10 Bat Algorithms

The *bat algorithm* (BA) is a bio-inspired algorithm developed by Xin-She Yang in 2010 that has been found very efficient. As a result, the literature has expanded significantly since then. This chapter provides a detailed introduction to BA and its new variants. A wide range of diverse applications and case studies are also reviewed and summarized briefly here.

10.1 Echolocation of Bats

10.1.1 Behavior of Microbats

Bats are fascinating animals. They are the only mammals with wings, and they also have advanced capability of echolocation. It is estimated that there are about 1000 different bat species, which accounts for up to 20% of all mammal species. Their size ranges from the tiny bumblebee bats (of about 1.5 to 2 g) to giant bats with a wingspan of about 2 m and weight of up to about 1 kg. Microbats typically have a forearm length of about 2.2 to 11 cm. Most bats use echolocation to a certain degree; among all the species, microbats are a famous example because they use echolocation extensively, whereas megabats do not [2,4,27].

Most microbats are insectivores. Microbats use a type of sonar, called *echolocation*, to detect prey, avoid obstacles, and locate their roosting crevices in the dark. These bats emit a very loud sound pulse and listen for the echo that bounces back from the surrounding objects. Their pulses vary in properties and can be correlated with their hunting strategies, depending on the species. Most bats use short, frequency-modulated signals to sweep through about an octave; others more often use constant-frequency signals for echolocation. Their signal bandwidth varies with species and often increases by using more harmonics.

Studies show that microbats use the time delay from the emission and detection of the echo, the time difference between their two ears, and the loudness variations of the echoes to build up a three-dimensional scenario of the surrounding. They can detect the distance and orientation of the target, the type of prey, and even the moving speed of the prey, such as small insects. Indeed, studies suggest that bats seem to be able to discriminate targets by the variations of the Doppler effect induced by the wing-flutter rates of the target insects [2].

Nature-Inspired Optimization Algorithms. http://dx.doi.org/10.1016/B978-0-12-416743-8.00010-5

10.1.2 Acoustics of Echolocation

Though each pulse lasts only a few thousandths of a second (up to about 8 to 10 ms), it has a constant frequency that is usually in the region of 25 kHz to 150 kHz. The typical range of frequencies for most bat species is in the region between 25 kHz and 100 kHz, though some species can emit higher frequencies up to 150 kHz. Each ultrasonic burst may last typically 5 to 20 ms, and microbats emit about 10 to 20 such sound bursts every second. When they are hunting for prey, the rate of pulse emission can be sped up to about 200 pulses per second when they fly near their prey. Such short sound bursts imply the fantastic ability of the signal processing power of bats. In fact, studies show that the equivalent integration time of the bat ear is typically about 300 to 400 μs.

As the speed of sound in air is typically $v = 340$ m/s at room temperature, the wavelength λ of the ultrasonic sound bursts with a constant frequency f is given by $\lambda = v/f$, which is in the range of 2 mm to 14 mm for the typical frequency range from 25 kHz to 150 kHz. Such wavelengths are obviously on the same order of their prey sizes [2,27].

Amazingly, the emitted pulse could be as loud as 110 dB, and, fortunately, they are in the ultrasonic region. The loudness also varies from the loudest when searching for prey to a quieter base when homing toward the prey. The traveling range of such short pulses is typically a few meters, depending on the actual frequencies. Microbats can manage to avoid obstacles as small as a thin human hair.

Obviously, some bats have good eyesight, and most bats also have very sensitive smell sense. In reality, they use all the senses in combination to maximize the efficient detection of prey and smooth navigation. However, here we are only interested in the echolocation and the associated behavior.

The echolocation behavior of microbats can be formulated in such a way that it can be associated with the objective function to be optimized, and this makes it possible to formulate new optimization algorithms. Here we first outline the basic formulation of BA and then discuss its implementation.

10.2 Bat Algorithms

If we idealize some of the echolocation characteristics of microbats, we can develop various bat-inspired or bat algorithms [32]. For simplicity, we now use the following approximate or idealized rules:

1. All bats use echolocation to sense distance, and they also "know" the difference between food/prey and background barriers.
2. Bats fly randomly with velocity $\boldsymbol{v}_i$ at position $\boldsymbol{x}_i$. They can automatically adjust the frequency (or wavelength) of their emitted pulses and adjust the rate of pulse emission $r \in [0, 1]$, depending on the proximity of their target.
3. Although the loudness can vary in many ways, we assume that the loudness varies from a large (positive) A_0 to a minimum value $A_{\min}$.

