

Chapter 3

Bat Inspired Algorithms

Abstract In this chapter, we present two algorithms that are inspired by the behaviours of bats, i.e., bat algorithm (BaA) and bat intelligence (BI) algorithm. We first describe the general knowledge of the foraging behaviour of bats in [Sect. 3.1](#). Then, the fundamentals and performances of the BaA and BI algorithm are introduced in [Sects. 3.2](#) and [3.3](#), respectively. Finally, [Sect. 3.4](#) summarises in this chapter.

3.1 Introduction

Bats exhibit mysterious behaviours that have long since attracted the attention of human beings. Probably one of the most noticeable behaviours visible to us is they are capable to orientate their environments and food acquisition without relying on their eyesight (Merritt 2010). This is accomplished by bats continuously emit echolocation signals (Yovel et al. 2008). Through analyzing the returning echoes in the auditory system, bats can easily identify their surroundings and locate preys. By continuously observing and studying the abilities of bats, recently, computer scientists proposed several bat inspired algorithms (i.e., bat algorithm (BaA) and bat intelligence (BI) algorithm) to solve well-known optimization problems such as image processing, scheduling, clustering, and data mining.

3.1.1 Foraging Behaviour of Bats

Traditionally, families of bats are divided into two major groups, i.e., megabats and microbats. Typically, bats have three habitats as they feed through the whole year, i.e., roost, foraging habitats, and commuting habitats. The foraging habitats are used to find food, while the commuting habitats are used to travel between roosts and foraging habitats (Mills et al. 2010). In general, the foraging behaviour of bats can be divided into three phase: search phase, pursuit phase, and capture phase

(Kalko 1995). Also, they are very flexible in their foraging behaviour, such as feeding by aerial hawking (catching insects on the wing), gleaning (picking insets from vegetation), and pouncing on prey close to the ground. At each phase, they show different behaviours based on a perceptual system, i.e., bats echolocation.

3.1.2 Characteristics of Echolocation

One of the unique characteristics of bats from other animals is their echolocation capacity, i.e., emitting ultrasonic pulses and listening the echoes (Mills et al. 2010). According to the different pulses, they can detect prey, avoid obstacles, and even locate their roosting crevices in the dark (Yang 2010). For example, as bats fly through the night, their echolocation calls bounce off the environment, helping the bats find their way to and from their habitats. Typically, each bat emits about 10–20 pulse every second. Though each pulse lasts a few thousands of a second, it has a constant frequencies which normally range from 25 to 150 kHz (Yang 2010). Amazingly, when bats hunt for prey, the rate of pulse emission can reach roughly 200 pulses per second when they fly near their prey (Yang 2010). In contrast to the microbats use vocalization to emit sonar signals, the megabats use their tongue to emit very brief wideband echolocation (Yovel et al. 2011). Furthermore, although echolocation was first identified in bats, it is also used by some other species, such as dolphin and toothed whales.

3.2 Bat Algorithm

3.2.1 Fundamentals of Bat Algorithm

Bat algorithm (BaA) was originally proposed by Yang (2010) which is based on the echolocation behaviour of bats. To implement the proposed algorithm, a set of approximations or idealized rules are given as follows (Yang 2010):

- Based on the echolocation behaviour, the bats can detect the distance and the differentiate between food and background barriers as well, even in the darkness.
- Bats usually fly randomly to search for prey. According to that, some numerical parameters associated with the bats' foraging behaviour are defined first, such as velocity (v_i) at position (x_i) with a fixed frequency (f_{\min}), varying wavelength (λ) and loudness (A_o).
- The loudness can varies from a large positive (A_o) when searching for prey to a minimum constant value (A_{\min}) when homing towards the prey.

The basic steps of implementing BaA are described as follows.

3.2.1.1 Movement of Virtual Bats

Just like the real bats, the virtual bats fly randomly by adjusting its frequency first and updating its velocity and position next. The new locations/solutions (x_i^t) and velocities (v_i^t) at time step t are given via Eqs. (3.1)–(3.3) respectively (Yang 2010):

$$f_i = f_{\min} + (f_{\max} - f_{\min}) \cdot \beta, \quad (3.1)$$

$$v_i^t = v_i^{t-1} + (x_i^t - x_*) \cdot f_i, \quad (3.2)$$

$$x_i^t = x_i^{t-1} + v_i^t, \quad (3.3)$$

where $\beta \in [0, 1]$ is a random vector drawn from a uniform distribution, f_i is the velocity increment, f_{\min} and f_{\max} are the lower and upper bounds imposed for the frequency range of virtual bats, respectively, and x_* is the current global best location.

