0 | 247.1 | 256.0 | 244.0 |
| | 32 | 197.5 | 189.8 | 186.6 | 197.1 | 183.2 |
| | 64 | 157.1 | 149.5 | 145.5 | 153.3 | 141.9 |
| | 128 | 125.8 | 120.7 | 118.8 | 127.7 | 115.2 |

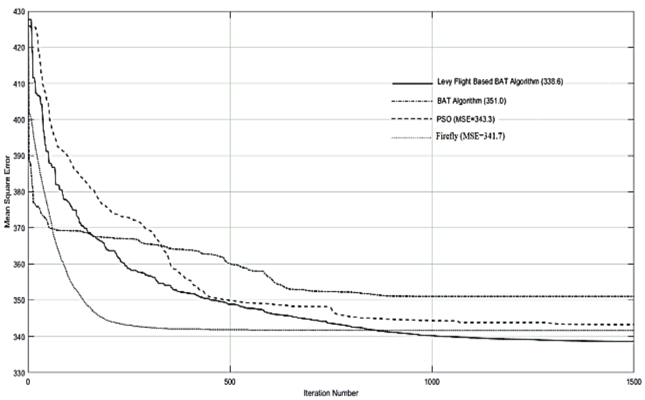



**Figure 3** Convergence results of PSO, FRA, BAT and LBAT algorithms for BARBARA image and 8 codewords of codebook

a) Original Barbara image
b) BAT (MSE=351.0)
c) LBAT (MSE=338.6)
d) FRA (MSE=341.7)
e) PSO (MSE=343.3)
f) VQ (MSE=357.9)

**Figure 4** Visual results of LBAT, BAT, FRA, PSO and VQ algorithms for BARBARA image for 8 codewords of codebook

**Table 2** MSE performances of different techniques for Clock image

| | Codebook Size | VQ | PSO | FRA | BAT | LBAT |
|---|---|---|---|---|---|---|
| Clock | 8 | 291.3 | 287.4 | 280.0 | 304.4 | 271.2 |
| | 16 | 198.9 | 195.0 | 189.7 | 205.5 | 179.7 |
| | 32 | 143.5 | 141.0 | 138.8 | 149.9 | 133.3 |
| | 64 | 125.7 | 123.8 | 120.1 | 131.0 | 117.0 |
| | 128 | 94.8 | 92.5 | 90.2 | 96.7 | 88.8 |

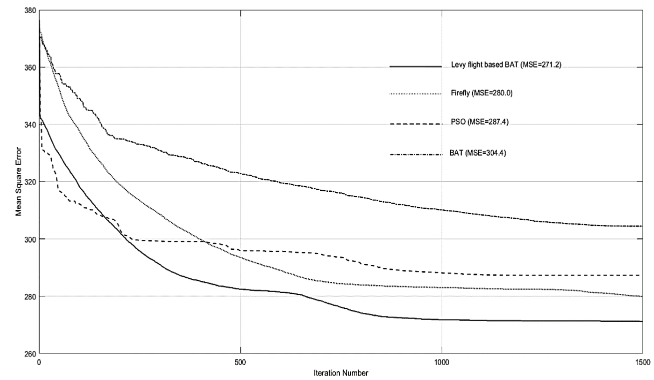
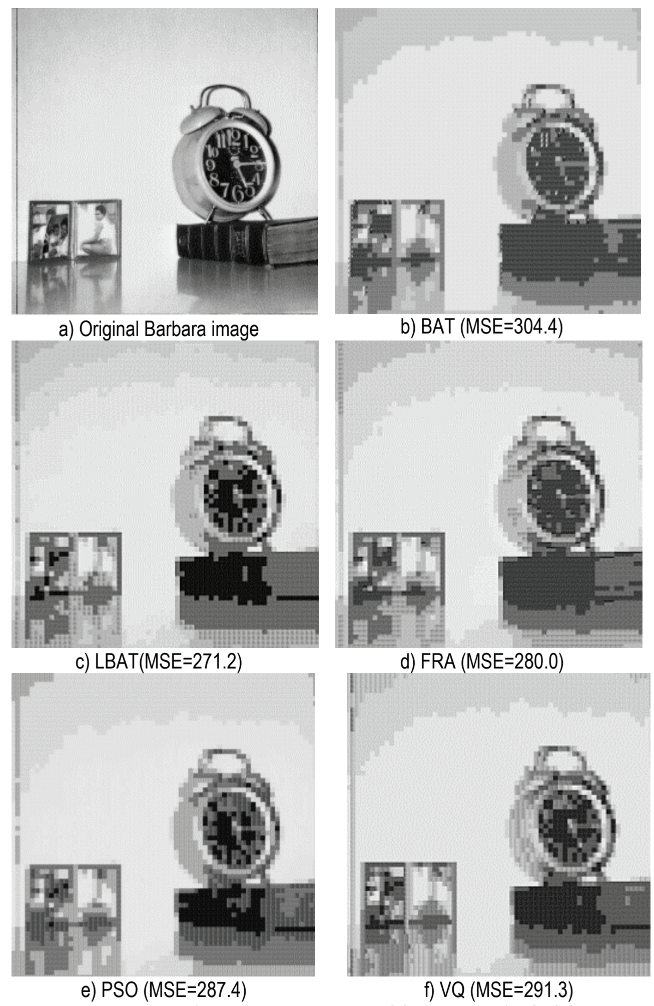