Another obvious simplification is that no ray tracing is used in estimating the time delay and three-dimensional topography. Though this might be a good feature for the

```
Bat Algorithm
Initialize the bat population x_i and v_i  (i = 1, 2, ..., n)
Initialize frequencies f_i, pulse rates r_i and the loudness A_i
while (t <Max number of iterations)
   Generate new solutions by adjusting frequency,
   Update velocities and locations/solutions [(10.1) to (10.3)]
   if (rand > r_i)
      Select a solution among the best solutions
      Generate a local solution around the selected best solution
   end if
   Generate a new solution by flying randomly
   if (rand < A_i & f(x_i) < f(x_*))
      Accept the new solutions
      Increase r_i and reduce A_i
   end if
   Rank the bats and find the current best x_*
end while
```

Figure 10.1 Pseudo code of the bat algorithm (BA).

application in computational geometry, we will not use this simplification here, since it is more computationally extensive in multidimensional cases.

In addition to these simplified assumptions, we also use the following approximations for simplicity. In general, the frequency f in a range $[f_{\min}, f_{\max}]$ corresponds to a range of wavelengths $[\lambda_{\min}, \lambda_{\max}]$. For example, a frequency range of [20 kHz, 500 kHz] corresponds to a range of wavelengths from 0.7 mm to 17 mm.

For a given problem, we can also use any wavelength for the ease of implementation. In the actual implementation, we can adjust the range by adjusting the frequencies (or wavelengths). The detectable range (or the largest wavelength) should be chosen such that it is comparable to the size of the domain of interest, then toned down to smaller ranges. Furthermore, we do not necessarily have to use the wavelengths themselves at all. Instead, we can also vary the frequency while fixing the wavelength λ. This is because λ and f are related, since λf is constant. We use this latter approach in our implementation.

For simplicity, we can assume $f \in [0, f_{\max}]$. We know that higher frequencies have short wavelengths and travel a shorter distance. For bats, the typical ranges are a few meters. The rate of pulse can simply be in the range of [0,1], where 0 means no pulses at all and 1 means the maximum rate of pulse emission.

Based on these approximations and idealized rules, the basic steps of BA can be summarized as the schematic pseudo code shown in Figure 10.1.

10.2.1 Movement of Virtual Bats

In simulations, we have to use virtual bats. We have to define the rules of how their positions $\boldsymbol{x}_i$ and velocities $\boldsymbol{v}_i$ in a d-dimensional search space are updated. The new solutions $\boldsymbol{x}_i^t$ and velocities $\boldsymbol{v}_i^t$ at time step t are given by

$$f_i = f_{\min} + (f_{\max} - f_{\min})\beta, \tag{10.1}$$

$$v_i^{t+1} = v_i^t + (x_i^t - x_*) f_i, \tag{10.2}$$

$$x_i^{t+1} = x_i^t + v_i^{t+1}, \tag{10.3}$$

where $\beta \in [0, 1]$ is a random vector drawn from a uniform distribution. Here x_* is the current global best location (solution), which is located after comparing all the solutions among all the n bats. Because the product $\lambda_i f_i$ is a constant, we can use f_i (or λ_i) to adjust the velocity change while fixing the other factor λ_i (or f_i), depending on the type of the problem of interest. In our implementation, we use $f_{\min} = 0$ and $f_{\max} = O(1)$, depending on the domain size of the problem of interest. Initially, each bat is randomly assigned a frequency that is drawn uniformly from $[f_{\min}, f_{\max}]$.

For the local search part, once a solution is selected among the current best solutions, a new solution for each bat is generated locally using random walk

$$x_{\text{new}} = x_{\text{old}} + \epsilon A^{(t)}, \tag{10.4}$$

where $\epsilon \in [-1, 1]$ is a random number, while $A^{(t)} = \langle A_i^t \rangle$ is the average loudness of all the bats at this time step. From the implementation point of view, it is better to provide a scaling parameter to control the step size. Therefore, we can rewrite this equation as

$$x_{\text{new}} = x_{\text{old}} + \sigma \epsilon_t A^{(t)}, \tag{10.5}$$

where ϵ_t is now drawn from a Gaussian normal distribution N(0, 1), and σ is a scaling factor. In our demo implementation, we set $\sigma = 0.01$. Obviously, σ should be linked to the scalings of the design variables of an optimization problem under consideration.

The update of the velocities and positions of bats may have some similarity to the procedure in the standard particle swarm optimization, since f_i essentially controls the pace and range of the movement of the swarming particles. However, BA can be more effective because it uses frequency tuning and parameter control to influence exploration and exploitation.

