

The Bat Algorithm, Variants and Some Practical Engineering Applications: A Review

T. Jayabarathi, T. Raghunathan and A.H. Gandomi

Abstract The bat algorithm (BA), a metaheuristic algorithm developed by Xin-She Yang in 2010, has since been modified, and applied to numerous practical optimization problems in engineering. This chapter is a survey of the BA, its variants, some sample real-world optimization applications, and directions for future research.

Keywords Algorithm · Bat algorithm · Engineering application · Optimization · Swarm intelligence · Metaheuristics

1 Introduction

Real-world optimization problems do not conform to the requirements of calculus or gradient based optimization methods that require functions to be continuous, smooth and unimodal, with ever present derivatives and ideal constraints. In reality, functions can be noisy, filled with discontinuities, have multiple optimums, and their derivatives may be non-existent [1]. To solve such problems, researchers have been increasingly looking to nature as the ultimate expert on optimization. A pioneer in the field who conclusively demonstrated that practical problems of significant complexity could be solved by nature inspired algorithms is Goldberg, who solved an oil transportation problem that was not amenable to gradient based optimization, and thought to be even less amenable to nature inspired methods, for his PhD in the year 1983 [2]. Given their enormous popularity, nature inspired algorithms are known by many names, including evolutionary computation, the oldest name. Since an element of learning or heuristics is involved, and this

T. Jayabarathi · T. Raghunathan (✉)
School of Electrical Engineering, VIT University, Vellore 632014, India
e-mail: raghunathan.t@vit.ac.in; nathraghu@yahoo.com

A.H. Gandomi
School of Business, Stevens Institute of Technology, Hoboken, NJ 07030, USA

heuristics is of higher level, they are also popularly known as metaheuristic algorithms at present.

The genetic algorithm (GA) used by Goldberg belongs to the family of evolutionary algorithms (EAs), which loosely model evolution in biology as an optimization process. The 1990s onwards saw the development of another family of nature based algorithms that modeled the collective behavior of social animals as an optimization process. The ant colony optimization (ACO) [3], particle swarm optimization (PSO) [4] and krill herd [5] algorithms belong to this family of swarm intelligence (SI) algorithms. Nature inspired algorithms continued to be developed at an even faster pace in the 2000s, and the year 2010 saw the emergence of the bat algorithm (BA), an SI algorithm developed by Xin-She Yang [6]. This was followed by the application of the BA to engineering applications in 2012 [7], and to constrained optimization problems in 2013 [8]. A good reference to swarm intelligence methods of the period is [9]. Since then, the BA has seen numerous variants being developed, and these applied to solve many real-world problems.

In the search for an optimum in multimodal search or solution space, two conflicting objectives need to be catered to: exploration or diversification, and exploitation or intensification. Striking the right balance these two objectives can be the difference between successfully locating the global optimum and not doing so. In the initial stages of the search, all areas of the solution space have to be explored, if the global optimum is not to be missed. On the other hand, once the most promising areas have been located, further exploration would lead the search to meander about aimlessly. Instead, the requirement at this later stage of the search is to exploit or intensify the search in the narrowed down promising areas, so that the optimum can be located. A good search algorithm must thus have operators for exploration and exploitation.

1.1 Exploration or Diversification

Reference [6] contains a detailed description of the food location behavior of bats in nature, and the modeling of this behavior to form the BA metaheuristic. Hence the motivation here is to recapture the most essential details of the BA in the language of general optimization theory. In summary, the BA mimics the collective behavior of a colony of bats that use a phenomenon known as echolocation. Echolocation by bats is characterized by the emission of pulses of some frequency f , loudness A and emission rate R . The exploration capability of the BA is provided by its velocity and position update equations, given by

$$V^i(t+1) = V^i(t) + f^i \{X^i(t) - X^{best}(t)\} \quad (1)$$

$$f^i(t) = f^{\min} + \text{rand}() (f^{\max} - f^{\min}) \quad (2)$$

$$X^i(t+1) = X^i(t) + V^i(t+1) \tag{3}$$

where t is the current iteration number, V^i, f^i and X^i are the velocity, frequency, and position, respectively, of the i th bat in the population of N_b bats, and X^{best} is the bat location (solution) that has the best fitness in the current population. $\text{rand}() \in [0, 1]$ is a randomly generated number from a uniform distribution in the interval $[0,1]$. f^{max} and f^{min} are the allowable maximum and minimum frequencies, which can assume the default values of 0 and 100, but can be varied to suit the problem being solved. At initialization ($t=0$), V^i is assumed to be 0.

1.2 Exploitation or Intensification

The exploitation or local search is by means of a random walk. Two parameters, the loudness $A^i(t)$ and the pulse emission rate $R^i(t)$ are updated at every iteration, for every bat in the population. Depending on $R^i(t)$, a local search is conducted, either around the best solution or a randomly chosen solution:

$$X^{i,\text{new}} = \begin{cases} X^{i,\text{old}}(t) + r_2 A^i(t) & \text{if } \text{rand}() > R^i(t) \\ X^r(t) + r_2 A^i(t) & \text{else} \end{cases} \tag{4}$$

where $\text{rand}() \in [0, 1]$ and $r_2 \in [-1, 1]$ are uniformly distributed random numbers, and $r \in [1, 2, \dots, N_b], r \neq i$ is a randomly chosen integer. In other words, $X^r(t)$ is a randomly chosen solution in the current iteration, and different from the i th solution.

