4

Bat Algorithm – Modifications and Application

Neeraj Kanwar
Department of Electrical Engineering
Manipal University Jaipur, Jaipur, India

Nand K. Meena
School of Engineering and Applied Science
Aston University, Birmingham, United Kingdom

Jin Yang
School of Engineering and Applied Science
Aston University, Birmingham, United Kingdom

CONTENTS

4.1 Introduction

With the beginning of fast computational services, a large number of evolutionary or swarm-based artificial intelligence (AI) techniques gained the attraction to solve real-life complex combinatorial optimization problems. However, these techniques have their own qualities and shortcomings. BA is a recently developed optimization technique proposed by Xin-She Yang [1] in 2010. This is a bio-inspired search technique which has shown an advanced competency to reach into a promising region. It is inspired by the social behavior of bats and the phenomenon of echolocation to sense distance between the bat's current location and prey. It is a simple, easy to implement, significantly faster than other algorithms, and robust numerical optimization approach [2]. It has been used in literature to solve several optimization problems of diverse nature such as: distributed resources allocation, feature selection, economic load dispatch, load frequency control, distribution network reconfiguration (DNR), etc. It is very suitable for highly nonlinear problems and can generate optimal solutions with good accuracy for such type of problem. So, it has good exploitation potential, but its exploration is insufficient to reach the promising region. It happens due to its local trapping, insufficient diversity or slow movement of the algorithm. Therefore, several attempts have been made to improve the performance of the standard BA to overcome its inherent flaws.

DNR is a frequently required process in a distribution system. It is done by managing the open and closed status of sectionalizing and tie-switches. This switch changing process basically alters the topological structure of the distribution network. It reallocates load from one distribution feeder to another and balances the load on the system. It also improves the performance of the distribution network by reducing losses, improving voltage profile, etc. Thus, this is a very important operational strategy that can enhance distribution network performance. In this chapter, the application of BA is shown to solve a complex optimal DNR problem of a distribution network.

The chapter is organized as follows. In Section 4.2 an introduction of the standard BA algorithm is given. The working of the standard BA is also presented in brief. In Section 4.3, some modifications suggested in literature to improve the performance of BA are presented as improved BA, bat algorithm with centroid strategy, self-adaptive BA, chaotic mapping based BA, self-adaptive BA with step-control and mutation mechanisms, adaptive position update, smart BA, adaptive weighting function and velocity. In Section 4.4, the application of the BA is presented for the optimal DNR problem of a distribution network.

4.2 Original bat algorithm in brief

BA is a new bio-inspired algorithm that mimics the bats' behavior for searching their prey. Bats use the echolocation phenomenon to sense proximity of prey. In BA, all bats randomly fly in different directions with some definite velocity and frequency at a certain position. Two control parameters are used to guide the bats' movement in random directions i.e. loudness and PER. Bats update their velocity and position by assigning time varying values to these parameters. Bats emit pulses and adjust PER depending on the vicinity of the prey. To characterize the behavior of foraging bats the optimization model is developed. Foraging behavior of bats is modeled in two steps i.e. random fly and local random walk (LRW) which are briefly explained below [3].

4.2.1 Random fly

Each bat is defined by its position x_b, velocity v_b, frequency f_b, loudness A_b, and PER r_b in a D-dimensional problem search space. The velocity and position of the bth bat at the tth iteration are updated using (4.1)–(4.3). These equations show the global searching process of BA.

$$f_b = f_{\min} + (f_{\max} - f_{\min})\beta \tag{4.1}$$

$$v_b^t = v_b^{t-1} + (x_b^t - x^*)f_b \tag{4.2}$$

$$x_b = x_b^{t-1} + v_b^t \tag{4.3}$$

Here β denotes a random number in range [0, 1] which is obtained from uniform distribution, x^* is current best solution, $f_{\max}$ and $f_{\min}$ are maximum and minimum frequency range.