10.2.2 Loudness and Pulse Emission

Furthermore, the loudness A_i and the rate r_i of pulse emission have to be updated accordingly as the iterations proceed. Because the loudness usually decreases once a bat has found its prey, whereas the rate of pulse emission increases, the loudness can be chosen as any value of convenience. For simplicity, we can also use $A_0 = 1$ and $A_{\min} = 0$, assuming $A_{\min} = 0$ means that a bat has just found the prey and temporarily stops emitting any sound. Now we have

$$A_i^{t+1} = \alpha A_i^t, \quad r_i^{t+1} = r_i^0 [1 - \exp(-\gamma t)], \tag{10.6}$$

where α and γ are constants. In fact, α is similar to the cooling factor of a cooling schedule in the simulated annealing discussed earlier in this book. For any $0 < \alpha < 1$ and $\gamma > 0$, we have

$$A_i^t \to 0, \quad r_i^t \to r_i^0, \quad \text{as } t \to \infty. \tag{10.7}$$

In the simplest case, we can use $\alpha = \gamma$, and we used $\alpha = \gamma = 0.9$ in our simulations. However, it is worth pointing out that the demo code does not include the variations of A and r, which is mainly to show the essence of frequency tuning in the bat algorithm.

The choice of parameters requires some experimentation. Initially, each bat should have different values of loudness and pulse emission rate; this can be achieved by randomization. For example, the initial loudness A_i^0 can typically be taken as 1, whereas the initial emission rate r_i^0 can be around zero or any value $r_i^0 \in (0, 1]$ if using (10.6). Their loudness and emission rates will be updated only if the new solutions are improved, which means that these bats are moving toward the optimal solution.

By analyzing BA closely, we can see that it can capture many charactersitics of other algorithms. If we replace the variations of the frequency f_i by a random parameter and setting $A_i = 0$ and $r_i = 1$, BA essentially becomes the standard PSO. Similarly, if we do not use the velocities, we use fixed loudness and rate: A_i and r_i. For example, for $A_i = r_i = 0.7$, this algorithm is virtually reduced to a simple harmony search (HS), as the frequency/wavelength change is essentially the pitch adjustment while the rate of pulse emission is similar to the harmonic acceptance rate (here with a twist) in the HS algorithm. In other words, HS and PSO can be considered the special cases of BA. Therefore, it is no surprise that BA is efficient.

The current studies imply that the proposed new algorithm is potentially more powerful and thus should be investigated further in many applications of engineering and in solving industrial optimization problems.

10.3 Implementation

From the pseudo code, it is relatively straightforward to implement BA in any programming language. For ease of visualization, we have implemented it using Matlab for various test functions.

There are many standard test functions for validating new algorithms. As a simple benchmark, let us look at the eggcrate function

$$f = x^2 + y^2 + 25(\sin^2 x + \sin^2 y), \quad (x, y) \in [-2\pi, 2\pi] \times [-2\pi, 2\pi].$$

We know that f has a global minimum $f_{\min} = 0$ at $(0, 0)$. In our implementation, we use $n = 25$ to 50 virtual bats, and $\alpha = 0.9$. For the multimodal eggcrate function, a snapshot of the last 10 iterations is shown in Figure 10.2, where all bats move toward the global best $(0, 0)$.

For demonstration purposes, we simplify the bat algorithm by setting A and r as constants. The following Matlab code should work well for function optimization.

```
% ------------------------------------------------------%
% Bat-inspired algorithm for continuous optimization    %
% Programmed by Xin-She Yang @2010                      %
% ------------------------------------------------------%
function [best,fmin,N_iter]=bat_algorithm(para)
% Default parameters
if nargin<1,  para=[10 0.25 0.5];  end
n=para(1);      % Population size, typically 10 to 25
A=para(2);      % Loudness  (constant or decreasing)
r=para(3);      % Pulse rate (constant or decreasing)
```

