International Journal of Hybrid Information Technology
Vol.8, No.1 (2015), pp.383-396
http://dx.doi.org/10.14257/ijhit.2015.8.1.34

A Novel Hybrid Bat Algorithm with Differential Evolution Strategy for Constrained Optimization

Xianbing Meng[1, 2], X. Z. Gao[1, 3] and Yu Liu[2]

[1]College of Information Engineering, Shanghai Maritime University, China
[2]Chengdu Green Energy and Green Manufacturing R&D Center, China
[3]Department of Electrical Engineering and Automation, Aalto University School of Electrical Engineering, Finland
x.b.meng12@gmail.com, xiao-zhi.gao@aalto.fi, yu.liu@vip.163.com

Abstract

A novel hybrid Bat Algorithm (BA) with the Differential Evolution (DE) strategy using the feasibility-based rules, namely BADE is proposed to deal with the constrained optimization problems. The sound interferences induced by other things are inevitable for the bats which rely on the echolocation to detect and localize the things. Through integration of the DE strategy with BA, the insects' interferences for the bats can be effectively mimicked by BADE. Moreover, the bats swarm' mean velocity is simulated as the other bats' effects on each bat. Having considered the living environments the bats inhabit, the virtual bats can be lifelike. Experiments on some benchmark problems and engineering designs demonstrate that BADE performs more efficient, accurate, and robust than the original BA, DE, and some other optimization methods.

Keywords: *Bat Algorithm (BA), Differential Evolution (DE), feasibility-based rules, constrained optimization*

1. Introduction

Dealing with the optimization problems using the natural-inspired algorithms, such as the Particle Swarm Optimization (PSO) [1], Differential Evolution (DE) [2], Genetic Algorithms [3], Cuckoo Search [4], Social Spider Optimization [5] and so on, has attracted great research interest during the recent years. The efficiency of the natural-inspired algorithms can be attributed to the fact that they imitate the best features in nature, which have evolved by natural selection over millions of years. From the well-known No-Free-Lunch theorems, no algorithm can adept at every problem. For example, although the DE exhibits good performances on a lot of problems including nonlinear, multimodal, un-constrained, constrained and so on [2, 6]. The performance of the DE heavily depends on the selected generation strategy and related parameter values used. Inappropriate choices of the generation strategies and parameters may lead to premature convergence [7-8].

The natural-inspired algorithms are only based on some abstraction of the nature. As a simplification of the nature, they usually are not perfect. Intuitively speaking, the more the algorithm is true to nature, the more the algorithm can solve problems efficiently and widely. How to extract the intelligence from the nature and design new powerful algorithm is still in progress.

The Bat Algorithm (BA) [9], proposed by Yang in 2010, is inspired by the echolocation characteristics of bats with varying pulse rates of emission and loudness. It can be viewed to link with the Harmony Search and PSO under appropriate conditions, and has shown

ISSN: 1738-9968 IJHIT

promising efficiency for solving a great many optimization applications [10], including image matching [11], uninhabited combat air vehicles [12], classifications [13] and various other problems. However, it may also face the problems about premature convergence [14].

Many studies focus on solving the deficiencies of BA. Wang *et al.* [15] improved the BA by dynamically and adaptively adjusting the virtual bats' flight speed and their flight direction. Li *et al.* [16] proposed a new bat algorithm based on complex-valued encoding to enhance the BA's performance. Gandomi et al. [17] introduce chaos into BA to increase the algorithm's global search mobility. Liu *et al.* [18] further mimicked the bats' echolocation characteristics by introducing the Doppler Effect into BA. All these variants further mimic the behaviors of the bats and successfully enhance the performance of the basic BA.

In this paper, a novel hybrid BA is proposed to overcome the imperfection of BA by further mimicking the living environments the bats inhabit. Using the DE strategy, the other insects' interferences for the bats can be effectively simulated. Moreover, the swarm's mean velocity is simulated as other bats' effects on the bats.