4.2.2 Local random walk

A new solution is generated for each bat by using LRW, expressed as

$$x_{new,b} = x_{old,b} + \epsilon\left\langle A_b^t\right\rangle \tag{4.4}$$

where $\epsilon \in$ [-1, 1] is a uniformly distributed random number, $\left\langle A_b^t\right\rangle$ is the average loudness value of all bats and r_b^t is the PER of the bth bat. Loudness and PER of each bat are updated with iteration using (4.5) and (4.6).

$$A_b^{t+1} = \alpha A_b^t \tag{4.5}$$

$$r_b^{t+1} = r_b^0\Big[1 - exp(-\gamma t)\Big] \tag{4.6}$$

where α and γ are constants having value of 0.9. r_b^0 shows the maximum possible PER. As the algorithm proceeds, loudness tends to zero and PER become maximum. The pseudo-code of standard BA is presented in Algorithm 5.

Algorithm 5 Pseudo-code of the original bat algorithm.

```
Determine the objective function OF(.)
Initialize the bat population x_b, v_b and f_{b,d}, ∀ b = 1, 2, ..., N_b, d = 1, 2, 3, ..., D
Initialize pulse rate r_b and loudness A_b
Set t = 0
while t < itr_max do
    Generate new solutions by adjusting frequency, updating velocities, and positions by using (4.1)–(4.3).
    if rand > r_b then
        Select the current best solution x*
        Generate a local solution OF(x_b) around x*
    end if
    Generate a new solution by flying randomly
    if rand < Ab and OF(x_b) < OF(x*) then          ▷ for minimization
        Accept the new solutions
        Increase r_b and reduce A_b
    end if
    Rank the bats and find the current best
    t = t + 1
end while
return the best bat and its fitness as a result
```

4.3 Modifications of the bat algorithm

Standard BA shows distinct performance while dealing with lower-dimensional optimization problems but performance degrades as the dimension of the problem increases. It all happens due to lack of diversity which ultimately drags BA to local optima and thus degrades convergence. Therefore, different modifications are suggested in literature to refine its exploration and exploitation potentials. Some of the modifications suggested by different authors are described in the following sections.

4.3.1 Improved bat algorithm

In [4] an improved BA (IBA) is proposed to adjust the loudness and PER in a better way and to enhance LRW of bats to refine its exploration and exploitation potentials, respectively. Furthermore, additional diversity in population is suggested to cope with the intense exploitation capability of BA. The first modification is self-adapted loudness and PER. Each bat is allowed to vary in accordance to the loudness assigned to it and are not allowed to vary too quickly. The gradual switching of exploration to exploitation is governed by the values that are assigned to loudness and PER. In this way, these two parameters become self-adapted for each bat. The suggested modeling for loudness and PER is given in (4.7) and (4.8).

$$A_b^t = \alpha^t A_b \tag{4.7}$$

$$r_b^t = 1 - A_b^t \tag{4.8}$$

where, $\alpha = \left(\frac{1}{2A_b}\right)^{\frac{1}{k*iter_{\max}}}$, and k is the desired fraction of maximum iteration at which loudness and PER equalize. In this manner both parameters become self-adaptive to each other.

The second modification is improved local random walk (ILRW) which is suggested to improve LRW of bats. First M_c number of replicas of the current best bat are generated and then each of them is mutated in the range [-1, 1] at randomly selected dimension. ILRW initiates only when the selected random number is less than PER, otherwise LRW will be performed as in the standard BA. Thus ILRW confirms the current best bat LRW during the evolutionary process and helps to maintain an appropriate balance between exploitation and exploration of the search space. If a better solution is found by ILRW, it replaces the current best bat.

BA performance also degrades when its intense exploitation capability is merged with its inherent poor exploration potential. It happens especially for large-scale optimization problems. It occurs because bats may remain busy to exploit the undesirable search space and be unable to identify the promising region. This causes local trapping of the algorithm. The mutation operator of GA has intense potential to discover a new area in the problem search space. As BA has very poor diversity, a very high mutation rate is required. Therefore the population is reinitialized in each iteration using mutation. Fitness is evaluated for all mutated bats and the current best bat is updated by the better mutated bat if obtained.

4.3.2 Bat algorithm with centroid strategy

In standard BA, the global and local search patterns are such that it blocks some neighboring area and limits the search ability. To overcome these flaws of BA, a new centroid strategy is proposed in [5]. Three different centroid strategies are proposed with six different designs. The main idea is to consider all the effects from the best position and neighbor bats. The local search capability is improved using the proposed strategy. A detailed comparison of six different centroid strategies is provided to give deep insight. These include arithmetic centroid, geometric centroid, harmonic centroid, weighted arithmetic centroid, weighted geometric centroid and weighted harmonic centroid.

