An Improved Bat Algorithm with Doppler Effect for Stochastic Optimization

Guanghui LIU, Heyan HUANG, Shumei WANG, Zhaoxiong CHEN

Abstract

Bat Algorithm is a powerful nature-inspired method for solving many multi-objective optimization problems. This paper presents a novel algorithm which is called Doppler Effect Bat Algorithm (DEBA). This algorithm intends to combine Doppler effect with the bat algorithm. Based on bats’ Doppler effect theory and the framework of the original bat algorithm, the new frequency equation in this algorithm is proposed, and the velocities and locations equation are also updated. The improved algorithm is also compared with PSO and the basic bat algorithm in the paper. In order to analyze the improvement on the accuracy of finding the near best solution and the reduction in the computational cost, five well-known and commonly used test functions are used in the experiments. Simulations show that the proposed algorithm seems much superior to PSO and the original bat algorithm.

Key words: Bat Algorithm, Doppler Effect, Global Optimization, Frequency, Loudness, Velocity, PSO

1. Introduction

For solving optimization problems, heuristic methods, such as genetic algorithms, have been successfully used. Modern metaheuristic algorithms are becoming powerful in solving global optimization problems\(^1,2\), especially for the NP-hard problems such as the travelling salesman problem. Particle swarm optimization (PSO) was developed by Kennedy and Eberhart in 1995 \(^3\), based on the swarm behaviour such as fish and bird schooling in nature. It has now been applied to find solutions for many optimization applications\(^4,5\).

By idealizing some of the echolocation characteristics of micro-bats, Yang proposed a new optimization algorithm, namely, Bat Algorithm (BA), in 2010. Although the original BA presents superior results in the experiments than PSO, we notice that the performance and the accuracy of the original BA still have the capacity to present better.

In this paper, based on the echolocation behaviour of bats, here we develop Modified Bat Algorithm with Doppler Effect (DEBA). According to the experimental results, our proposed DEBA presents more accurate results in finding near best solutions.
2. Bat Algorithm [BA]

The bat algorithm \([6,7]\) uses the echolocation behaviour of bats. These bats emit a very loud sound pulse (echolocation) and listens for the echo that bounces back from the surrounding objects. The ith bat flies randomly with velocity \(v_i\) at position \(X_i\) with a fixed frequency \(f_{\text{min}}\). The bat varies its wavelength \(\lambda\) and loudness \(A_0\) to search for food. In simulations, the movement of the virtual bat in Yang’s method is given by Eq. (1) – Eq.(3).

\[
f_i = f_{\text{min}} + (f_{\text{max}} - f_{\text{min}}) \cdot \beta \quad (1)
\]

\[
v'_i = w \cdot v'_i + (x'_i - x_{\text{best}}) \cdot f'_i \quad (2)
\]

\[
x'_i = x'_{i-1} + v'_i \quad (3)
\]

Where \(f\) is the frequency used by the bat seeking for its prey, the suffixes, \(\text{min}\) and \(\text{max}\), represent the minimum and maximum value, respectively. \(x_i\) denotes the location of the \(i^{th}\) bat in the solution space, \(v_i\) represents the velocity of the bat, \(t\) indicates the current iteration, \(\beta\) is a random vector, which is drawn from a uniform distribution, and \(\beta \in [0,1]\), and \(x_{\text{best}}\) indicates the global near best solution found so far over the whole population.

For local search procedure (exploitation) each bat takes a random walk creating a new solution for itself based on the best selected current solution.

\[
x_{\text{new}} = x_{\text{old}} + \epsilon A^t \quad (4)
\]

Where \(\epsilon \in [-1,1]\) is a random number, \(A^t\) is the average loudness of all bats at this time step. The loudness decreases as a bat tends closer to its food and pulse emissions rate increases.

\[
A^t_{i+1} = \alpha \cdot A_i^t \quad (5)
\]

\[
r^t_{i+1} = r^0_i \left[1 - e^{-\alpha t}\right] \quad (6)
\]

Where \(\alpha\) and \(\gamma\) are constants. In the simplicity case, \(\alpha = \gamma = 0.9\).
3. Bat’s Doppler Effect

Doppler Effect is the change in frequency of a wave for an observer moving relative to the source of the wave. The received frequency is higher (compared to the emitted frequency) during the approach, it is identical at the instant of passing by, and it is lower during the recession.