The right balance between exploration and exploitation as the search progresses is provided by adjusting the pulse emission rate $R^i(t)$ and loudness $A^i(t)$ dynamically:

$$R^i(t+1) = R^i(0)[1 - \exp(-\gamma t)] \tag{5}$$

$$A^i(t+1) = \alpha A^i(t) \tag{6}$$

where $R^i(0) \in [0, 1]$ and $A^i(t) \in [1, 2]$, both randomly generated, within their respective limits. As a first choice, the default values that can be used are $\gamma = \alpha = 0.9$, as in [6].

1.3 Selection

In order to improve the solutions over the iterations, a fitness based, tournament type of solution, in which the competitors are the old and new solutions is implemented. The fitter solution replaces the less fit one, with a probability

$$A^i(t): X^i(t) = X^{i,\text{new}}(t) \text{ if } F(X^{i,\text{new}}(t)) < F(X^i(t)) \text{ and } \text{rand}() < A^i(t) \forall i, i \in [1, 2, \dots, N_b] \quad (7)$$

where $\text{rand}() \in [0, 1]$ is a uniformly distributed random number, and $F(X(t))$ is the fitness or cost function to be minimized.

The BA is shown to perform well on some benchmark unimodal and multimodal functions in comparisons against the PSO and GA in [6].

2 Variants of the Bat Algorithm

Once the basic BA showed initial promise as a good metaheuristic algorithm, the focus shifted towards improving it further and applying it to real-world optimization problems for better results. Some of the modifications proposed to improve the performance of the BA are outlined next.

2.1 Chaotic Bat Algorithm

The basic BA outlined in Sect. 1 used default values of the algorithm parameters, and exponentially decreasing values for loudness A^i . For reasons yet to be studied, using chaotic maps or sequences to update the loudness has been shown to produce better performance on the complicated real-world problem solved in [10]. Instead of using Eq. (6) to update the loudness, Ref. [10] updated the loudness using

$$A^i(t+1) = a[A^i(t)]^2 \sin[\pi A^i(t)] \quad (8)$$

where a is set to 2.3 and $A^i(0) \in [0, 1]$, and is randomly generated. Equation (8) describes the sinusoidal chaotic map or sequence of the variable A^i . Other chaotic maps are given in [11], which proposed the chaotic BA (CBA) used in [10]. Another version of the CBA was proposed by Jordehi [12].

2.2 *Directional Bat Algorithm*

Instead of updating the bat positions using Eqs. (1)–(3), the directional BA proposed by Chakri et al. [13] probes the search space in two different directions: one in the direction of the best bat, and the other a randomly selected one. The one with better fitness is used to update the current bat position. When used along with a few other modifications, this directional BA is claimed to improve the performance when tested on a suite of benchmark functions.

2.3 *θ -Modified Bat Algorithm*

The central idea of the θ -modified BA is to use a polar framework, instead of the Cartesian coordinate framework used by Eqs. (1)–(3) [14]. A second modification is to dynamically update of the parameter α , instead of using a fixed value, as in the basic BA described in Sect. 1. This is claimed to have produced the best results on the stochastic multi-objective problems solved in [14].

2.4 *The BA with Mutation*

One of the earliest ideas was to equip the BA, which is an SI algorithm, with operators from EAs.

In nature, mutation introduces new genetic material into the existing gene pool, thereby helping to maintain the diversity of the population. The mutation operator in EAs plays the same role of maintaining the diversity of the search space. Its operation can be explained quite simply with the help of the binary genetic algorithm (BGA) in [1], since the solutions or chromosomes or strings in a BGA consist of just 1's and 0's. If the string length of the individual solution or chromosome or string is len_{str} and the population consists of N_p number of individuals, the total number of bits in the population is $N_p \times str_{len}$. If the probability of mutation is equal to $1/(N_p \times str_{len})$, applying the mutation operator to this population involves flipping one randomly chosen bit in the population: if this bit is a 1, it is changed to 0 and vice versa. In multimodal search space, mutation applied correctly can help the search break out of being trapped in local minima. In real valued or continuous GA, mutation is slightly more complex to implement. The interested reader is referred to [15] for more details.

On some problems, the BA could be lacking in exploratory capability, [16] introduced the BA with mutation, to successfully produce better results on a variant of the basic economic dispatch problem, which is to be discussed in Sect. 3 of this chapter. The BA with mutation for a different application can be found in [17]. An

enhanced BA with four different types of mutation for self-adaptive learning mechanism (SALM) can be found in [18].

2.5 The BA with Mutation and Crossover

As a logical next step, another evolutionary operator of crossover was also introduced in [19]. Crossover between two parent individuals produces a new offspring or solution that could be potentially fitter than the parents, thereby progressing the search towards the optimum.

2.6 The BA with DE Mutation and Crossover

Differential evolution (DE) is an evolutionary algorithm whose performance was found to be superior to the simple GA on complicated real-world problems [20, 21], by virtue of the kind of mutation and crossover that the DE uses, and known as DE mutation and crossover. Reference [22] explored the hybridization of the bat algorithm with differential evolution, to obtain better results.

2.7 The BA with DE Mutation and Lévy Flights Trajectory (DLBA)

The basic BA is equipped with differential evolution (DE) mutation, and Lévy flights trajectory, to improve its performance. The DLBA is shown to perform better than the basic BA on a suite of unimodal and multimodal benchmark functions in [23].