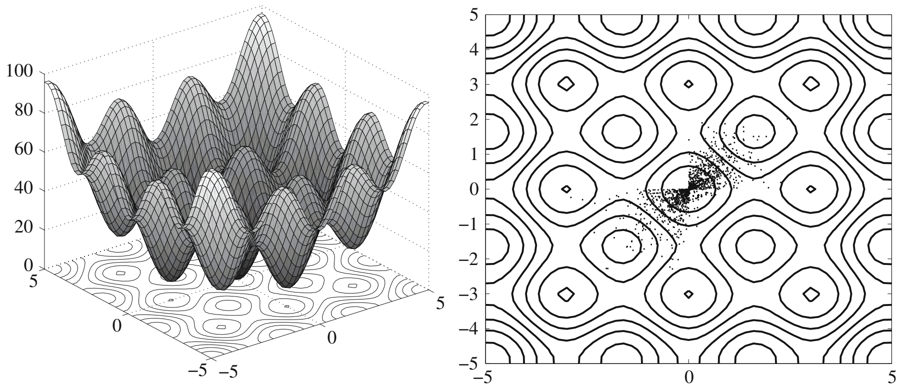

Figure 10.2 The eggcrate function (left) and the locations of 40 bats in the last 10 iterations (right).

```
% This frequency range determines the scalings
Qmin=0;         % Frequency minimum
Qmax=2;         % Frequency maximum
% Iteration parameters
tol=10^(-5);    % Stop tolerance
N_iter=0;       % Total number of function evaluations
% Dimension of the search variables
d=3;
% Initial arrays
Q=zeros(n,1);   % Frequency
v=zeros(n,d);   % Velocities
% Initialize the population/solutions
for i=1:n,
  Sol(i,:)=randn(1,d);
  Fitness(i)=Fun(Sol(i,:));
end
% Find the current best
[fmin,I]=min(Fitness);
best=Sol(I,:);
% Start the iterations -- Bat Algorithm
while (fmin>tol)
        % Loop over all bats/solutions
        for i=1:n,
          Q(i)=Qmin+(Qmin-Qmax)*rand;
          v(i,:)=v(i,:)+(Sol(i,:)-best)*Q(i);
          S(i,:)=Sol(i,:)+v(i,:);
          % Pulse rate
          if rand>r
              S(i,:)=best+0.01*randn(1,d);
          end
```

```
     % Evaluate new solutions
           Fnew=Fun(S(i,:));
     % If the solution improves or not too loudness
           if (Fnew<=Fitness(i)) & (rand<A) ,
                Sol(i,:)=S(i,:);
                Fitness(i)=Fnew;
           end

          % Update the current best
          if Fnew<=fmin,
                best=S(i,:);
                fmin=Fnew;
          end
        end
        N_iter=N_iter+n;
end
% Output/display
disp(['Number of evaluations: ',num2str(N_iter)]);
disp(['Best =',num2str(best),' fmin=',num2str(fmin)]);
% Objective function -- Rosenbrock's 3D function
function z=Fun(u)
z=(1-u(1))^2+100*(u(2)-u(1)^2)^2+(1-u(3))^2;
```

BA is much superior to other algorithms in terms of accuracy and efficiency for many test functions because it usually provides a fast convergence rate.

10.4 Binary Bat Algorithms

The basic bat algorithm works well for continuous problems. But to deal with discrete and combinatorial problems, some modifications are needed to discretize the bat algorithm. Nakamura et al. developed the so-called *binary bat algorithm* (BBA) for feature selection and image processing [21]. For feature selection, they proposed that the search space is modeled as a d-dimensional Boolean lattice in which bats move across the corners and nodes of a hypercube. For binary feature selection, a feature is represented by a bat's position as a binary vector.

In BBA, a sigmoid function is used to restrict a bat's position. That is,

$$x_i^j = \begin{cases} 1 & \text{if } S(v_i^j) > \rho, \\ 0 & \text{otherwise} \end{cases}, \tag{10.8}$$

and

$$S(v_i^j) = \frac{1}{1 + \exp[-v_i^j]}, \tag{10.9}$$

where the velocity component v_i^j corresponds to the jth dimension of bat i, and $\rho \sim U(0, 1)$ is a random number drawn from a uniform distribution. This means that the positions or coordinates of bats are in the Boolean lattice.

Obviously, it can be expected that this BBA variant can be extended to deal with other discrete and combinatorial problems. In addition, there are other ways to discretize the bat algorithm. In fact, any efficient ways to discrete swarm-based algorithms such as PSO can also be used to discretize BA and produce new variants of it. In fact, this has become an active research topic.

10.5 Variants of the Bat Algorithm

The standard BA has many advantages, a key one being that it can provide very quick convergence at an early stage of iteration by switching from exploration to exploitation if necessary. This makes BA an efficient algorithm for applications such as classifications and others when a fast solution is needed. However, if we allow the algorithm to switch to exploitation stage too fast by varying A and r too quickly, it may lead to stagnation after some initial stage. To improve performance, many methods and strategies have been attempted to increase the diversity of the solution and thus to enhance performance, which produced a few good BA variants.