Bats perform echolocation by sending out cries and gathering information about their surroundings based on the echoes that return back to them. The incoming direction of the echoes helps determine where things are located. The time it takes for the echoes to return relates to how far various objects are away. By the intensity of the returning signal they can tell information about the size of the objects and distance as well[8,9,10].

When bats are flying they experience a Doppler-shift in the frequency of the cries they initially sent out compared to the frequency of echoes that returns to them. Bats change the frequency of their cries to keep the returning echoes centered within a narrow frequency range where they have very sensitive hearing. This is called Doppler-shift compensation and allows them to hear the beating wings of insects as slight deviations of frequency within their most sensitive hearing range[11].

In determining the frequency that the bat hears returning back to him two Doppler shifts take place. First the bat is a moving source of frequency \( f_0 \), and the speed of sound in air is \( c = 340 \text{ m/s} \). The velocity of flying is positive if the source is moving away from the observer, and negative if the source is moving towards the observer. Assumed that \( v_s \) is positive, so that the frequency that hits a stationary object is

\[
\begin{align*}
    f_1' &= \frac{c}{c - v_s} f_0 \quad (7)
\end{align*}
\]

The bat is then an observer moving toward the stationary object that acts like a source of \( f'' \).

\[
\begin{align*}
    f_2'' &= \frac{c + v_s}{c} f_1' \quad (8)
\end{align*}
\]

Finally, the bat observes the frequency that echoes to it:

\[
\begin{align*}
    f_0'' &= \frac{c + v_s}{c - v_s} f_0 \quad (9)
\end{align*}
\]

4. Doppler Effect Bat Algorithm

A modified Bat Algorithm with Doppler Effect (DEBA) is proposed. Based on bat’s doppler effect theory, The pulse frequency \( f_i \) can be reformed into Eq.(10)
Where $f$ denotes the frequency used by the bat seeking for its prey, in the initial, $f \in [0,1]$ is a random number. $c$ is the speed of sound in air, $c = 340 \text{ m/s}$; $v_i$ is the velocity of bat flying. $t$ indicates the current iteration.

The velocity update of the bat is $v_i$.

$$v_i^{t} = w_1 \cdot v_i^{t-1} + (x_{best} - x_i^{t}) \cdot f_i^{t}$$  \hspace{1cm} (11)$$

$$x_i^{t} = w_2 \cdot x_i^{t-1} + v_i^{t}$$  \hspace{1cm} (12)$$

Where $x_{best}$ is the global best of all the bats, $w$ is inertia weight. The reason for this added parameters $(w_1, w_2)$ in the velocity and locations equation is because the parameters $(w_1,w_2)$ can balance the proportional relationship between the global convergence ability and the local convergence ability, and improve the performance of the algorithm. In general, $(w_1,w_2) \in [0.7,1]$, when $V_{max}$ is not small ($\geq 3$), an inertia weight $w_1 = 0.8$ is a good choice.

The basic steps of the Doppler Effect Bat Algorithm [DEBA] can be summarized as the pseudo code shown in Figure1.