2.8 The Double-Subpopulation Lévy Flight Bat Algorithm

In this variant [24], the bat population is divided into two subpopulations, internal and external. The internal subpopulation aims at better exploitation, by employing the current speed and global optimal values for updating the speed, and a Lévy flight model. The external subpopulation aims at better exploration, employing DE mutation and crossover, whenever the population diversity reaches a minimum value called the diversity threshold.

2.9 *BA with Habitat Selection and Self-adaptive Compensation for Doppler Effect in Echoes*

In this variant [25], the position of a bat in the population can display both quantum and mechanical behavior. In quantum behavior, it can appear anywhere in the whole search space with a certain probability. In physical behavior, the frequency update equation has a term that compensates the Doppler effect of change of frequency as the target moves relative to its source.

3 Application of the Bat Algorithm to the Economic Dispatch Problem

3.1 Problem Formulation

Whenever a new algorithm is proposed, its performance is first tested by applying it on unconstrained, unimodal and multimodal benchmark test functions. In contrast, most real-world problems are constrained, and often, even the existence of a solution that satisfies all the constraints is not known beforehand. Details like how violations of the constraints are handled during the solution process too can often be far from clear to beginners. Hence it is instructive to go through the solution procedure for solving a complicated real-world problem by applying the BA. We choose to demonstrate this by going through the steps involved in solving the economic dispatch (ED) problem in electrical power systems engineering [10]. The cost function in this problem is nonlinear, multimodal, with discontinuities, subject to numerous inequality and equality constraints as outlined below.

3.1.1 Objective Function

The cost function is assumed to be quadratic, and minimization of the fuel cost of N_g number of power plants in the system is the objective here:

$$\min_{P \in R^{N_g}} F = \sum_{j=1}^{N_g} F_j(P_j) = \sum_{j=1}^{N_g} (a_j + b_j P_j + c_j P_j^2) \tag{9}$$

where $F_j(P_j)$ is the fuel cost of the j th generating unit in \$/hr, P_j is the power generated by the j th generating unit in MW, and a_j , b_j and c_j are cost coefficients of the j th generator.

However, the valve point effect superimposes ripples on this quadratic cost curve, thereby making it multimodal, and the cost curve becomes

$$\min_{P \in R^{N_g}} F = \sum_{j=1}^{N_g} F_j(P_j) = \sum_{j=1}^{N_g} (a_j + b_j P_j + c_j P_j^2) + \left| e_j \sin(f_j(P_j^{\min} - P_j)) \right| \quad (10)$$

where e_j and f_j are the constants of the valve-point effect of generators.

If there are multiple fuel options, the fuel cost of the j th generator is given by

$$F_j(P_j) = \begin{cases} a_{j1} + b_{j1}P_j + c_{j1}P_j^2, \text{ fuel 1, } P_j^{\min} < P_j \leq P_{j1} \\ a_{j2} + b_{j2}P_j + c_{j2}P_j^2, \text{ fuel 2, } P_{j1} < P_j \leq P_{j2} \\ \vdots \\ a_{jk} + b_{jk}P_j + c_{jk}P_j^2, \text{ fuel } k, P_{j,k-1} < P_j \leq P_j^{\max} \end{cases} \quad (11)$$

A generator with k fuel options has k discrete regions.

3.1.2 Optimization Constraints

The power generated has to obviously satisfy the minimum and maximum power generation limits:

$$P_j^{\min} \leq P_j \leq P_j^{\max} \quad (12)$$

While the above inequality constraints are relatively easier to satisfy, the more difficult-to-satisfy equality constraint is that the solution or total power generated P_G must satisfy the total load demand P_D plus the total losses P_L in the system:

$$\sum_{j=1}^{N_g} P_j = P_D + P_L \quad (13)$$

where P_L represents the line losses which is calculated using B-coefficients, given by

$$P_L = \sum_{j=1}^{N_g} \sum_{i=1}^{N_g} P_j B_{ji} P_i + \sum_{j=1}^{N_g} B_{0j} P_j + B_{00} \quad (14)$$

where P_i and P_j are the real power injection at i th and j th buses, respectively, and the B_{ij} 's are the loss coefficients which can be assumed to be constant under normal operating conditions.

3.1.3 Practical Operating Constraints of Generators

3.1.4 Prohibited Operating Zones (POZ)

The prohibited zones arise due to practical operational constraints on generators. The feasible operating zones of unit j can be described as follows:

$$P_j \in \left\{ \begin{array}{l} P_j^{\min} \leq P_j \leq P_{j,1}^l \\ P_{j,k-1}^u \leq P_j \leq P_{j,k}^l, \quad k=2, 3, \dots, n_j, j=1, 2, \dots, n \\ P_{j,n_j}^u \leq P_j \leq P_j^{\max} \end{array} \right. \quad (15)$$

where n_j is the number of prohibited zones of the j th generator. $P_{j,k}^l, P_{j,k}^u$ are the lower and upper power outputs of the k th prohibited zone of the j th generator, respectively.

3.1.5 Ramp Rate Limits

The physical limitations of starting up and shutting down of generators impose ramp rate limits, which are modeled as follows. The increase in generation is limited by

$$P_j - P_j^0 \leq UR_j \quad (16)$$

Similarly, the decrease is limited by

$$P_j^0 - P_j \leq DR_j \quad (17)$$

where P_j^0 is the previous output power, UR_j and DR_j are the up-ramp limit and the down-ramp limit, respectively, of the j th generator.