From a quick but incomplete literature survey [38], we found the following BA variants:

- *Fuzzy logic bat algorithm (FLBA)*. Khan and Sahai [14] presented a variant by introducing fuzzy logic into BA. They called their variant FLBA [14].
- *Multi-objective bat algorithm (MOBA)*. Yang [33] extended BA to deal with multi-objective optimization, which has demonstrated its effectiveness for solving a few design benchmarks in engineering [33].
- *K-means bat algorithm (KMBA)*. Komarasamy and Wahi [16] presented a combination of K-means and bat algorithm (KMBA) for efficient clustering [16].
- *Chaotic bat algorithm (CBA)*. Lin et al. [17] presented a chaotic BA using Lévy flights and chaotic maps to carry out parameter estimation in dynamic biological systems [17]. Gandomi and Yang [10] proposed another CBA using various iterative maps [10].
- *Binary bat algorithm (BBA)*. Nakamura et al. [21] developed a discrete version of BA to solve classifications and feature selection problems [21].
- *Improved bat algorithm (IBA)*. Jamil et al. [13] extended BA with a good combination of Lévy flights and subtle variations of loudness and pulse emission rates. They tested the IBA versus more than 70 different test functions, and IBA proved to be very efficient [13].
- *Modified bat algorithm (MBA)*. Huang et al. [11] developed this invariant by using orthogonal Latin square sampling as the initial population and incorporating autonomous danger-averting behavior. The researchers proved that MBA has guaranteed global convergence [11].

There are other improvements and variants of BA. For example, Wang and Guo [31] hybridized BA with harmony search and produced a hybrid BA for numerical optimization of function benchmarks [30]. They also applied BA to vehicle routing problems [31].

On the other hand, Fister, Jr., et al. [8] developed a hybrid BA using differential evolution as a local search part of BA [8]. We can expect that more variants are still under active research.

10.6 Convergence Analysis

Huang et al. carried out a detailed convergence analysis for BA using the finite Markov process theory [11].

In theory, an algorithm with an order-m reducible stochastic matrix P can be rewritten as

$$P = \begin{pmatrix} S & \cdots & 0 \\ R & \cdots & T \end{pmatrix}, \tag{10.10}$$

where $R \neq 0, T \neq 0$ and S is an order-q stochastic matrix (with $q < m$). Then we have

$$P^{\infty} = \lim_{k\to\infty} P^k$$
$$= \lim_{k\to\infty} \begin{pmatrix} S^k & \cdots & 0 \\ \sum_{i=1}^{k-1} T^i R S^{k-i} & \cdots & T^k \end{pmatrix} = \begin{pmatrix} S^{\infty} & \cdots & 0 \\ R^{\infty} & \cdots & T \end{pmatrix}, \tag{10.11}$$

which is a stable stochastic matrix and independent of the initial distribution [12]. In addition, we also have

$$P^{\infty} = [p_{ij}]_{m\times m}, \quad \begin{cases} p_{ij} > 0, & (1 \le i \le m, 1 \le j \le q), \\ p_{ij} = 0, & (1 \le i \le m, q < j \le m). \end{cases} \tag{10.12}$$

The search algorithm will converge with almost probability one to the global optimality, starting from any initial random states, if the transition probability p to a better solution or state is $p > 0$. Conversely, if the transition probability p to a worse state is greater, the algorithm will not converge.

With this main result, it has been proved that PSO will not converge to the global optimality [26], whereas BA will converge to the true global optimality [11].

Huang et al. concluded that for unconstrained function optimization, BA satisfies all the conditions for guaranteed global convergence. For nonlinear constrained problems, BA will converge with additional initialization of orthogonal Latin squares and has guaranteed global convergence to the true global optimality. They further concluded that

$$S^{\infty} = (1), \quad R^{\infty} = (1, 1, \ldots, 1)^T \tag{10.13}$$

and

$$P^{\infty} = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 1 & 0 & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ 1 & 0 & \cdots & 0 \end{pmatrix}, \tag{10.14}$$

which leads to

$$\lim_{t\to\infty} p\{f(\boldsymbol{x}) \to f(\boldsymbol{x}_*)\} = 1. \tag{10.15}$$

That is, the global convergence is guaranteed.

They also proposed a BA variant, called a *modified bat algorithm* (MBA), can further improve the convergence rate with guaranteed global optimality. They also showed that this variant is suitable for large-scale, global optimization.

10.7 Why the Bat Algorithm is Efficient

Like many metaheuristic algorithms, BA has the advantage of simplicity and flexibility. BA is easy to implement, and such a simple algorithm can be very flexible to solve a wide range of problems, as we have seen in our review.

