Towards the Development of a Parameter-free Bat Algorithm

Iztok Fister Jr., Iztok Fister
University of Maribor
Faculty of Electrical Engineering and Computer Science
Smetanova 17, 2000 Maribor
iztok.fister1@um.si

Xin-She Yang
Middlesex University
School of Science and Technology
The Burroughs
London, United Kingdom

ABSTRACT

The bat algorithm is a very simple and efficient nature-inspired algorithm belonging to the swarm intelligence family. Recent studies showed its potential in solving the continuous and discrete optimization problems and many researchers have applied bat algorithm to solve real-world problems. However, empirical studies showed that the main issue of most algorithms is the setting of control parameters. An algorithm such as the bat algorithm typically has a few parameters, and it is very time-consuming to find its best parameter combination. Here, we propose a new parameter-free bat algorithm variant without the need for control parameters. Initial experiments on basic benchmark functions showed the potential of this new approach.

Keywords

bat algorithm, control parameters, optimization

1. INTRODUCTION

As the world is becoming a global village, advances in technologies means that new challenges are emerging. Better and greener products are needed for companies to have a competitive edge. Thus, the optimization of product designs and manufacturing processes become ever-more important. It is expected that artificial intelligence is among the most promising developments for the next 20 years. Many artificial intelligence tasks can be achieved by optimization and machine learning techniques.

One of the most important tasks for many productions is how to improve the production of products and services for the market, which requires the optimal design and the optimal use of resources. Until recently, many methods have been used to help designers and developers to solve such optimization problems. Some methods are pure mathematical, while others are combined with computer sciences. In line with this, optimization algorithms play a big role. Recently, stochastic nature-inspired methods are among the promising kind of optimization algorithms. This family consists of the following algorithms: genetic algorithms [7], genetic programming [11], evolution strategies [1], evolutionary programming [5], differential evolution [14], particle swarm optimization [9], firefly algorithm [18], bat algorithm [19] and dozens of others [4].

The bat algorithm (BA)is one of the latest algorithms in the swarm intelligence (SI) domain. Due to its simplicity, it is a very efficient as well. The original BA works with five parameters that represent a potential problem for users which usually may not know how to specify their values properly. Therefore, this paper introduces a new parameter-free or parameterless BA (PLBA) that eliminates this drawback and then proposes techniques for a rational and automated parameter setting on behalf of the user. This study is based on the paper of Lobo and Goldberg [12].

The structure of the remainder of the paper is as follows. Section 2 presents a description of the original BA. In Section 3, a design of the parameter-free or parameterless BA (also PLBA) is presented. Experiments and results are discussed in Section 4. The paper concludes in Section 5 where the directions for the future work are also outlined.

2. BAT ALGORITHM

The origin of the BA development can date back to the year of 2010 when Xin-She Yang in [19] proposed the new nature-inspired algorithm that mimics the phenomenon of echolocation by some species of bats for the optimization process. This algorithm can integrate the characteristics of many algorithms such as particle swarm optimization (PSO) algorithm [9] and simulated annealing (SA) [10]. Primarily, BA was developed for continuous optimization, but many recent papers showed the potential of the algorithm in solving discrete optimization problems. In a nutshell, this algorithm is widespread in many areas of optimization and industrial applications [8,13,15–17]. Yang proposed the following rules which mimics the bat behavior:

- All bats use echolocation to sense the distance to target objects.
- Bats fly with the velocity v_i at position x_i, the frequency $Q_i \in [Q_{min}, Q_{max}]$ (also the wavelength λ_i), the rate of pulse emission $r_i \in [0, 1]$, and the loudness

$A_i \in [A_0, A_{min}]$. The frequency (and wavelength) can be adjusted depending on the proximities of their targets.

- The loudness varies from a large (positive) A_0 to a minimum constant value A_{min}.

These three rules mimics the natural behavior of bats, while the full basic pseudo-code is presented in Algorithm 1.