4.3.3 Self-adaptive bat algorithm (SABA)

Fister *et al.* [6] presented the self-adaptive bat algorithm (SABA) in which the parameters (loudness and PER) at the beginning of the search process change as the process reaches maturity. It overcomes the difficulties of setting parameter values at the beginning and in different searching phases. A self-adaptive feature is also employed in [7] to improve algorithm performance. It enables a self-adaptation of its control parameters and gives the advantages that

1. the control parameters need not to be set before starting the algorithm.
2. the control parameters are self-adapted during the fitness evaluation

4.3.4 Chaotic mapping based BA

The chaotic mapping mechanism is a new optimization method to extend the searching range of some evolutionary algorithms to eliminate their premature convergence [8]. Three search algorithms are proposed in [8] for chaotic mapping. The first sub-algorithm is the global chaos traversal disturbance algorithm. In this algorithm, traversal search is performed in the region of feasible solutions. It helps to overcome the declining diversity of the BA population. The second sub-algorithm is the local niche accelerate search algorithm. It is suggested to improve local search in the anaphase of the algorithm. It increases the searching speed of BA. First the optimal solution is set as the center of a local niche acceleration searching area. Then, each component is mapped in interval [0, 1]. Search radius is calculated to set the local niche acceleration. Then, a traversal search is conducted. At the end, the optimal solution, loudness, PER and frequency are updated. The third sub-algorithm is adaptive speed control algorithm. It is suggested to modify conversion of bats between global and local search. It first defines the population gathering rate and then based on it updates the weight of the speed between global and local search.

4.3.5 Self-adaptive BA with step-control and mutation mechanisms

PER increases and loudness decreases with the distance from the prey. It is done to observe the movement of prey and to remain undetected by prey. In the early stage of BA, the search process is mainly affected by loudness and help in global search, whereas during anaphase of the algorithm, the search process is more influenced by PER. Therefore, the global and local search probability of BA should be guided by loudness and PER. These characteristics inspired Shilei Lyu et al. to propose a self-adaptive BA (SABA) [9]. Two modifications are proposed i.e. step-control and mutation mechanisms.

In SABA, step sizes are controlled by step-control during the search process. It is different from standard BA. It uses two frequencies f_1 and f_2 to control step sizes during global and local search. New modeling is suggested for velocity, frequency and position as given by (4.9)–(4.12).

$$v_b^t = \omega v_b^{t-1} + f_1 r_1 \big(h_b^* - x_b^{t-1}\big) \times f_2 r_2 \big(x^* - x_b^{t-1}\big) \tag{4.9}$$

$$f_1 = \alpha\Big(1 - exp\big(-\left|F_{avg} - F_{best}\right|\big)\Big) + \gamma\big(1-k\big) + f_{\min} \tag{4.10}$$

$$C_w = f_1 + f_2 \tag{4.11}$$

$$x_b^t = x_b^{t-1} + \mu v_b^t \tag{4.12}$$

where ω is the weight coefficient, h_b^* is the optimal solution for bth bat, x^* is the current best bat, f_1 and f_2 are the frequencies, and r_1 and r_2 are uniform random numbers in the range of [0.5, 1.5]. F_{avg} is the average fitness value for current optimal solutions and F_{best} is the current best fitness. $F_{\min}$ is a constant value to represent minimum value of f_1. k is the evaluation index. α, γ, and μ are weight coefficients.

The mutation mechanism is proposed to avoid local trapping. In this mechanism, PER and loudness are improved by suggesting (4.13).

$$A_b^{t+1} = \frac{f_1}{f_{\max}}; \qquad \text{and} \qquad R_b^{t+1} = \frac{f_2}{f_{\max}} \tag{4.13}$$

SABA is effective to avoid local trapping during the early stage of the algorithm and has ability to improve accuracy during anaphase of the algorithm.

4.3.6 Adaptive position update

An adaptive position update based BA (APU-BA) is proposed [10] to improve the search performance and accuracy of the BA. The local search part is upgraded by assigning values for the BA position update index. It is assigned to each bat per dimension. Also a discrete layer peeling algorithm is combined with APU-BA to improve its convergence. This modification improves the performance of APU-BA compared to standard BA.