A natural question is: Why is the bat algorithm so efficient? There are many reasons for the success of bat-based algorithms. By analyzing the key features and updating equations, we can summarize the following three key points and features:

- *Frequency tuning*. BA uses echolocation and frequency tuning to solve problems. Though echolocation is not directly used to mimic the true function in reality, frequency variations are used. This capability can provide some functionality that may be similar to the key feature used in PSO, SA, and HS. Therefore, BA possesses the advantages of other swarm-intelligence-based algorithms.
- *Automatic zooming*. BA has a distinct advantage over other metaheuristic algorithms. That is, BA has a capability of automatically zooming into a region where promising solutions have been found. This zooming is accompanied by the automatic switch from explorative moves to local intensive exploitation. As a result, BA has a quick convergence rate, at least at early stages of the iterations, compared with other algorithms.
- *Parameter control*. Many metaheuristic algorithms used fixed parameters by using some pre-tuned algorithm-dependent parameters. In contrast, BA uses parameter control, which can vary the values of parameters (A and r) as the iterations proceed. This provides a way to automatically switch from exploration to exploitation when the optimal solution is approaching. This gives another advantage of BA over other metaheuristic algorithms.

In addition, preliminary theoretical analysis by Huang et al., discussed in an earlier section, suggested that BA has guaranteed global convergence properties under the right conditions, and BA can also solve large-scale problems effectively.

10.8 Applications

The standard BA and its many variants mean that the applications are also very diverse. In fact, since the original BA was developed [32], bat algorithms have been applied in almost every area of optimization, classifications, image processing, feature selection, scheduling, data mining, and other problems. In the rest of the chapter, we briefly highlight some of the applications [9,23,32,33,36,37]. This review is based on detailed review articles in the literature [34,35,38].

10.8.1 Continuous Optimization

Among the first set of applications of BA, continuous optimization in the context of engineering design optimization has been extensively studied and demonstrated that BA can deal with highly nonlinear problems efficiently and can find the optimal solutions accurately [32,36]. Case studies include pressure vessel design, car side design, spring and beam design, truss systems, tower and tall building design, and others. Tsai et al. [29] solved numerical optimization problems using BA [29].

In addition, Bora et al. [3] optimized the brushless DC wheel motors using BA with superior results [3]. BA can also handle multi-objective problems effectively [33].

10.8.2 Combinatorial Optimization and Scheduling

From the computational complexity point of view, continuous optimization problems can be considered easy to solve, though they may still be very challenging. However, combinatorial problems can be really hard, often nondeterministic polynomial time hard (NP-hard). Ramesh et al. [24] presented a detailed study of combined economic load and emission dispatch problems using BA [24]. They compared BA with ant colony algorithms (ABC), hybrid genetic algorithms, and other methods and concluded that BA is easy to implement and much superior to the other algorithms in terms of accuracy and efficiency.

Musikapun and Pongcharoen (2012) solved multistage, multimachine, multiproduct scheduling problems using BA, and they solved a class of NP-hard problems with a detailed parametric study. They also implied that the performance can be further improved by about 8.4% using an optimal set of parameters [20].

10.8.3 Inverse Problems and Parameter Estimation

Yang et al. use BA to study topological shape optimization in microelectronic applications so that materials of different thermal properties can be placed in such a way that the heat transfer is most efficient under stringent constraints [37]. It can also be applied to carry out parameter estimation as an inverse problem. If an inverse problem can be properly formulated, BA can provide better results than least-squares methods and regularization methods.

Lin et al. [17] presented a chaotic Lévy flight BA to estimate parameters in nonlinear dynamic biological systems, which proved the effectiveness of the proposed algorithm [17].

10.8.4 Classifications, Clustering, and Data Mining

Komarasamy and Wahi [16] studied K-means clustering using BA and concluded that the combination of both K-means and BA can achieve higher efficiency and thus perform better than other algorithms tested in their work [16].

Khan and Sahari presented a comparison study of BA with PSO, GA, and other algorithms in the context of e-learning and thus suggested that BA clearly has some advantages over other algorithms [14]. They also presented a study of clustering problems using BA and its extension as a bisonar optimization variant, with good results [15].

On the other hand, Mishra et al. [19] used BA to classify microarray data [19], whereas Natarajan et al. [22] presented a comparison study of cuckoo search and BA for Bloom filter optimization [22]. Damodaram and Valarmathi [5] studied phishing Website detection using modified BA and achieved very good results [5].

Marichelvam and Prabaharan [18] used BA to study hybrid flowshop scheduling problems so as to minimize the makespan and mean flow time [18]. Their results suggested that BA is an efficient approach for solving hybrid flowshop scheduling problems. Faritha Banu and Chandrasekar [7] used a modified BA to record deduplication as an optimization approach and data compression technique. Their results suggested that the modified BA can perform better than genetic programming [7].