For the local search part, the random walks are used to generate a new solution for each bat from the current population via Eq. 3.4 (Yang 2010):

$$x_{new} = x_{old} + \varepsilon A^t, \quad (3.4)$$

where $\varepsilon \in [-1, 1]$ is a random number, while $A^t = \langle A_i^t \rangle$ is the average loudness of all the bats at this time step.

3.2.1.2 Loudness and Pulse Emission

The general rule is that the loudness decreases while the rate of pulse emission increases, when a bat has found its prey. The loudness and the rate of pulse emission are updated via Eqs. (3.5) and (3.6), respectively (Yang 2010):

$$A_i^{t+1} = \alpha A_i^t, \quad (3.5)$$

$$r_i^{t+1} = r_i^0 [1 - \exp(-\gamma t)], \quad (3.6)$$

where A_i^t and A_i^{t+1} are the previous and updated values of the loudness for the i th bat, α and γ are constants, and r_i^{t+1} is the pulse rate of the i th bat at iteration $t + 1$. For any $0 < \alpha < 1$ and $\gamma > 0$, we have Eq. (3.7) (Yang 2010):

$$A_i^t \rightarrow 0, r_i^t \rightarrow r_i^0, \text{ as } t \rightarrow \infty. \quad (3.7)$$

Summarizing the steps in BaA yields to (Yang 2010; Carbas and Hasancebi 2013):

- Step 1: Initializing bat population [such as position (x_i) and velocities (v_i)].
- Step 2: Initializing parameters, such as pulse frequency (f_i), pulse rates (r_i), and loudness parameters (A_i).

- Step 3: Evaluating bats in the initial population according to the fitness function.
- Step 4: Generating candidate bats through random flying and local search.
- Step 5: Evaluating candidate bats according to the fitness function.
- Step 6: Echolocation parameters update.
- Step 7: Rank the bats and find the current best x_* .
- Step 8: Termination. The steps 4–7 are employed in the same way until a termination criterion is met.

3.2.2 Performance of BaA

To test the effectiveness of BaA, a set of well-known test functions were adopted in (Yang 2010), namely, Rosenbrock's function, De Jong's standard sphere function, Ackley's function, Michalewica's test function, Schewfer's function, Rastrigin's function, Easom's function, Griewangk's function, and Shubert's test function. Compared with other algorithms [i.e., particle swarm optimization (PSO) and genetic algorithm (GA)], computational results showed that BaA performs significantly better than other algorithms in terms of accuracy and efficiency.

3.3 Emerging Bat Inspired Algorithms

In addition to the aforementioned BaA, the characteristics of this interesting animal also motivate researchers to develop other bat inspired innovative computational intelligence (CI) algorithms.

3.3.1 Bat Intelligence Algorithm

3.3.1.1 Fundamentals of Bat Intelligence (BI) Algorithm

Bat intelligence (BI) algorithm was originally proposed in (Malakooti et al. 2013b) that mimics the prey hunting behaviours of bats. For BI, a set of signals are generated not only based on the echolocation characteristic, but also the constant absolute target direction (CATD) technique, where the bat maintains same pursuit angle to the prey, has been incorporated. Also, during the prey hunting process, after seizing and consuming a prey, bats can proceed to capture other preys in the foraging area. The main steps of BI are described as follows (Malakooti et al. 2013b):

- Step 1: Initializing the parameters and population.
- Step 2: The fitness function value (i.e., signal strength) corresponding to each candidate solution is calculated.

- Step 3: Generating a set of solutions containing common elements.
- Step 4: Select the best solution.
- Step 5: Choosing a common element and include it in the common element list.
- Step 6: Checking the terminating condition. If it is reached, go to step 3 for new beginning. If a specified termination criteria is satisfied, stop and return the best solution.

3.3.1.2 Performance of BI

To illustrate the performance of BI, a multiple objective energy aware multiprocessor scheduling problem with three objectives (i.e., makespan, tardiness, and energy consumption) are presented in (Malakooti et al. 2013a). Compared with GA, the BI algorithm shows better performance to move towards the optimal solution.

3.4 Conclusions

In this chapter, we introduced two bat inspired CI algorithms. Although they are newly introduced CI method, we have witnessed the following rapid spreading of at least one of them, i.e., BaA:

First, several enhanced versions of BaA can be found in literature as outlined below:

- Differential operator and Lévy flight BaA (Xie et al. 2013).
- Modified BaA (Damodaram and Valarmathi 2012).