**Objective function** $F(x)$, $x=(x_1,...,x_d)^T$

**Initialize** dimensions $D$ and inertia weight $w$

Create a swarm with $n$ bats

Initialize the bat population $x_i (i=1,2...,n)$ and $v_i$

Define Pulse frequency $f_i$ at $x_i$

Initialize the rates $r_i$ and the loudness $A_i$

Initialize current-global-best and current-local-best for the swarm

**While** ($t < \text{Max number of iterations}$)

Generate new solutions by adjusting frequency and updating velocities and locations /solutions [Eq.(10), (11) and (12)],

If (rand $> r_i$)

Select a solution among the best solutions [Eq.(4)]

Generate a local solution around the selected best solution

End if

Generate a new solution by flying randomly

If (rand $< A_i \& F(x_i) < F(x_{best})$)

Accept the new solutions [Eq.(12)]

Increase $r_i$ and reduce $A_i$ [Eq.(5) and Eq.(6)]

End if

Calculate objective function value $F(x)$
Rank the bats and find the current best $x_{best}$

end while

Figure 1. Pseudo code of the Doppler Effect Bat Algorithm (DEBA).

5. Simulation and Comparison

5.1. Comparison of DEBA with PSO

Various studies show that PSO algorithms can outperform genetic algorithms (GA) \cite{3,12} and other conventional algorithms for solving many optimization problems. This is partially due to the fact that the broadcasting ability of the current best estimates gives better and quicker convergence towards the optimality\cite{13,14,15}. A general framework for evaluating statistical performance of evolutionary algorithms has been discussed in detail by Shilane et al \cite{16}. Various test functions for optimization algorithms have been developed over many years, and a relatively comprehensive review of these test functions can be found in \cite{17}. Despite the superiority of BA over PSO has been discussed in detail in Yang’s method\cite{6}, DEBA, as an improved BA algorithm which is better than PSO, must be validated.

Now we will compare the DEBA with PSO for a standard test functions, Ackley’s function,

$$F(x) = 20 + e - 20 \exp\left(-0.2 \sqrt{\frac{1}{d} \sum_{i=1}^{d} x_i^2}\right) - \exp\left(\frac{1}{d} \sum_{i=1}^{d} \cos(2\pi x_i)\right)$$

(13)

Where (d=1, 2, …)\cite{18}. The global minimum $F_*=0$ occurs at (0,0,…,0) in the domain of $-30 \leq x_i \leq 30$, where $i=1,2,…,d$. The landscape of the 2D Ackley’s function is shown in Figure 2, and this global minimum can be found after about 800 evaluations for 40 bats after 20 iterations as shown in Figure 3.

Figure 2. Ackley function for two independent variables with a global minimum $F_*=0$ at (0, 0).
We will use the same population size of $n=40$ for all algorithms in all simulations. The parameter settings for DEBA are listed in Table 3 and Table 4. The PSO used is the standard version without any inertia function. After implementing these algorithms using Matlab, we have carried out extensive simulations and each algorithm has been run at least 25 times so as to carry out meaningful statistical analysis. Dimension (d) of Ackley’s function is set to: $d=128$. PSO parameters are set: Inertia weight $w=0.8$; learning factor $c_1=c_2=2$.

The algorithms stop when the variations of function values are less than a given tolerance $\xi \leq 10^{-14}$. The results are summarized in the Table 1 where the global optima are reached.

![Figure 3. The initial locations of the 40 bats (left) and their locations after 20 iterations (right).](image1)

![Figure 4. The convergence graphs of the DEBA and PSO for Ackley’s function.](image2)

**Table 1.** The average outcome and average iterations in 25 runs.

<table>
<thead>
<tr>
<th></th>
<th>PSO</th>
<th>DEBA</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Average outcome</td>
<td>Average iterations</td>
</tr>
<tr>
<td>Ackley’s function</td>
<td>9.69</td>
<td>5000</td>
</tr>
<tr>
<td>$-30 \leq x_i \leq 30$</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
We can see that the DEBA is noticeably more efficient in finding the global optima with the success rates of 100%. This is no surprising as the aim of developing the improved algorithm was to try to use the advantages of existing algorithms and other interesting feature inspired by the fantastic behaviour of echolocation of microbats.

5.2. Comparison of DEBA with BA

In order to validate and analyze the accuracy and the computational speed of our proposed algorithm, we have implemented it in Matlab. In our simulations, four test functions, which are listed in Eq.(14) – Eq.(17), are used. The results are compared with the original BA. The optimization goal for all test functions is to minimize the outcome.