Algorithm 1 Bat algorithm

Input: Bat population $\mathbf{x_i} = (x_{i1}, \ldots, x_{iD})^T$ for $i = 1 \ldots Np$, MAX_FE.
Output: The best solution $\mathbf{x}_{best}$ and its corresponding value $f_{min} = \min(f(\mathbf{x}))$.

```
1:  init_bat();
2:  eval = evaluate_the_new_population;
3:  f_min = find_the_best_solution(x_best); {initialization}
4:  while termination_condition_not_met do
5:    for i = 1 to Np do
6:      y = generate_new_solution(x_i);
7:      if rand(0,1) < r_i then
8:        y = improve_the_best_solution(x_best)
9:      end if { local search step }
10:     f_new = evaluate_the_new_solution(y);
11:     eval = eval + 1;
12:     if f_new ≤ f_i and N(0,1) < A_i then
13:       x_i = y; f_i = f_new;
14:     end if { save the best solution conditionally }
15:     f_min=find_the_best_solution(x_best);
16:   end for
17: end while
```

The BA is a population-based algorithm, where the population size is controlled by a parameter Np. The main loop of the algorithm (lines 4-16 in Algorithm 1) starts after the initialization (line 1), the evaluation of the generated solutions (line 2) and determination of the best solutions (line 4). It consists of the following elements:

- generating a new solution (line 6),
- improving the best solution (lines 7-9),
- evaluating the new solution (line 10),
- saving the best solution conditionally (lines 12-14),
- determining the best solution (line 15).

The generation of a new solution obeys the following equation:

$$\begin{aligned} Q_i^{(t)} &= Q_{min} + (Q_{max} - Q_{min})N(0,1), \\ \mathbf{v}_i^{(t+1)} &= \mathbf{v}_i^t + (\mathbf{x}_i^t - \mathbf{x}_{best})Q_i^{(t)}, \\ \mathbf{x}_i^{(t+1)} &= \mathbf{x}_i^{(t)} + \mathbf{v}_i^{(t+1)}, \end{aligned} \tag{1}$$

where $N(0,1)$ is a random number drawn from a Gaussian distribution with a zero mean and a standard deviation of one, and $Q_i^{(t)} \in [Q_{min}^{(t)}, Q_{max}^{(t)}]$ is the frequency determining the magnitude of the velocity change. The improvement of the current best solution is performed according to the following equation:

$$\mathbf{x^{(t)}} = \mathbf{x}_{best} + \epsilon A_i^{(t)} N(0,1), \tag{2}$$

where $N(0,1)$ denotes the random number drawn from a Gaussian distribution with a zero mean and a standard deviation of one. In addition, ϵ is the scaling factor and $A_i^{(t)}$ is the loudness. The improvement is controlled by a parameter r_i. It is worth pointing that the parameter balances the exploration and exploitation components of the BA search process, where exploitation is governed by Eq. 1 and exploitation by Eq. 2.

The evaluation function models the characteristics of the problem to be solved. The archiving of the best solution conditionally is similar to that used in SA, where the best solution is taken into the new generation according to a probability controlled by the parameter A_i in order to avoid getting stuck into a local optimum.

2.1 Control parameters in the bat algorithm and their bottlenecks

Control parameters guide algorithms during the search space and may have a huge influence on the quality of obtained solutions. In summary, the BA is controlled by the following control parameters:

- the population size NP,
- loudness A_i,
- pulse rate r_i,
- minimum frequency Q_{min} and
- maximum frequency Q_{max}.

As we can see, the BA has five control parameters that can be difficult for users to set a proper combination of control parameter settings that are the most suitable for the particular problem. Though some algorithms have fewer parameters such as differential evolution [14] that employs only three basic control parameters. However, the parameter tuning can be a time-consuming task for all algorithms. On the other hand, there are also some other robust self-adaptive [3] BA variants that can modify the values of control parameters during the run.

The main task for us is to develop a parameterless variant of the bat algorithm.

3. DESIGN OF A PARAMETERLESS BAT ALGORITHM

In order to develop a new parameterless BA (PLBA), the influence of control parameters was studied and extensive studies revealed that some algorithm parameters can be effectively set. For example, the parameter $Q_i \in [Q_{min}, Q_{max}]$ determines the magnitude of the change and settings of parameters Q_{min} and Q_{max} depend on the problem of interest. However, the rational setting of this parameter can be approximated with the lower x_i^{Lb} and upper x_i^{Ub} bounds of the particular decision variables as follows:

$$Q_i^{(t)} = \frac{x_i^{(Ub)} - x_i^{(Lb)}}{Np} \cdot N(0,1), \tag{3}$$

where $N(0,1)$ has the same meaning as in Eq. (1). For example, when the $x_i^{(Lb)} = -100.0$ and $x_i^{(Ub)} = 100.0$, the frequency is obtained in the interval $Q_i^{(t)} \in [0,2]$.