Combining (16) and (17) with (12) results in the change of the effective operating or generation limits to

$$\underline{P}_j \leq P_j \leq \overline{P}_j \quad (18)$$

where

$$\underline{P}_j = \max(P_j^{\min}, P_j^0 - DR_j) \quad (19)$$

$$\overline{P}_j = \min(P_j^{\max}, P_j^0 + UR_j) \quad (20)$$

Combining this with (15), the ED problem can be formulated as

$$\min_{P \in R^{N_g}} F = \sum_{j=1}^{N_g} F_j(P_j) = \sum_{j=1}^{N_g} (a_j + b_j P_j + c_j P_j^2) + \left| e_j \sin \left[f_j (P_j^{\min} - P_j) \right] \right| \quad (21a)$$

$$\begin{aligned} \text{s.t. } & \sum_{j=1}^{N_g} P_j = P_D + P_L \\ & \max(P_j^{\min}, P_j^0 - DR_j) \leq P_j \leq P_{j,1}^l \\ & P_{j,k-1}^u \leq P_j \leq P_{j,k}^l, \quad k = 2, 3, \dots, n_j, j = 1, 2, \dots, N_g \\ & P_{j,n_j}^u \leq P_j \leq \min(P_j^{\max}, P_j^0 + UR_j) \end{aligned} \quad (21b)$$

3.2 Implementation of the BA to ED Problem

Step 0: The 0th or the first step consists of initialization, which is executed as follows:

- For every solution (or bat or generating unit), generate randomly within the specified limits the generation values.
- For units with POZ, if the randomly generated value falls within the POZ, fix it at the nearest limit that is violated.
- If a unit has ramp-rate limits, the power output is uniformly distributed between the effective lower and upper limits.

Generate N_b number of bats or solutions, each comprising N_g number of generating units:

$$\begin{bmatrix} p_1^1 & p_2^1 & \cdots & p_{N_g}^1 \\ p_1^2 & p_2^2 & \cdots & p_{N_g}^2 \\ \vdots & \vdots & \ddots & \vdots \\ p_1^{N_b} & p_2^{N_b} & \cdots & p_{N_g}^{N_b} \end{bmatrix} = \begin{bmatrix} P^1 \\ P^2 \\ \vdots \\ P^{N_b} \end{bmatrix} \quad (22)$$

Step 1: Calculate the fitness values of all the bats using the objective or fitness function F , in (21a).

Step 2: For i th bat, define pulse frequency f^i using (2).

Step 3: Update the velocity and position (which is a vector of generation values) of each bat using (1) and (3), respectively.

Step 4: Generate a new solution by random walk using (4).

Step 5: Select the fitter of the old and new solutions, with a probability $A^i(t)$, using (7).

Step 6: Update the values of R^i and A^i using (5) and (6), respectively.

Step 7: Check if the effective generation limits and POZ limits are violated. Fix the generation at the limit that is violated. This takes care of the inequality constraints. After this is done, any violation of the power balance equality constraint (13) is dealt with by using a penalty factor approach. By this approach, (21a) is modified to

$$\min_{P \in R^{N_g}} L = F + \lambda \left| \sum_{j=1}^{N_g} P_j - (P_D + P_L) \right| \quad (23)$$

where λ is the penalty coefficient, and a fixed large, positive real number.

Step 8: Repeat steps 1–7 until the maximum number of iterations is reached.

4 Application of the Bat Algorithm to Real-World Problems

Given that the BA is a good metaheuristic algorithm capable of solving complicated optimization problems, it was not long before it was applied to practical real-world problems. These are classified as falling into one of the areas below.

4.1 Structural Optimization

One of the first problems solved by using the BA is the welded beam design problem [26]. The problem has four design variables and two objectives: to minimize both the overall fabrication cost and the end deflection. The noteworthy aspect of the problem is that, since multiple solutions of the same objective function value exist, and an efficient algorithm or optimizer must discover all the solutions that form the Pareto front. The BA successfully solved this problem. Probably the next real-world problem to be solved by the BA is the design of a brushless DC wheel motor [27]. The objective here is to maximize the efficiency, with five design variables, six constraints and seventy-eight nonlinear equations. Other problems in this category that were successfully solved are (a) the design of steel truss structures, a problem of medium to high dimensionality [28], (b) optimal design of steel frames for minimum weight [29], and (c) the design of a shell and tube heat exchanger to optimize the bi-objective fitness function comprising cost and effectiveness [30].

4.2 *Classification and Feature Selection*

Reference [31] solves the problem of classification of high dimensional microarray data sets by a functional link artificial neural network (FLANN) classifier. The BA is used to optimize the weights of the FLANN. The BA has also been used to find both the optimal structure as well as the weights and biases of a neural network used for data classification [32]. In [33], a neural network used for data classification is trained using a BA with multiple co-operative sub-populations, and a chaotic map to preserve the diversity of solutions. Such co-operation, in the place of the usual competition between solutions, is claimed to perform better, for this particular application.

In [34], the multispectral satellite image classification using the BA is shown to be a better performer than the bat-K-means clustering, GA, and PSO methods, both in terms of classification efficiency and time complexity.