10.8.5 Image Processing

Akhtar et al. [1] presented a study for full-body human-pose estimation using BA [1], and they concluded that BA performs better than PSO, particle filter (PF), and annealed particle filter (APF).

Du and Liu [6] presented a variant of BA with mutation for image matching, and they indicated that their bat-based model is more effective and feasible in imagine matching than other models, such as differential evolution and genetic algorithms [6].

10.8.6 Fuzzy Logic and Other Applications

Reddy and Manoj [25] presented a study of optimal capacitor placement for loss reduction in distribution systems using BA. It combines with fuzzy logic to find optimal capacitor sizes so as to minimize the total losses. Their results suggested that the real power loss can be reduced significantly [25]. Furthermore, Tamiru and Hashim [28] applied an approach based on BA to study fuzzy systems and to model exergy changes in a gas turbine [28].

At the time of writing, when we searched Google Scholar and other databases, we found other papers on BA that either had just been accepted or were conference presentations. However, they do not contain enough detail to be included in this review. In fact, as the literature is expanding, more and more papers on BA are emerging, so a further timely review will be needed within the next few years.

An interesting extension will be to use different schemes of wavelengths or frequency variations instead of the current linear implementation. In addition, the rates of pulse emission and loudness can also be varied in a more sophisticated manner. Another extension for discrete problems is to use the time delay between pulse emission and the echo bounced back. For example, in the traveling salesman problem, the distance between two adjacent nodes or cities can easily be coded as the time delay. Because microbats use time difference between their two ears to obtain three-dimensional information, they can identify the type of prey and the velocity of a flying insect. Therefore, a further natural extension to the current BA would be to use the directional echolocation and Doppler effect, which may lead to even more interesting variants and new algorithms.

References

[1] Akhtar S, Ahmad AR, Abdel-Rahman EM. A metaheuristic bat-inspired algorithm for full body human pose estimation. In: Ninth conference on computer and robot vision (CRV2012) 28–30 May 2012, Toronto, ON. Toronto, ON, Canada: Conference Publishing Service; 2012. p. 369–75.

[2] Altringham JD. Bats: biology and behaviour. Oxford, UK: Oxford University Press; 1996.

[3] Bora TC, Coelho LS, Lebensztajn L. Bat-inspired optimization approach for the brushless DC wheel motor problem. IEEE Trans Magn 2012;48(2):947–50.

[4] Colin T. The varienty of life. Oxford, UK: Oxford University Press; 2000.

[5] Damodaram R, Valarmathi ML. Phishing website detection and optimization using modified bat algorithm. Int J Eng Res Appl 2012;2(1):870–6.

[6] Du ZY, Liu B. Image matching using a bat algorithm with mutation. Appl Mech Mater 2012;203(1):88–93.

[7] Faritha Banu A, Chandrasekar C. An optimized appraoch of modified bat algorithm to record deduplication. Int J Comput Appl 2012;62(1):10–5.

[8] Fister Jr I, Fister D, Yang XS. A hybrid bat algorithm. Elekrotehnivški Vestn 2013;80(1–2):1–7.

[9] Gandomi AH, Yang XS, Alavi AH, Talatahari S. Bat algorithm for constrained optimization tasks. Neural Comput Appl 2013;22(6):1239–55.

[10] Gandomi AH, Yang XS. Chaotici bat algorithm. J Comput Sci 2013. <http://www.sciencedirect.com/science/article/pii/S1877750313001099>.

[11] Huang GQ, Zhao WJ, Lu QQ. Bat algorithm with global convergence for solving large-scale optimization problem. Appl Res Comput 2013;30(5):1323–8. [in Chinese].

[12] Iisufescu M. Finite Markov processes and their applications. Chichester, UK: John Wiley; 1980.

[13] Jamil M, Zepernic H-J, Yang XS. Improved bat algorithm for global optimization. Appl Soft Comput [submitted, Sept 2013].

[14] Khan K, Sahai A. A comparison of BA, GA, PSO, BP and LM for training feed forward neural networks in e-learning context. Int J Intell Sys Appl 2012;4(7):23–9.

[15] Khan K, Sahai A. A fuzzy c-means bi-sonar-based metaheuristic optimization algorithm. Int J Interact Multimedia Art Intell 2012;1(7):26–32.

[16] Komarasamy G, Wahi A. An optimized K-means clustering technique using bat algorithm. Euro J Sci Res 2012;84(2):263–73.