The rationality for setting the values of parameters r_i and A_i are obtained, based on the following consideration. Parameter r_i controls the exploration/exploitation components of the BA search process. The higher the value, the more the process is focused on the exploitation. However, the higher r_i also means that the modified copies of the best solution are multiplied in the population. As a result, the premature convergence to the local optimum can occur. The appropriate value of this parameter is $r_i = 0.1$, as revealed during the extensive experimental work.

Similar consideration is valid also for parameter A_i; i.e., the lower the parameter, the less time the best current solution is preserved in the new generation. Therefore, the rational selection of this parameter is $A_i = 0.9$ that preserves the 90% of each best solutions and ignores the remaining 10%.

The population size is also a crucial parameter in the BA. A lower population size may suffer from the lack of diversity, while a higher population size may cause slow convergence. However, the rational setting of the population size depends on the problem of interest. Therefore, this parameter is varied in the interval $Np \in [10, 1280]$ in the proposed PLBA such that each population size multiplied by two in each run starting with $Np = 10$. In this way, eight instances of the PLBA is executed and a user selects the best results among the instances.

4. EXPERIMENTS

The purpose of our experimental work was to show that the results of the proposed PLBA are comparable if not better than the results of the original BA. In line with this, the original BA was compared with the PLBA using the rational selected parameter setting, i.e., Q_i was calculated according Eq. (3) (e.q., $Q_i \in [0.0, 2.0]$), pulse rate and loudness were fixed as $r_i = 0.1$ and $A_i = 0.9$, while the population size was varied in the interval $Np = \{10, 20, 40, 80, 160, 320, 640, 1280\}$. In contrast, the original BA was run using the following parameters: $Np = 100$, $r_i = 0.5$, $A_i = 0.5$ and $Q_i \in [0.0, 2.0]$ as proposed in Fister et al. [3].

Experiments were run on the benchmark suite consisting of five functions (Fig. 1). Here, the domains of parameters were limited into the interval $[-10, 10]$ by functions, while the dimensions of functions $D = 10$ were applied during this preliminary work. The optimization process of functions was stopped after 10,000 fitness/function evaluations. Each algorithm was run 25 times.

The results of the comparative study are summarized in Table 2, where the mean values and standard deviations are given for each algorithm. The BA in Table 2 denotes the original BA while PL-i the parameter-less BA using different population sizes Np.

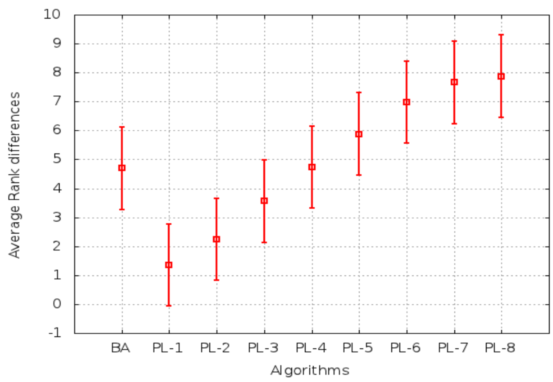

Figure 1: Summary of the comparison analysis.

The results are also statistically evaluated using the Friedman's non-parametric statistical tests [2, 6]. These results are shown in Table 3 where the ranks, confidence intervals with significance level $\alpha = 0.05$, and dagger signs denoting the significance difference are summarized.

Table 3: Friedman's statistical tests.