4.3.7 Smart bat algorithm

In [11], smart BA is proposed with an idea to make an artificial bat that regulates its searching process using fuzzy logic and decision theory. It provides better search direction, frequency and velocity to the artificial bat. A utility function is defined for each bat to decide its search direction. It is made up of two parts; the first part consists of difference between this artificially generated bat and the best bat, whereas the second one shows the ratio of the bat number threshold of a cluster to the number of bats in the cluster. It helps to avoid local trapping.

4.3.8 Adaptive weighting function and velocity

A modified version of (4.1) is proposed in [12] to improve the exploration and exploitation potential of BA.

$$v_b^t = \omega v_b^{t-1} + f_b^t\big(x_b^t - x^*\big)k_1 + f_b^t\big(x_b^t - x_{cbest}\big)k_2 \tag{4.14}$$

$$k_1 + k_2 = 1 \tag{4.15}$$

$$k_1 = 1 + (k_{init} - 1)\frac{(iter_{\max} - iter)}{iter_{\max}^n} \tag{4.16}$$

where k_1 and k_2 are weighting factors, k_{init} is the initial value of k_1, ω shows positive weighting factor. This modification shows a balance between global and local search. An adaptive weighting function is also proposed as given by (4.17).

$$\Omega_b^t = \omega_b^0 e^{-\psi t} \tag{4.17}$$

Here, ψ represents a positive value of fixed quantity inspired by the Lyapunov stability theorem. Weighting function controls the gap between obtained solution and expected solution.

State error between expected solution and current bth bat is defined by (4.18). The velocity equation is modified using the state error equation. When bats come near the final solution, velocity reduces and bats move slowly towards the final solution. It ultimately improves convergence of BA by reducing fluctuation around the final solution.

$$error_b^t = x_{\exp} - x_b^{t-1} \tag{4.18}$$

4.4 Application of BA for optimal DNR problem of distribution system

In this section, the BA is used to solve the real-life engineering optimization problem.

4.4.1 Problem description

DNR is a well-known and effective operational strategy used to improve performance of modern automated radial distribution systems. The problem of DNR is characterized as finding the best possible topology of the distribution network that can give minimum losses and better node voltage profile in the system. Here, the DNR problem is formulated to reduce line losses while improving node voltages. Different operating constraints should be satisfied while solving the objective function such as voltage limits, feeder current capacity and feasible radial arrangement of the distribution system.

4.4.2 How can the BA algorithm be used for this problem?

We want to generate an optimization method to give optimal radial topology in the distribution system. The optimal topology is represented by optimal switches that are to be open to generate the optimal configuration of a given distribution network. In this chapter, the standard version of BA is used to solve this problem. All optimization techniques require some algorithm-specific and some common control parameters. Algorithm-specific parameters are different for each algorithm. The common control parameters are number of individuals and maximum number of iterations to solve an optimization problem. However, optimum number of these control parameters are found out by trade-off.

BA is a nature inspired algorithm. All bats in a swarm use echolocation in the searching process for their food and to avoid obstacles. They emit pulses in their surroundings and based on return echoes they are able to locate their prey. In addition, bats also identify the most nutritious area to move on. DNR is a complex, combinatorial optimization problem, therefore, so many combinations of switches are possible. Each bat has a defined position i.e. x_b. Initially we define the different control parameters i.e. dimension of the search space (d), total number of bats in the population (N_b), maximum number of iterations (N_{iter}), design constants (α, γ), pulse loudness (A), maximum and minimum pulse frequencies ($f_{\max}/f_{\min}$), initial pulse emission rate (r_b^0).

The initial swarm of bats is randomly generated using a defined problem and algorithm specific control parameters. It consists of N_b bats (each bat is a d-dimensional vector). The dimensions depend upon design variables considered for the optimal DNR problem in a distribution system. The design variables in each bat is switches that are randomly selected in a given range. The maximum range is the system size for which the DNR problem is solved. Each bat is generated using an initially defined frequency, velocity, loudness and pulse rate. Each bat in the swarm represents a possible solution of the objective function. After initialization, fitness is evaluated for each bat. Each individual represents a combination of switches corresponding to some fitness value of objective function. Here objective function is formulated for power loss reduction. So the combination of switches in each bat has some value of power loss. The fitness value of each bat is calculated and the best one is saved.