**Figure 5** Convergence results of PSO, FRA, BAT and LBAT algorithms for CLOCK image and 8 codewords of codebook



a) Original Barbara image
b) BAT (MSE=304.4)
c) LBAT(MSE=271.2)
d) FRA (MSE=280.0)
e) PSO (MSE=287.4)
f) VQ (MSE=291.3)

**Figure 6** Visual results of LFRA, FRA, BAT and PSO algorithms for CLOCK image for 8 codewords of codebook

The results of other standard images of 256×256 pixel size are given in Tab. 3.

It is seen from Tabs. 1-3 and Figs. 3-6 that the proposed LBAT codebook MSE performances are better than both classical techniques and metaheuristic algorithms. On the other hand LBAT, FRA, BAT and PSO classical metaheuristic algorithms are superior to the VQ technique. The proposed LFRA algorithm performs better and reaches to the global optimum codebook whereas VQ technique is seem to be captured local minimums. Consequently in the proposed LBAT, when the bats are on the local search they rarely search around by Lévy Flight, which is a small random step in most of the iterations but rarely a big random step. Therefore this mechanism protects a bat to be captured by a possible local minimum and also leads to it on the way of global minimum solution.

**Table 3** MSE performances of different techniques for standard images

| Compression Ratio 0.122 bpp | VQ | PSO | FRA | BAT | LBAT |
|---|---|---|---|---|---|
| Peppers | 332.4 | 330.1 | 323.9 | 339.0 | 320.2 |
| Clock | 291.3 | 287.4 | 280.0 | 304.4 | 271.2 |
| Airplane | 122.8 | 122.5 | 121.1 | 131.0 | 118.5 |
| Boat | 376.7 | 372.8 | 373.3 | 383.1 | 363.2 |
| Chemical P. | 369.9 | 362.2 | 364.0 | 376.8 | 355.9 |
| Cameraman | 416.6 | 410.0 | 411.3 | 422.7 | 405.4 |
| Barbara | 357.9 | 343.3 | 341.7 | 351.0 | 338.6 |
| Einstein | 242.1 | 238.8 | 240.1 | 249.6 | 229.2 |
| Couple | 312.8 | 308.1 | 309.7 | 319.2 | 303.0 |
| Lena | 296.3 | 293.6 | 322.6 | 303.3 | 286.3 |
| Moon | 147.7 | 142.2 | 145.6 | 154.0 | 137.4 |
| Baboon | 433.3 | 427.7 | 432.3 | 440.4 | 418.7 |
| Aerial | 532.2 | 526.6 | 528.5 | 540.3 | 518.4 |

## 5 CONCLUSION

In this paper, the BAT metaheuristic algorithm is improved by using Lévy Flight distribution for codebook generation in image processing. In the proposed LBAT, besides the local search for the global solution, a special random Lévy Flight step opportunity is given to the bat. Lévy Flight satisfies to a bat an opportunity of special random movement, which is generally small random steps in most of the iterations but rarely a big one. Therefore this mechanism protects a bat to be captured by a possible local minimum and also leads to it on the way of global minimum solution. Numerical results suggest that the proposed LBAT is better than both the classical techniques and the metaheuristic algorithms and provides the global optimum codebook.