Alg.	Np	Rank	CI	Sign.
BA	100	4.7	[3.28,6.12]	†
PL-1	10	1.36	[-0.06,2.78]	†
PL-2	20	2.24	[0.82,3.66]	†
PL-3	40	3.56	[2.14,4.98]	†
PL-4	80	4.74	[3.32,6.16]	†
PL-5	160	5.88	[4.46,7.30]	†
PL-6	320	6.98	[5.56,8.40]	
PL-7	640	7.66	[6.24,9.08]	
PL-8	1280	7.88	[6.46,9.30]	‡

The same results are also presented graphically in Fig. 1, where ranges and confidence intervals are shown as points and lines, respectively. Two algorithms are significantly different, if their confidence intervals do not overlap. From Fig. 1, it can be seen that the population size has a significant influence on the results of the PLBA. The higher the population size, the better the results. Interestingly, using the PLBA with population sizes $Np = 640$ and $Np = 1,280$ significantly outperformed the results of the original BA.

5. CONCLUSION

Population-based nature-inspired algorithms can be a powerful tool for solving the hard optimization problems. However, these kinds of algorithms can depend crucially on the good parameter setting. Unfortunately, finding the proper parameter setting for a specific problem of interest is not easy and can pose challenges for users. In this paper, a parameterless BA (also PLBA) algorithm has been proposed in order to avoid these issues.

The PLBA sets parameters either by guessing their rational values, such as setting the parameters pulse rate r_i, loudness A_i, minimum Q_{min} and maximum Q_{max} frequencies, or by searching for their proper values experimentally, such as setting the population size Np. The results of the proposed PLBA are comparable with the results of the original BA by solving the benchmark suite of five well-known functions. A statistical analysis of the results has showed that

Table 1: Summary of the benchmark functions.

Tag	Function	Definition	Domain
f_1	Ackley's	$f(\mathbf{x}) = -20\exp(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^n x_i^2}) - \exp(\frac{1}{n}\sum_{i=1}^n \cos(2\pi x_i)) + 20 + e$	$[-10, 10]$
f_2	Griewank	$f(\mathbf{x}) = 1 + \frac{1}{4000}\sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos(\frac{x_i}{\sqrt{i}})$	$[-10, 10]$
f_3	Rastrigin	$f(\mathbf{x}) = 10n + \sum_{i=1}^n (x_i^2 - 10\cos(2\pi x_i))$	$[-10, 10]$
f_4	Sphere	$f(\mathbf{x}) = \sum_{i=1}^n x_i^2$	$[-10, 10]$
f_5	Whitley	$f(\mathbf{x}) = \sum_{i=1}^n \sum_{j=1}^n \left[\frac{(100(x_i^2-x_j)^2+(1-x_j)^2)^2}{4000} - \cos(100(x_i^2 - x_j)^2 + (1 - x_j)^2) + 1\right]$	$[-10, 10]$

Table 2: Experimental results.

Alg.	Np	f_1	f_2	f_3	f_4	f_5
BA	40	1.11e+01±8.66e-01	2.76e-01±2.70e-01	1.70e+02±3.92e+01	7.79e+01±2.53e+01	3.21e+04±1.95e+04
PL-1	10	1.15e+01±8.91e-01	3.96e-01±2.68e-01	2.18e+02±4.61e+01	9.65e+01±3.33e+01	4.87e+04±2.47e+04
PL-2	20	1.13e+01±1.15e+00	3.63e-01±2.94e-01	1.72e+02±5.09e+01	7.80e+01±2.82e+01	4.65e+04±2.65e+04
PL-3	40	1.06e+01±7.11e-01	2.52e-01±2.59e-01	1.46e+02±4.47e+01	6.56e+01±2.00e+01	3.03e+04±1.27e+04
PL-4	80	1.01e+01±7.59e-01	1.79e-01±1.64e-01	1.27e+02±3.58e+01	5.49e+01±1.88e+01	1.90e+04±1.26e+04
PL-5	160	9.68e+00±8.44e-01	1.07e-01±3.24e-02	1.24e+02±3.59e+01	4.58e+01±1.58e+01	1.63e+04±7.32e+03
PL-6	320	9.42e+00±7.32e-01	1.16e-01±3.82e-02	1.01e+02±2.25e+01	3.67e+01±1.17e+01	1.16e+04±3.95e+03
PL-7	640	8.76e+00±9.29e-01	1.02e-01±2.45e-02	9.28e+01±2.45e+01	3.16e+01±1.38e+01	9.08e+03±3.49e+03
PL-8	1280	8.57e+00±8.62e-01	1.02e-01±2.08e-02	9.22e+01±2.63e+01	3.23e+01±1.07e+01	1.08e+04±4.40e+03

the proposed PLBA can achieve comparable results as those obtained by the original BA. Such findings encourage us to continue with this work in the future. As the first goal, we would like to develop also a variant of a parameterless cuckoo search algorithm. We will also investigate the effect of the population size further in various applications.