Second, the BaA has also been successfully applied to a variety of optimization problems as listed below:

- Data mining (Damodaram and Valarmathi 2012).
- Gas turbine generator exergy analysis (Lemma and Hashim 2011).
- Motor control (Bora et al. 2012).
- Scheduling optimization (Musikapun and Pongcharoen 2012).
- Steel space frames design optimization (Carbas and Hasancebi 2013).

Interested readers are referred to them, together with several excellent reviews [e.g., (Yang 2013; Schnitzer 2002; Sureja 2012)], as a starting point for a further exploration and exploitation of bat inspired algorithms.

References

- Bora, T. C., Coelho, L. D. S., & Lebensztajn, L. (2012). Bat-inspired optimization approach for the brushless DC wheel motor problem. *IEEE Transactions on Magnetics*, *48*, 947–950.
- Carbas, S., & Hasancebi, O. (2013, May 19–24). Optimum design of steel space frames via bat inspired algorithm, *10th World Congress on Structural and Multidisciplinary Optimization* (pp. 1–10). Orlando, Florida, USA.
- Damodaram, R., & Valarmathi, M. L. (2012). Phishing website detection and optimization using modified bat algorithm. *International Journal of Engineering Research and Applications*, *2*, 870–876.
- Kalko, E. K. V. (1995). Insect pursuit, prey capture and echolocation in pipistrelle bats. *Animal Behaviour*, *50*, 861–880.
- Lemma, T.A. & Hashim, F.B.M. (2011). Use of fuzzy systems and bat algorithm for exergy modeling in a gas turbine generator. IEEE Colloquium on Humanities, Science and Engineering Research (CHUSER) (pp. 305–310). IEEE, Penang, 5–6 December 2013.
- Malakooti, B., Kim, H. & Sheikh, S. (2013a). Bat intelligence search with application to multi-objective multiprocessor scheduling optimization. *International Journal of Advanced Manufacturing Technology*, (Vol. 60, pp. 1071–1086). DOI [10.1007/s00170-011-3649-z](https://doi.org/10.1007/s00170-011-3649-z).
- Malakooti, B., Sheikh, S., Al-Najjar, C. & Kim, H. (2013b). Multi-objective energy aware multiprocessor scheduling using bat intelligence. *Journal of Intelligent Manufacturing*, *24*, 805–819. DOI [10.1007/s10845-012-0629-6](https://doi.org/10.1007/s10845-012-0629-6).
- Merritt, J. F. (2010). *The biology of small mammals*. Baltimore: John Hopkins University Press.
- Mills, D. S., Marchant-Forde, J. N., McGreevy, P. D., Morton, D. B., Nicol, C. J., Phillips, C. J. C., et al. (Eds.). (2010). *The encyclopedia of applied animal behaviour and welfare*. Wallingford: CAB International. ISBN 978-0-85199-724-7.
- Musikapun, P., & Pongcharoen, P. (2012). Solving multi-stage multi-machine multi-product scheduling problem using bat algorithm. *2nd International Conference on Management and Artificial Intelligence* (pp. 98–102). IACSIT.
- Schnitzer, M. J. (2002). Amazing algorithms. *Nature*, *416*, 683.
- Sureja, N. (2012). New inspirations in nature: a survey. *International Journal of Computer Applications and Information Technology*, *1*, 21–24.
- Xie, J., Zhou, Y., & Chen, H. (2013). A novel bat algorithm based on differential operator and Lévy flights trajectory. *Computational Intelligence and Neuroscience*, 1–13.
- Yang, X.-S. (2010). A new metaheuristic bat-inspired algorithm. *Nature inspired cooperative strategies for optimization (NISCO 2010)*, Studies in Computational Intelligence, SCI 284 (pp. 65–74). Berlin: Springer.
- Yang, X.-S. (2013). Bat algorithm: literature review and applications. *International Journal of Bio-Inspired Computation*, *5*, 141–149.
- Yovel, Y., Franz, M. O., Stilz, P., & Schnitzler, H.-U. (2008). Plant classification from bat-like echolocation signals. *PLoS Computational Biology*, *4*, 1–13.
- Yovel, Y., Geva-Sagiv, M., & Ulanovsky, N. (2011). Click-based echolocation in bats: not so primitive after all. *Journal of Comparative Physiology A*, *197*, 515–530.