Feature selection methods aim to provide as simple a classification model as possible, since feature selection is a high dimensional problem affected by the curse of dimensionality. Using a wrapper approach that consists of a heuristic search of a subspace of all possible feature combinations, and a fitness function that is the classifier's performance, [35] uses a binary BA to maximize classifier performance. Reference [36] proved the ability of the BA to solve a high dimensional, combinatorial optimization problem involving feature selection by modifying the continuous valued basic BA into a binary BA. The solution space comprised Boolean hypercube, and the most informative features had to be selected.

Reference [37] considers image thresholding as a constrained optimization problem, and determines the optimal one- or multi-level, fuzzy entropy based thresholds that are maximized using the BA. The BA is used to optimize the parameters of support vector machine, to reduce classification errors in classification problems in [38].

4.3 *Electrical Power Systems*

The BA has been applied to solve a large number of problems in electrical power systems. Reference [10] solves the ED problem in electrical power systems engineering, using the chaotic BA (in Sect. 2.1), an enhanced version of the basic BA. The problem herein has a cost function that is nonlinear, multimodal, has discontinuities, and has to satisfy numerous inequality and equality constraints as outlined in Sect. 3. Reference [39] solves the same ED problem using chaotic BA as in [10], but using a pseudo-code based algorithm to deal with the power balance equality constraint, instead of the penalty factor approach in [10]. Reference [16] solved an advanced version of the ED problem using the BA with mutation, described in Sect. 2.4. Another application of the BA to the ED problem can be found in [40].

The optimal power flow problem whose objective is to minimize the losses in the presence of power balance equality constraints, and inequality constraints like limits on voltages, real and reactive powers at buses is solved using the BA, without and with an unified power flow controller, a flexible ac transmission system device, in [41].

The objective of load frequency control (LFC) in an electrical power system is to minimize frequency oscillations and tie-line flows due to sharp load changes. Reference [19] solves the multi-area LFC problem using the BA with mutation and crossover, described in Sect. 2.5. A PD-PID controller tuned by the BA is used for solving the multi-area LFC in [42]. The LFC of a two-area system is achieved by using a dual mode gain scheduling of PI controllers, which are tuned by using the BA in [43]. Reference [44] contains the LFC of a two-area system with superconducting magnetic energy storage (SMES) units using model predictive control (MPC) is solved by using the BA, to choose the parameters of the MPC and SMES.

The problem of tuning the parameters of the power system stabilizer (PSS) to minimize system oscillations due to load changes and disturbances is solved using the BA, and compared against the other approaches of the conventional PSS (CPSS) and GA based PSS (GAPSS) in [45]. It found the BA based PSS to be the best performer. The same problem of tuning the parameters of the PSS to minimize oscillations due to disturbances, but for a nonlinear model of the power system comprising single machine connected to an infinite-bus through a transmission line is solved in [46]. It uses the integral of time weighted errors as the cost function.

The optimal phasor measurement unit (PMU) placement to ensure observability of the power system is an NP-hard, combinatorial optimization problem. This problem is solved using a binary bat algorithm hybridized with Taguchi method (TBBA) in [47]. Reference [48] solves the problem of parameter estimation in a power system based on PMU recorded data using a hybridized algorithm comprising the BA and DE, and found that this hybridized algorithm performs better than the other algorithms therein.

Reference [49] uses the binary bat algorithm (in which the solution vector comprises just 1's and 0's) to extract wavelet based fault features for predicting low speed bearing faults. The maximum power point tracking (MPPT) problem, whose objective is to maximize the power output of a photovoltaic array solar panel, is solved by employing a PI controller that is tuned using the BA in [50]. Reference [14] solved the distribution feeder reconfiguration problem, using θ -modified BA (in Sect. 2.3), another enhanced version of the basic BA.

The problem of improvement of power quality has been successfully solved using the BA. The cost function comprising multiple objectives of minimization of total harmonic distortion, initial investment cost, and total fundamental power losses, subject to multiple inequality constraints is minimized by optimal design of a passive power filter, using the BA, in [51].

For the optimal speed control of a brushless dc motor using an online adaptive neuro-fuzzy inference system (ANFIS) controller, the learning parameters of the

ANFIS controller are tuned using the BA in [52]. The position control of a piezoelectric actuator is complicated by the nonlinear and multi-valued mapping between the input and output of the actuator, due to the presence of highly nonlinear hysteresis. Hence a neural network (NN) controller trained by using the BA is proposed in the place of conventional controllers and claimed to perform better in [53].

Using the BA, [54] solves the problem of optimal sizing of the battery energy storage (BES) in a microgrid, taking the fixed and running cost of the distributed generators and running cost of the BES per day as the cost function, subject to power balance equality constraint and a number of inequality constraints including charging and discharging rates of the BES, and operating reserve constraint.

Using the BA, [55] solves the problem of optimal spot pricing in a deregulated electricity market, using the fuel cost as the cost function, and power flows as inequality constraints that have to be satisfied optimally.

Using four different types of mutation for self-adaptive learning mechanism (SALM) along with the basic BA to produce the enhanced BA, for finding the linear supply function equilibrium of generating companies in a competitive electricity market is solved in [18].

4.4 Applications in Other Areas

In the field of arrays in electronic engineering, the design of a linear array antenna to minimize the side lobe level, mutual coupling effect, and null control is solved using the BA in [56]. In the field of process control, the NP-hard problem of minimizing the total production cost, of a process with five different tasks with their own costs is solved using the BA with mutation to improve the exploratory capability in [57].