Initially bats fly in random directions with some initial velocity, fixed frequency and loudness. The wavelength of emitted pulses is automatically adjusted using control equations. The frequency and rate of emitted pulses are selected in a defined range. Frequency is generated in between minimum and maximum defined frequency. The pulses emission rate is selected in the range of [0, 1]. Loudness varies from large to minimum value. Bats movement in the random direction is modeled by (4.1) to (4.3). For better exploration and exploitation of search space both loudness and pulse emission rate are varied during each iteration using (4.5) and (4.6). The loudness of each bat is decreased and pulse emission rate is increased after each bat updates its current position using equation (4.4). Rank the bats and update the current best solution. If the termination criterion is satisfied, it generates the final results. When this solution is implemented on network, it will give minimum power loss. This solution shows the optimal topology of the distribution network after the first iteration. The iterative process continues until the maximum iteration counts are exhausted.

4.4.3 Description of experiments

By interchanging sectionalizing and tie switches of any distribution network, a new topology can be obtained. In BA, the final optimal solution shows an optimal configuration of switches. This configuration causes reduction in power losses and node voltage deviations. Therefore, the optimal DNR problem is formulated to minimize power losses while maintaining better node voltage profiles in the distribution system. In order to limit the voltage deviation at different nodes, a hard node voltage constraint is used as desired. Similarly to check the current carrying capacities of distribution feeders, a feeder ampacity constraint is essential. The problem is structured as a single-objective constrained optimization problem where optimal radial topology is obtained. The objective function $f(x)$ is formulated as:

$$\text{Min } P_{Loss} = \sum_{j=1}^{N_L} R_j \frac{P_{j,k}^2 + Q_{j,k}^2}{V_{j,k}^2} \tag{4.19}$$

This objective function is solved using the standard BA technique, subject to different network and operational constraints as defined by (4.20)–(4.23). Power flow constraints are defined by (4.20) that represents a set of power flow equations. Equation (4.21) shows a node voltage limit constraint. Node voltages V_n of all system buses must be kept within the minimum and maximum permitted limits i.e. $V_{\min}$ and $V_{\max}$, respectively, during the optimization process. The current flow in each branch must satisfy the rated ampacity of each branch as defined by the branch current constraint of (4.22). In addition to these operational constraints, the radial topology constraint is also defined to solve the optimal DNR problem. The reconfigured network topology must always be radial. It means it should not have any closed path or loop. The radiality constraint for the Yth radial topology is defined and shown by (4.23).

$$g(h) = 0 \tag{4.20}$$

$$V_{\min} \leq V_n \leq V_{\max} \qquad \forall\ n \in N \tag{4.21}$$

$$I_{j,k} \leq I_j^{\max} \qquad \forall\ j, k \tag{4.22}$$

$$\Phi_j(Y) = 0 \tag{4.23}$$

Here, N, N_L, $P_{j,k}$, $Q_{j,k}$, $V_{j,k}$, $I_{j,k}$, $I_j^{\max}$ denote the total number of nodes and branches in the system, real and reactive power flows in line j, node voltage at the sending end of line j, present and maximum current carrying capacity of line j respectively, all at load level k.

4.4.4 Results

The benchmark 33-bus [13] test distribution system is used to solve this optimal DNR problem. It is a 12.66 kV system with 32 sectionalizing and 5 tie-switches (i.e 33-37). The nominal active and reactive demand of this system is 3.175 MW and 2.3 MVAr, respectively. The swarm size and maximum iterations in BA are set at 10 and 200, respectively. The best result obtained after 100 independent trials of BA is presented in Table 4.1. The annual load profile is segmented into three different load levels, i.e., light, nominal and peak to show 50%, 100% and 160% of the nominal system loading, respectively. The corresponding load durations are taken as 2000, 5260 and 1500 hours, respectively. The objective function value is calculated at each load level separately. The table shows the optimal configuration obtained at all three load levels. The network performance is improved at each load level when compared with base configuration. After applying this solution an annual loss reduction of about 32.24% can be obtained. This shows that a significant reduction in annual energy losses can be obtained by optimal DNR.

TABLE 4.1
Optimal solution for DNR using standard bat algorithm.