[17] Lin JH, Chou CW, Yang CH, Tsai HL. A chaotic Lévy flight bat algorithm for parameter estimation in nonlinear dynamic biological systems. J Comput Inf Tech 2012;2(2):56–63.

[18] Marichelvam MK, Prabaharam T. A bat algorithm for realistic hybrid flowshop scheduling problems to minimize makespan and mean flow time. ICTACT J Soft Comput 2012;3(1):428–33.

[19] Mishra S, Shaw K, Mishra D. A new meta-heuristic bat inspired classification approach for microarray data. Procedia Tech 2012;4(1):802–6.

[20] Musikapun P, Pongcharoen P. Solving multi-stage multi-machine multi-product scheduling problem using bat aglorithm. Second international conference on management and artificial intelligence (IPEDR), vol. 35. Singapore: IACSIT Press; 2012. p. 98–102.
[21] Nakamura RYM, Pereira LAM, Costa KA, Rodrigues D, Papa JP, Yang XS. BBA: a binary bat algorithm for feature selection. In: 25th SIBGRAPI conference on graphics, patterns and images (SIBGRAPI), August 22–25, 2012. IEEE Publication; 2012. p. 291–7.
[22] Natarajan A, Subramanian S, Premalatha K. A comparative study of cuckoo search and bat algorithm for Bloom filter optimisation in spam filtering. Int J Bio-Inspired Comput 2012;4(2):89–99.
[23] Parpinelli RS, Lopes HS. New inspirations in swarm intelligence: a survey. Int J Bio-Inspired Comput 2011;3(1):1–16.
[24] Ramesh B, Mohan VCJ, Reddy VCV. Application of bat algorithm for combined economic load and emission dispatch. Int J Electr Eng Telecommun 2013;2(1):1–9.
[25] Reddy VU, Manoj A. Optimal capacitor placement for loss reduction in distribution systems using bat algorithm. IOSR J Eng 2012;2(10):23–7.
[26] Ren ZH, Wang J, Gao YL. The global convergence of particle swarm optimization based on Markov chain. Control Theory Appl 2011;28(4):462–6.
[27] Richardson P. Bats. London: Natural History Museum; 2008.
[28] Tamiru AL, Hashim FM. Application of bat algorithm and fuzzy systems to model exergy changes in a gas turbine. In: Yang XS, editor. Artificial intelligence, evolutionary computing and metaheuristics. Studies in computational intelligence, vol. 427. Heidelberg, Germany: Springer; 2013. p. 685–719.
[29] Tsai PW, Pan JS, Liao BY, Tsai MJ, Istanda V. Bat algorithm inspired algorithm for solving numerical optimization problems. Appl Mech Mater 2011;148–149(1):134–7.
[30] Wang GG, Guo LH, Duan H, Liu L, Wang HQ. A bat algorithm with mutation for UCAV path planning. Sci World J 2012. Available from: <www.hindawi.com/journals/tswj/2012/418946/>.
[31] Wang GG, Guo LH. A novel hybrid bat algorithm with harmony search for global numerical optimization. J Appl Math 2013. Available from: <http://dx.doi.org/10.1155/2013/696491> .
[32] Yang XS. A new metaheuristic bat-inspired algorithm. In: Cruz C, González JR, Pelta DA, Terrazas G, editors. Nature inspired cooperative strategies for optimization (NISCO 2010). Studies in computational intelligence. Berlin, Germany: Springer; 2010. p. 65–74.
[33] Yang XS. Bat algorithm for multi-objective optimisation. Int J Bio-Inspired Comput 2011;3(5):267–74.
[34] Yang XS. Bat algorithm and cuckoo search: a tutorial. In: Yang XS, editor. Artificial intelligence, evolutionary computing and metaheuristics. Studies in computational intelligence, vol. 427; 2013. p. 421–34.
[35] Yang XS, Cui ZH, Xiao RB, Gandomi AH, Karamanoglu M. Swarm intelligence and bio-inpsired computation: theory and applications. London: Elsevier; 2013.
[36] Yang XS, Gandomi AH. Bat algorithm: a novel approach for global engineering optimization. Eng Comput 2012;29(5):464–83.
[37] Yang XS, Karamanoglu M, Fong S. Bat aglorithm for topology optimization in microelectronic applications. In: IEEE international conference on future generation communication technology (FGCT2012). London, UK: British Computer Society; 2012. p. 150–5. December 12–14.
[38] Yang XS. Bat algorithm: literature review and applications. Int J Bio-Inspired Comput 2013;5(3):141–9.