6. REFERENCES

[1] Thomas Bäck. *Evolutionary Algorithms in Theory and Practice: Evolution Strategies, Evolutionary Programming, Genetic Algorithms*. Oxford University Press, Oxford, UK:, 1996.

[2] Janez Demšar. Statistical comparisons of classifiers over multiple data sets. *J. Mach. Learn. Res.*, 7:1–30, December 2006.

[3] Iztok Fister Jr., Simon Fong, Janez Brest, and Iztok Fister. A novel hybrid self-adaptive bat algorithm. *The Scientific World Journal*, 2014, 2014.

[4] Iztok Fister Jr, Xin-She Yang, Iztok Fister, Janez Brest, and Dušan Fister. A brief review of nature-inspired algorithms for optimization. *Elektrotehniški vestnik*, 80(3):116–122, 2013.

[5] Lawrence Jerome Fogel, Alvin J. Owens, and Michael John Walsh. *Artificial Intelligence through Simulated Evolution*. John Willey & Sons, Inc., New York, USA:, 1966.

[6] Milton Friedman. A comparison of alternative tests of significance for the problem of m rankings. *Ann. Math. Statist.*, 11(1):86–92, 03 1940.

[7] David E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley Longman Publishing Co., Inc., Boston, USA:, 1989.

[8] Andrés Iglesias, Akemi Gálvez, and Marta Collantes. Global-support rational curve method for data approximation with bat algorithm. In *Artificial Intelligence Applications and Innovations: 11th IFIP WG 12.5 International Conference, AIAI 2015, Bayonne, France, September 14-17, 2015, Proceedings*, volume 458, page 191. Springer, 2015.

[9] James Kennedy and Russell Eberhart. Particle swarm optimization. In *Neural Networks, 1995. Proceedings., IEEE International Conference on*, volume 4, pages 1942–1948. IEEE, 1995.

[10] S Kirkpatrick, CD Gelatt Jr, and MP Vecchi. Optimization by simulated annealing. *Science*, 220(4598), 1983.

[11] John Koza. *Genetic Programming 2 - Automatic Discovery of Reusable Programs*. MIT Press, Cambridge, USA:, 1994.

[12] Fernando G. Lobo and David E. Goldberg. An overview of the parameter-less genetic algorithm. In *Proceedings of the 7th Joint Conference on Information Sciences. (Invited paper)*, pages 20–23, 2003.

[13] Navaneethakrishna Makaram and Ramakrishnan Swaminathan. A binary bat approach for identification of fatigue condition from semg signals. In *Swarm, Evolutionary, and Memetic Computing*, pages 480–489. Springer, 2014.

[14] Rainer Storn and Kenneth Price. Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. *Journal of global optimization*, 11(4):341–359, 1997.

[15] Anass Taha, Mohamed Hachimi, and Ali Moudden. Adapted bat algorithm for capacitated vehicle routing problem. *International Review on Computers and Software (IRECOS)*, 10(6):610–619, 2015.

[16] TP Talafuse and EA Pohl. A bat algorithm for the redundancy allocation problem. *Engineering Optimization*, (ahead-of-print):1–11, 2015.

[17] Jinfeng Wang, Xiaoliang Fan, Ailin Zhao, and Mingqiang Yang. A hybrid bat algorithm for process planning problem. *IFAC-PapersOnLine*, 48(3):1708–1713, 2015.

[18] Xin-She Yang. Firefly algorithm. In *Nature-Inspired Metaheuristic Algorithms*, pages 79–90. Luniver Press, London, UK:, 2008.

[19] Xin-She Yang. A new metaheuristic bat-inspired algorithm. In *Nature inspired cooperative strategies for optimization (NICSO 2010)*, pages 65–74. Springer, 2010.