Cases	Particulars	Load Levels			Annual energy loss reduction
		L	N	P	
Case 1	Base configuration	33, 34, 35, 36, 37	33, 34, 35, 36, 37	33, 34, 35, 36, 37	–
	P_{Loss}	47.06	202.67	575.39	
	$V_{min}(p.u.)\,\forall\, n$	0.9583	0.9131	0.8528	
Case 2	Optimal configuration	7, 9, 14, 32, 37	7, 9, 14, 32, 37	7, 9, 14, 32, 37	32.24%
	P_{Loss}	33.27	139.52	379.95	
	$\Delta P_{Loss}(\%)$	29.30	31.16	33.95	
	$V_{min}(p.u.)\,\forall\, n$	0.9698	0.9378	0.8968	

4.5 Conclusion

In this chapter the Bat Algorithm is presented in detail. Some modifications suggested by different authors to improve the performance of BA are demonstrated and discussed. Finally, the application of the algorithm to a complex combinatorial problem of optimal DNR is shown. Power losses at different load levels are calculated and compared with the base case. The application results of the algorithm show that there is a significant improvement in the desired objectives.

Acknowledgement

This work was supported by the Engineering and Physical Sciences Research Council (EPSRC) of United Kingdom (Reference Nos.: EP/R001456/1 and EP/S001778/1).

References

1. X. S. Yang, Bat algorithm for multi-objective optimization, International Journal of Bio-Inspired Computation (2011) vol. 3(5), pp. 267-274.
2. T. Niknam, R. A. Abarghooee, M. Zare, and B. B. Firouzi, Reserve constrained dynamic environmental/economic dispatch: a new multiobjective self-adaptive learning Bat Algorithm, IEEE Systems Journal (2013) vol. 7(4) pp. 763-776.
3. N. Kanwar, N. Gupta, K. R. Niazi, A. Swarnkar, and R. C. Bansal, Multi-objective optimal DG allocation in distribution networks using bat algorithm, 3rd Southern African Solar Energy Conference (SASEC2015), Kruger National Park, South Africa, 11th-13th May 2015.
4. N. Kanwar, N. Gupta, K.R. Niazi, and A. Swarnkar, Optimal distributed generation allocation in radial distribution systems considering customer-wise dedicated feeders and load patterns, Journal of Modern Power and Clean Energy (2015) vol. 3(4), pp. 475-484.
5. Z. Cui, Y. Cao, X. Cai, J. Cai, and J. Chen, Optimal LEACH protocol with modified bat algorithm for big data sensing systems in Internet of Things, Journal of Parallel and Distributed Computing (2019) vol. 132, pp. 217-229.
6. I. Fister Jr., S. Fong, J. Brest, and I. Fister, Towards the self-adaptation in the bat algorithm, in: Proceedings of the 13th IASTED International Conference on Artificial Intelligence and Applications, Innsbruck, Austria, pp. 400-406, 2014.

7. S. Das, S.S. Mullick, and P.N. Suganthan, Recent advances in differential evolution- An updated survey, Swarm and Evolutionary Computation. (2016) vol. 27 pp. 1-30.
8. W. C. Hong, M. W. Li, J. Geng, and Y. Zhang, Novel chaotic bat algorithm for forecasting complex motion of floating platforms, Applied Mathematical Modelling (2019) vol. 72 pp. 425-443.
9. S. Lyu, Z. Li, Y. Huang, J. Wang, and J. Hu, Improved self-adaptive bat algorithm with step-control and mutation mechanisms, Journal of Computational Science (2019) vol. 30, pp. 65-78.
10. A. Al-Muraeb and H. A. Aty-Zohdy, Optimal design of short fiber bragg grating using bat algorithm with adaptive position update, IEEE Photonics Journal (2016), vol. 8(1).
11. C. K. Ng, C. H. Wu, W. H. Ip, and K. L. Yung, A smart bat algorithm for wireless sensor network deployment in 3-D environment, IEEE Communications Letters (2018) vol. 22(10), pp. 2120-2123.
12. D. Igrec, A. Chowdhury, B. Stumberger, and A. Sarjas, Robust tracking system design for a synchronous reluctance motor-SynRM based on a new modified bat optimization algorithm, Applied Soft Computing (2018) vol. 69, pp. 568-584.
13. M.E. Baran and F.F. Wu, Network reconfiguration in distribution system for loss reduction and load balancing. IEEE Transaction on Power Delivery (1989) vol. 4(2), pp. 1401-1407.