## Notice

The paper was presented at the International Congress of Electrical and Computer Engineering (ICECENG'22), which took place in Bandırma (Turkey), on February 9-12, 2022. The paper will not be published anywhere else.

## 6 REFERENCES


[1] Gray, R. M. (1984). Vector quantization. *IEEE Assp Magazine*, 4-29. https://doi.org/10.1109/MASSP.1984.1162229

[2] Hu, Y. C., Chen, W. L., et al. (2012). Improved vector quantization scheme for grayscale image compression. *Opto Electronics Review*, *20(2),* 187-193. https://doi.org/10.2478/s11772-012-0016-z

[3] Chuang, J. C., Hu, Y. C., et al. (2013). Improved Mean-Removed Vector Quantization Scheme for Grayscale Image Coding. *International Journal of Signal Processing, Image Processing and Pattern Recognition, 6*(5), 315-332. https://doi.org/10.14257/ijsip.2013.6.5.28

[4] Zhang, L., Zhang, M., et al. (2014). An Effective Image Coding Method using Lattice Vector Quantization in Wavelet Domain. *International Journal of Signal Processing, Image Processing and Pattern Recognition, 7*(2), 305-316. https://doi.org/10.14257/ijsip.2014.7.2.28

[5] Chang, C. C. & Hu, Y. C. (1998). A fast codebook training algorithm for vector quantization. *IEEE Transactions on Consumer Electronics*, *44*, 1201-1208. https://doi.org/10.1109/30.735818

[6] Sun, H., Kwok-Yan Lam, et al. (2005). Efficient vector quantization using genetic algorithm. *Neural Computing and Application, 14*, 203-211. https://doi.org/10.1007/s00521-004-0455-7

[7] Feng, H. M., Chen, C. Y., et al. (2007). Evolutionary fuzzy particle swarm optimization vector quantization learning scheme in image compression. *Expert Systems with Applications, 32*, 213-222. https://doi.org/10.1016/j.eswa.2005.11.012

[8] Horng, M. H. & Jiang, T. W. (2011). Image vector quantization algorithm via honey bee mating optimization. *Expert Systems with Applications, 38*, 1382-1392. https://doi.org/10.1016/j.eswa.2010.07.037

[9] Nag, S. (2019). Vector quantization using the improved differential evolution algorithm for image compression. *Genetic Programming and Evolvable Machines,* 20, 187-212. https://doi.org/10.1007/s10710-019-09342-8

[10] Tsaia, C. W., Tsengb, S. P., et al. (2013). PREACO: A fast ant colony optimization for codebook generation. *Applied Soft Computing, 13*, 3008-3020. https://doi.org/10.1016/j.asoc.2013.01.017

[11] Rani, M. L. P., Rao, G. S., et al. (2020). An efficient codebook generation using firefly algorithm for optimum medical image compression. *Journal of Ambient Intelligence and Humanized Computing,* 1-13. https://doi.org/10.1007/s12652-020-01782-w

[12] Chiranjeevi, K. & Jena, U. R. (2018). Image compression based on vector quantization using cuckoo search optimization technique. *Ain Shams Engineering Journal, 9*, 1417-1431. https://doi.org/10.1016/j.asej.2016.09.009

[13] Yang, X. S. & Gandomi, A.H. (2012). Bat algorithm: a novel approach for global engineering *Engineering Computations: International Journal for Computer-Aided Engineering and Software* optimization, *29*(5), 464-483. https://doi.org/10.1108/02644401211235834

[14] Adarsh, B. R., Raghunathan, T., et al. (2016). Economic dispatch using chaotic bat algorithm. *Energy, 96*, 666-675. https://doi.org/10.1016/j.energy.2015.12.096

[15] Jordehi, A. R. (2015). Chaotic bat swarm optimisation (CBSO). *Applied Soft Computing*, *26*, 523-530. https://doi.org/10.1016/j.asoc.2014.10.010

[16] Yang, X. S. (2014). *Nature-inspired optimization algorithms*. Elsevier. https://doi.org/10.1016/B978-0-12-416743-8.00010-5

[17] Chakri, A., Khelif, R, et al. (2017). New directional bat algorithm for continuous optimization problems. *Expert Systems with Applications*, *69*, 159-75. https://doi.org/10.1016/j.eswa.2016.10.050