In aerospace engineering, the challenging, high dimensional optimization problem of three-dimensional path planning for an unmanned combat air vehicle (UCAV), whose objective is to find the minimum cost path, subject to numerous constraints, is solved using the BA with DE mutation and crossover in [58].

In the field of petroleum engineering, the problem of minimizing drilling costs by predicting the rate of penetration (ROP) is solved using the BA. First, the data of simultaneous effect of six variables on the ROP is used to develop a mathematical relationship between the ROP and these variables. Next, the BA is used to determine the optimal values of these variables in [59]. Maximization of net present value (NPV) by optimal placement of wells for oil production is solved by the BA, and compared with the other algorithms of GA and PSO, in [60].

In the field of nuclear engineering, a fitness factor that involves maximization of the multiplication factor and minimization of the power peaking factor of a nuclear reactor core is maximized using the BA in [61].

Reference [62] solves the problem of minimizing the energy consumption of a heating, ventilation, and air conditioning (HVAC) system. Using a cubic cost curve

to define the total power consumption of the daily optimal chiller loading (DOCL), the minimum DOCL subject to cooling load balance equality constraint, and lower/upper limits on chillers as inequality constraints is solved by the BA.

The examples and case studies cited herein are merely a typical sample of the real-world problems solved by the BA. Given the popularity of the BA, the interested reader of any background is probably quite likely to find applications of interest within their own fields.

5 Conclusion

As one of the better performing metaheuristic algorithms, the bat algorithm has been applied to solve numerous challenging real-world problems that are not that easily solvable by conventional calculus-based methods.

There is immense scope for further research on the bat algorithm. The BA has been applied to numerous continuous optimization problems. However, its application to solving combinatorial optimization problems like the traveling salesman problem have been far fewer in number. Another area in which the BA has been relatively untested is the solution of large scale optimization problems.

At present, the user has to tune the parameters of loudness $A^i(t)$ and pulse emission rate $R^i(t)$ to suit the problem being solved. An alternate, ideal solution would be to equip the algorithm with self-tuning capabilities so that these parameters are automatically adjusted to suit the problem being solved. This too has not attracted enough research at present.

Since metaheuristic algorithms work on the Darwinian principle of selection of the fittest, other types of selection like rank based selection could be experimented with, instead of the knockout or tournament type of selection described in this chapter.

Given that numerous metaheuristic algorithms exist at present, another direction for further research would be the hybridization of the bat algorithm with the operators of other metaheuristic algorithms, so that the hybrid would combine the best features of the algorithms being combined.

Another kind of hybridization that could be explored profitably is that between the bat algorithm and gradient based methods, eliminating the limitations and combining the strengths of these two contrasting types of optimization methods.

Some steps have been taken in many of the directions indicated here. However, fundamental research that addresses these difficult, core issues in optimization theory requires research at a different level altogether, rather than simple comparisons of performance between some x and y algorithms on some specific problem or set of problems. In that sense, it is hoped that this chapter can inspire more research in the foreseeable future.

References

1. Goldberg, D.E.: Genetic Algorithms in Search, Optimization and Machine Learning. Pearson Education, India (1989)
2. Goldberg, D.E.: Computer-aided gas pipeline operation using genetic algorithms and rule learning (Doctoral dissertation, University of Michigan). Dissertation Abstracts International, 44(10), 3174B (University Microfilms No. 8402282) (1983)
3. Dorigo, M.: Optimization, Learning and Natural Algorithms, PhD thesis, Politecnico di Milano, Italy (1992)
4. Kennedy, J., Eberhart R.: Particle swarm optimization. In: Proceedings of IEEE International Conference on Neural Networks, pp. 1942–1948 (1995)
5. Gandomi, A.H., Alavi, A.H.: Krill herd: a new bio-inspired optimization algorithm. Commun. Nonlinear Sci. Numer. Simul. **17**(12), 4831–4845 (2012)
6. Yang, X.-S.: A new metaheuristic bat-inspired algorithm. Nature Inspired Cooperative Strategies for Optimization (NICSO) **284**, 65–74 (2010)
7. Yang, X.S., Gandomi, A.H.: Bat algorithm: a novel approach for global engineering optimization. Eng. Comput. **29**(5), 464–483 (2012)
8. Gandomi, A.H., Yang, X.S., Alavi, A.H., Talatahari, S.: Bat algorithm for constrained optimization tasks. Neural Comput. Appl. **22**(6), 1239–1255 (2013)
9. Yang, X.S., Cui, Z., Xiao, R., Gandomi, A.H., Karamanoglu, M. (eds.): Swarm Intelligence and Bio-Inspired Computation: Theory and Applications. Elsevier, Waltham, MA (2013)
10. Adarsh, B.R., Raghunathan, T., Jayabarathi, T., Yang, X.-S.: Economic dispatch using chaotic bat algorithm. Energy **96**, 666–675 (2016)
11. Gandomi, A.H., Yang, X.S.: Chaotic bat algorithm. J. Comput. Sci. **5**(2), 224–232 (2014)
12. Jordehi, A.R.: Chaotic bat swarm optimisation (CBSO). Appl. Soft Comput. **26**, 523–530 (2015)
13. Chakri, A., Kehlif, R., Benouaret, M., Yang, X.-S.: New directional bat algorithm for continuous optimization problems. Expert Syst. Appl. **69**, 159–175 (2017)
14. Kavousi-Fard, A., Niknam, T., Fotuhi-Firuzabad, M.: A novel stochastic framework based on cloud theory and θ -modified bat algorithm to solve the distribution feeder reconfiguration. IEEE Trans. Smart Grid **7**(2), 740–750 (2016)
15. Haupt, R.L., Haupt, S.E.: Practical genetic algorithms. Wiley, (2004)
16. Niknam, T., Azizipanah-Abarghooee, R., Zare, M., Bahmani-Firouzi, B.: Reserve constrained dynamic environmental/economic dispatch: a new multiobjective self-adaptive learning bat algorithm. IEEE Syst. J. **7**(4), 763–776 (2013)
17. Wang, G., Guo, L., Duan, H., Liu, L., Wang, H.: A bat algorithm with mutation for UCAV path planning. Sci. World J. (2012)
18. Niknam, T., Sharifinia, S., Azizipanah-Abarghooee, R.: A new enhanced bat-inspired algorithm for finding linear supply function equilibrium of GENCOs in the competitive electricity market. Energy Convers. Manag. **76**, 1015–1028 (2013)
19. Khooban, M.H., Niknam, T.: A new intelligent online fuzzy tuning approach for multi-area load frequency control: self adaptive modified bat algorithm. Int. J. Electr. Power Energy Syst. **71**, 254–261 (2015)
20. Raghunathan, T., Ghose, D.: An online-implementable differential evolution tuned all-aspect guidance law. Control Eng. Prac. **18**(10), 1197–1210 (2010)
21. Raghunathan, T., Ghose, D.: Differential evolution based 3-D guidance law for a realistic interceptor model. Appl. Soft Comput. **16**, 20–33 (2014)
22. Fister Jr., I., D. Fister, and X.-S. Yang. A hybrid bat algorithm. [arXiv:1303.6310](https://arxiv.org/abs/1303.6310) (2013)
23. Xie, J., Zhou Y., Chen, H.: A novel bat algorithm based on differential operator and Lévy flights trajectory. Computat. Intell. Neurosci. (2013)
24. Jun, L., Liheng, L., Xianyi, W.: A double-subpopulation variant of the bat algorithm. Appl. Math. Comput. **263**, 361–377 (2015)