1. the eggcrate’s function:

\[
F_1(x, y) = x^2 + y^2 + 25 \cdot (\sin^2 x + \sin^2 y), \quad (x, y) \in [-2\pi, 2\pi] \times [-2\pi, 2\pi]
\] (14)

2. the Rosenbrock’s function:

\[
F_2(x) = \sum_{i=1}^{D-1} [100 \cdot (x_{i+1} - x_i^2)^2 + (x_i - 1)^2], \quad -50 \leq x_i \leq 50
\] (15)

3. the De Jong’s sphere function:

\[
F_3(x) = \sum_{i=1}^{D} x_i^2, \quad -10 \leq x_i \leq 10
\] (16)

4. the Griewank’s function:

\[
F_4(x) = \frac{1}{4000} \sum_{i=1}^{D} x_i^2 - \prod_{i=1}^{D} \cos \left( \frac{x_i}{\sqrt{i}} \right) + 1, \quad -600 \leq x_i \leq 600
\] (17)

In our implementation, the total population size is set to 40. Each test function contains the full iterations is repeated by 25 runs with different random seeds. The final result is obtained by taking the average of the outcomes from all runs.

The initial dimension of the solution space, theoretical global best point and global best value(min) for all tested functions are listed in Table2. The parameter settings for BA and our proposed method DEBA are listed in Table 3 and Table 4. The outcome and the computational time are listed in Table 5 and Table 6. The convergence graphs of four tested functions are in Figure 5.

The experiments are taken on the personal computer with an Intel Core-i7 2600K 3.4GHz CPU, 4GB RAM, Windows XP OS, with Matlab Version R2007b.
Table 2. The initial dimension and the theoretical optimal for all tested functions.

<table>
<thead>
<tr>
<th>Tested functions</th>
<th>Dimension(D)</th>
<th>Theoretical global best point</th>
<th>Theoretical global best value (min)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$F_1$</td>
<td>2</td>
<td>(0,0)</td>
<td>0</td>
</tr>
<tr>
<td>$F_2$</td>
<td>2</td>
<td>(1,1)</td>
<td>0</td>
</tr>
<tr>
<td>$F_3$</td>
<td>256</td>
<td>(0,0,….0)₀</td>
<td>0</td>
</tr>
<tr>
<td>$F_4$</td>
<td>30</td>
<td>(0,0,….0)₀</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 3. The parameter setting for BA.

<table>
<thead>
<tr>
<th>initial $A_i^0$</th>
<th>Initial $r_i^0$</th>
<th>$[f_{min}, f_{max}]$</th>
<th>iteration</th>
<th>$\varepsilon$</th>
<th>$\alpha$</th>
<th>$\gamma$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.99</td>
<td>[0,1]</td>
<td>5000</td>
<td>[-1,1]</td>
<td>0.9</td>
<td>0.9</td>
</tr>
</tbody>
</table>

Table 4. The parameter setting for DEBA.

<table>
<thead>
<tr>
<th>$v_{max}$</th>
<th>$w_1$</th>
<th>$w_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>0.75</td>
<td>[0.7,1]</td>
</tr>
</tbody>
</table>

Table 5. The average outcome and best point in 25 runs.