[18] Karri, C. & Jena, U. (2016). Fast vector quantization using a Bat algorithm for image compression. *Engineering Science and Technology, an International Journal, 19*, 769-781. https://doi.org/10.1016/j.jestch.2015.11.003

[19] Wang, G., Guo, L., et al. (2013). Lévy-Flight Krill Herd Algorithm. *Mathematical Problems in Engineering*, 1-14, Article ID: 682073. https://doi.org/10.1155/2013/682073

[20] Hakli, H. & Uguz, H. (2014). A novel particle swarm optimization algorithm with Levy flight. *Applied Soft Computing*, *23*, 333-345. https://doi.org/10.1016/j.asoc.2014.06.034

[21] Jensi, R. & Jiji, G. W. (2016). An enhanced particle swarm optimization with levy flight for global optimization. *Applied Soft Computing, 43*, 248-261. https://doi.org/10.1016/j.asoc.2016.02.018
[22] Sharmaa, H., Bansal, J. C., et al. (2016). Lévy flight artificial bee colony algorithm. *International Journal of Systems Science*, *47*(11), 2652-2670. https://doi.org/10.1080/00207721.2015.1010748
[23] Xian, S., Kang, L., et al. (2016). Modified Bat Algorithm Based on Lévy Flight and Opposition Based Learning. *Hindawi Publishing Corporation Scientific Programming*, 1-13, Article ID 8031560. https://doi.org/10.1155/2016/8031560
[24] Ling, Y., Zhou, Y., et al. (2017). Lévy Flight Trajectory-Based Whale Optimization Algorithm for Global Optimization. *IEEE Access, 5*, 6118-6186. https://doi.org/10.1109/ACCESS.2017.2695498
[25] Amirsadri, S., Mousavirad, S. J., et al. (2018). A Levy flight-based grey wolf optimizer combined with back propagation algorithm for neural network training. *Neural Computing & Applications, 30*, 3707-3720. https://doi.org/10.1007/s00521-017-2952-5
[26] Li, Y., Li, X., et al. (2019). An Improved Bat algorithm based on Lévy Flights and adjustment factors. *Symmetry*, *11,* 925-944. https://doi.org/10.3390/sym11070925
[27] Liu, Y. & Cao, B. (2020). A Novel ant colony optimization algorithm with Lévy Flight. *IEEE Access, 8*, 67205-67213. https://doi.org/10.1109/ACCESS.2020.2985498
[28] Abdulwahab, H. A., Noraziah, A., et al. (2019). An enhanced version of Black Hole algorithm via Levy Flight for optimization and data clustering problems. *IEEE Access, 7*, 142085-142096. https://doi.org/10.1109/ACCESS.2019.2937021
[29] Zhou, Y., Ling, Y., et al. (2018). Lévy flight trajectory-based whale optimization algorithm for engineering optimization. *Engineering Computations*, *35(7),* 2406-2428. https://doi.org/10.1108/EC-07-2017-0264
[30] Zhang, J. & Wang, J. S. (2020). Improved Salp Swarm Algorithm Based on Levy Flight and Sine Cosine Operator. *IEEE Access, 8*, 99740-99771. https://doi.org/10.1109/ACCESS.2020.2997783
[31] Deotti, L. M. P., Pereira, J. L. R., et al. (2020). Parameter extraction of photovoltaic models using an enhanced Lévy flight bat algorithm. *Energy Conversion and Management, 221*, 113114. https://doi.org/10.1016/j.enconman.2020.113114
[32] Zhao, J., Gao, Z. M., et al. (2020). The improved slime mould algorithm with Levy flight. *Journal of Physics: Conference Series, 1617*, 012033. https://doi.org/10.1088/1742-6596/1617/1/012033
[33] Ingle, K. K. & Jatoth, R. K. (2020). An efficient JAYA algorithm with Lévy Flight for non-linear channel equalization. *Expert Systems with Applications, 145*, 112970. https://doi.org/10.1016/j.eswa.2019.112970
[34] Iacca, G., Junior, V. C. S., et al. (2021). An improved Jaya optimization algorithm with Lévy flight. *Expert Systems with Applications, 165*, 113902. https://doi.org/10.1016/j.eswa.2020.113902

**Author's contacts:**

**Ilker Kilic**
Pamukkale University,
Electrical and Electronics Department,
Kinikli Campus, 20160, Denizli, Turkey
Tel: +90-0258-2963010
E-mail: ilkerk@pau.edu.tr