25. Meng, X.B., Gao, X.Z., Liu, Y., Zhang, H.: A novel bat algorithm with habitat selection and Doppler effect in echoes for optimization. *Expert Syst. Appl.* **42**(17), 6350–6364 (2015)
26. Yang, X.S.: Bat algorithm for multi-objective optimisation. *Int. J. Bio-Inspired Comput.* **3**(5), 267–274 (2012)
27. Bora, T.C., Coelho, L.D.S., Lebensztajn, L.: Bat-inspired optimization approach for the brushless DC wheel motor problem. *IEEE Trans. Magn.* **48**(2), 947–950 (2012)
28. Hasançebi, O., Teke, T., Pekcan, O.: A bat-inspired algorithm for structural optimization. *Comput. Struct.* **128**, 77–90 (2013)
29. Hasançebi, O., Carbas, S.: Bat inspired algorithm for discrete size optimization of steel frames. *Adv. Eng. Softw.* **67**, 173–185 (2014)
30. Tharakeshwar, T.K., Seetharamu, K.N., Prasad, B.D.: Multi-objective optimization using bat algorithm for shell and tube heat exchangers. *Appl. Therm. Eng.* **110**, 1029–1038 (2017)
31. Mishra, S., Shaw, K., Mishra, D.: A new meta-heuristic bat inspired classification approach for microarray data. *Procedia Technol.* **4**, 802–806 (2012)
32. Jaddi, N.S., Abdullah, S., Hamdan, A.R.: Optimization of neural network model using modified bat-inspired algorithm. *Appl. Soft Comput.* **37**, 71–86 (2015)
33. Jaddi, N.S., Abdullah, S.S., Hamdan, A.R.: Multi-population cooperative bat algorithm-based optimization of artificial neural network model. *Inf. Sci.* **294**, 628–644 (2015)
34. Senthilnath, J., Kulkarni, S., Benediktsson, J.A., Yang, X.S.: A novel approach for multispectral satellite image classification based on the bat algorithm. *IEEE Geosci. Remote Sens. Lett.* **13**(4), 599–603 (2016)
35. Rodrigues, D., Pereira, L.A., Nakamura, R.Y., Costa, K.A., Yang, X.S., Souza, A.N., Papa, J. P.: A wrapper approach for feature selection based on bat algorithm and optimum-path forest. *Expert Syst. Appl.* **41**(5), 2250–2258 (2014)
36. Nakamura, R.Y., Pereira, L.A., Costa, K.A., Rodrigues, D., Papa, J.P., Yang X.-S.: BBA: a binary bat algorithm for feature selection. In: 2012 25th SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI), pp. 291–297. IEEE (2012, August)
37. Ye, Z.W., Wang, M.W., Liu, W., Chen, S.B.: Fuzzy entropy based optimal thresholding using bat algorithm. *Appl. Soft Comput.* **31**, 381–395 (2015)
38. Tharwat, A., Hassanien, A.E., Elnaghi, B.E.: A BA-based algorithm for parameter optimization of support vector machine. *Pattern Recogn. Lett.* **93**, 13–22 (2017)
39. Shukla, A., Singh S.N.: Pseudo-inspired CBA for ED of units with valve-point loading effects and multi-fuel options. *IET Gener. Transm. Distrib.* **11**(4), 1039–1045 (2017)
40. Hosseini, S.S.S., Yang, X.S., Gandomi, A.H., Nemati, A.: Solutions of non-smooth economic dispatch problems by swarm intelligence. In: Fister, I., Fister Jr., I. (eds.) *Adaptation and hybridization in computational intelligence*, pp. 129–146. Springer International Publishing, Switzerland (2015)
41. Rao, B.V., Kumar, G.N.: Optimal power flow by BAT search algorithm for generation reallocation with unified power flow controller. *Int. J. Electr. Power Energy Syst.* **68**, 81–88 (2015)
42. Dash, P., Saikia, L.C., Sinha, N.: Automatic generation control of multi area thermal system using Bat algorithm optimized PD–PID cascade controller. *Int. J. Electr. Power Energy Syst.* **68**, 364–372 (2015)
43. Sathya, M.R., Ansari, M.M.T.: Load frequency control using Bat inspired algorithm based dual mode gain scheduling of PI controllers for interconnected power system. *Int. J. Electr. Power Energy Syst.* **64**, 365–374 (2015)
44. Elsis, M., Soliman, M., Aboelela, M.A.S., Mansour, W.: Optimal design of model predictive control with superconducting magnetic energy storage for load frequency control of nonlinear hydrothermal power system using bat inspired algorithm. *J. Energy Storage* **12**, 311–318 (2017)
45. Ali, E.S.: Optimization of power system stabilizers using BAT search algorithm. *Int. J. Electr. Power Energy Syst.* **61**, 683–690 (2014)