<table>
<thead>
<tr>
<th>Bat Algorithm(BA)</th>
<th>modified Algorithm(DEBA)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average outcome</td>
<td>Best point (in 25 runs)</td>
</tr>
<tr>
<td></td>
<td>Average outcome</td>
</tr>
<tr>
<td></td>
<td>Best point (in 25 runs)</td>
</tr>
<tr>
<td>Eggcrate’s</td>
<td>3.2908e-006</td>
</tr>
<tr>
<td></td>
<td>1.0e-003 *(0.2468, 0.0282)</td>
</tr>
<tr>
<td></td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>1.0e-162 *(0.4619, 0.2749)</td>
</tr>
<tr>
<td>Rosenbrock’s</td>
<td>4.1887</td>
</tr>
<tr>
<td></td>
<td>(1.0149,0.9691)</td>
</tr>
<tr>
<td></td>
<td>7.5748e-004</td>
</tr>
<tr>
<td></td>
<td>(0.9989,0.9977)</td>
</tr>
<tr>
<td>De Jong’s</td>
<td>7.1686e+002</td>
</tr>
<tr>
<td></td>
<td>(-0.0021, -0.0005, 0.0020, -10.0000,…)_20</td>
</tr>
<tr>
<td></td>
<td>1.0e-161 *(0.0275, 0.0343, -0.0242,-0.00 82 , …)_20</td>
</tr>
<tr>
<td>Griewank’s</td>
<td>61.3870</td>
</tr>
<tr>
<td></td>
<td>(55.5452, -20.1470, -74.2772,-16.7718,…)_30</td>
</tr>
<tr>
<td></td>
<td>1.0e-007 *(0.3053, 0.1121, -0.1837, 0.2883,…)_30</td>
</tr>
</tbody>
</table>

Table 6. The average computational time in 25 runs.

<table>
<thead>
<tr>
<th>Bat Algorithm(BA)</th>
<th>modified Algorithm(DEBA)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average computational time (sec.)</td>
<td></td>
</tr>
<tr>
<td>Eggcrate’s</td>
<td>2.1121</td>
</tr>
<tr>
<td></td>
<td>2.2064</td>
</tr>
<tr>
<td>Rosenbrock’s</td>
<td>2.2918</td>
</tr>
<tr>
<td></td>
<td>2.3724</td>
</tr>
<tr>
<td>De Jong’s</td>
<td>3.5963</td>
</tr>
<tr>
<td></td>
<td>3.2441</td>
</tr>
<tr>
<td>Griewank’s</td>
<td>10.2522</td>
</tr>
<tr>
<td></td>
<td>5.1690</td>
</tr>
</tbody>
</table>
From the experimental results in Table 6 obtained from all test functions, we can see DEBA presents higher accuracy than the original BA on minimizing the outcome as the optimization goal. The experimental results indicate that DEBA improves the accuracy on average than the original BA. From the results in Table 5, when the dimension \((D)\) is not large, the computational time of EBA is almost the same with the original BA’s. In simulation, dimension of Eggcrate’s and Rosenbrock’s function is set to 2. However, when \(D\) is too large, the computational time of DEBA is shorter than the original BA. In simulation, dimension of De Jong’s function and Griewank’s function is respectively set to 256 and 30. Results suggest that the proposed DEBA is very efficient for stochastic multi-objective optimization.

Studying the convergence of the four functions in Figure 5, we can directly observe that the initial solution of DEBA is better than the basic BA obviously, and the convergence speed is significantly better than the basic BA. The improved superiority is fully reflected. Moreover, the exciting results suggest that more studies will highly be needed to carry out the sensitivity analysis, to analyze the rate of algorithm convergence, and to improve the convergence rate even further.
6. Conclusions

In this paper, by combining Doppler effect with the bat algorithm, we have successfully proposed a newly improved BA, which is called Doppler Effect Bat Algorithm (DEBA), for stochastic optimization. By reanalyzing the characteristics of the bat and redefining the corresponding operations based on the basic framework of Bat Algorithm (BA), we can see that new algorithm is more efficient for multi-objective optimization. The experimental results indicate that our proposed DEBA produces a more accurate outcome. We can see it is potentially more powerful than BA and PSO.

The Doppler Effect Bat Algorithm is very efficient. A further improvement on the convergence of the algorithm is to carry out sensitivity studies by varying various parameters such as $\alpha$, $\gamma$ and more interestingly Inertia weight $w$. These could form important topics for further research. In addition, further studies on the application of DEBA in combination with other algorithms may form an exciting area for further research in optimization.

7. References

Neural Networks Based on Genetic Algorithm (GA)", Journal of Computers, Academy Publisher, vol. 6, no. 5, pp.939-946, 2011.