46. Sambariya, D.K., Prasad, R.: Robust tuning of power system stabilizer for small signal stability enhancement using metaheuristic bat algorithm. *Int. J. Electr. Power Energy Syst.* **61**, 229–238 (2014)
47. Basetti, V., Chandel, A.K.: Optimal PMU placement for power system observability using Taguchi binary bat algorithm. *Measur.* **95**, 8–20 (2017)
48. Rashidi, F., Abiri, E., Niknam, T., Salehi, M.R.: On-line parameter identification of power plant characteristics based on phasor measurement unit recorded data using differential evolution and bat inspired algorithm. *IET Sci. Meas. Technol.* **9**(3), 376–392 (2015)
49. Kang, M., Kim, J., Kim, J.M.: Reliable fault diagnosis for incipient low-speed bearings using fault feature analysis based on a binary bat algorithm. *Inf. Sci.* **294**, 423–438 (2015)
50. Oshaba, A.S., Ali, E.S., Elazim, S.A.: MPPT control design of PV system supplied SRM using BAT search algorithm. *Sustain. Energy, Grids and Netw.* **2**, 51–60 (2015)
51. Yang, N.C., Le, M.D.: Optimal design of passive power filters based on multi-objective bat algorithm and pareto front. *Appl. Soft Comput.* **35**, 257–266 (2015)
52. Premkumar, K., Manikandan, B.V.: Speed control of Brushless DC motor using bat algorithm optimized adaptive neuro-fuzzy inference system. *Appl. Soft Comput.* **32**, 403–419 (2015)
53. Svec̆ko, R., Kusić, D.: Feedforward neural network position control of a piezoelectric actuator based on a BAT search algorithm. *Expert Syst. Appl.* **42**(13), 5416–5423 (2015)
54. Bahmani-Firouzi, B., Azizpanah-Abarghoee, R.: Optimal sizing of battery energy storage for micro-grid operation management using a new improved bat algorithm. *Int. J. Electr. Power Energy Syst.* **56**, 42–54 (2014)
55. Murali, M., Kumari, M.S., Sydulu, M.: Optimal spot pricing in electricity market with inelastic load using constrained bat algorithm. *Int. J. Electr. Power Energy Syst.* **62**, 897–911 (2014)
56. Das, A., Mandal, D., Ghoshal, S.P., Kar, R.: An efficient side lobe reduction technique considering mutual coupling effect in linear array antenna using BAT algorithm. *Swarm Evol. Comput.* (2017)
57. Wang, J., Fan, X., Zhao, A., Yang, M.: A hybrid bat algorithm for process planning problem. *IFAC-PapersOnLine* **48**(3), 1708–1713 (2015)
58. Wang, G.G., Chu, H.E., Mirjalili, S.: Three-dimensional path planning for UCAV using an improved bat algorithm. *Aerosp. Sci. Technol.* **49**, 231–238 (2016)
59. Moraveji, M.K., Naderi, M.: Drilling rate of penetration prediction and optimization using response surface methodology and bat algorithm. *J. Nat. Gas Sci. Eng.* **31**, 829–841 (2016)
60. Naderi, M., Khomehchi, E.: Well placement optimization using metaheuristic bat algorithm. *J. Petrol. Sci. Eng.* **150**, 348–354 (2017)
61. Kashi, S., Minuchehr, A., Poursalehi, N., Zolfaghari, A.: Bat algorithm for the fuel arrangement optimization of reactor core. *Ann. Nuc. Energy* **64**, 144–151 (2014)
62. dos Santos Coelho, L., Askarzadeh, A.: An enhanced bat algorithm approach for reducing electrical power consumption of air conditioning systems based on differential operator. *Appl. Therm. Eng.* **99**, 834–840 (2016)