The rest of this paper is organized as follows. Section 2 briefly introduces the background knowledge about the BA, DE, and the feasibility-based rules. The details of our new hybrid optimization algorithm, BADE, and its computational complexity are described in Section 3. Section 4 presents the simulations and comparative study. Some conclusions are provided in Section 5.

2. Related Works

2.1. Bat Algorithm (BA)

Described by its position x_i^t, velocity v_i^t, frequency f_i, loudness A_i^t, and the emission pulse rate r_i^t ($i \in [1, \cdots, N]$), each bat searches its prey in a *D*-dimensional space.

2.1.1. Movement of Virtual Bats: After a random initialization, the new positions (solutions) and velocities at time step *t* are updated as follows:

$$f_i = f_{min} + (f_{max} - f_{min}) \times \beta, \tag{1}$$

$$v_i^{t+1} = v_i^t + (x_i^t - x_*) \times f_i, \tag{2}$$

$$x_i^{t+1} = x_i^t + v_i^{t+1}, \tag{3}$$

Where $\beta \in [0, 1]$ is a random vector drawn from a uniform distribution. x_* is the current global best solution. Initially, each bat is randomly assigned with a frequency in $[f_{min}, f_{max}]$. The values of f_{min} and f_{max} depend on the domain size of the problem of interest.

For the local search part, once a solution is selected among the current best solutions, a new solution for each bat is generated locally according to the following equation:

$$x_{new} = x_{old} + \varepsilon * A^t, \tag{4}$$

Where ε is a random vector drawn from a uniform distribution in the interval [−1, 1]. A^t is the average loudness of all the bats at time step *t*. The global best solution x_* can be updated when the best fitness value obtained by all the *N* bats is superior to the previous $f(x_*)$.

2.1.2. Loudness and Pulse Emission: The loudness A_i^t and emission rates r_i^t decrease and increase respectively, only if the new solutions are updated, which means that these bats are moving towards their prey. These can be formulated as follows.

$$\text{If } (rand\ (0, 1) < A_i^t\ \&\&\ f(x_i) < f(x)) \tag{5}$$

$$f(x) = f(x_i), \tag{6}$$

$$A_i^{t+1} = \alpha A_i^t, \tag{7}$$

$$r_i^{t+1} = r_i^0 (1 - e^{-\gamma t}), \tag{8}$$

Where $rand\ (0,1)$is a random vector drawn from a uniform distribution. α and γ are two constants, α is similar to the cooling factor of a cooling schedule in simulated annealing. The initial loudness A_i^0 and pulse rate r_i^0 are random numbers uniformly distributed in the interval [1, 2] and [0, 1], respectively.

2.2. Differential Evolution (DE)

The DE is an evolutionary algorithm. There exist several variants of the traditional DE. The DE/rand/1/bin scheme of Storn and Price [19] is chosen in this work for a general description. In the DE algorithm, candidate solutions are represented as chromosomes in a vector like $\overrightarrow{x_i} = \left(x_{i,1}, x_{i,2}, \cdots x_{i,D}\right)$, where $i \in [1, \cdots, N]$, $j \in [1, \cdots, D]$. The solutions are randomly initialized in the D-dimensional search space. They can evolve by undergoing simple operations of mutation $\overrightarrow{v_i^t}$, crossover $\overrightarrow{u_i^t}$ and selection $\overrightarrow{x_i^t}$.

$$\overrightarrow{v_i^{t+1}} = \overrightarrow{x_a^t} + F(\overrightarrow{x_b^t} - \overrightarrow{x_c^t}), \tag{9}$$

$$\overrightarrow{u_{i,J}^{t+1}} = \begin{cases} \overrightarrow{v_{i,J}^{t+1}}, if\ rand_j\ (0,1) < CR_j\ or\ j = k\ , \\ \qquad \overrightarrow{x_{i,J}^t}\ ,\ otherwise. \end{cases}, \tag{10}$$

$$\overrightarrow{x_i^{t+1}} = \begin{cases} \overrightarrow{u_i^{t+1}}, if\ f(\overrightarrow{u_i^{t+1}}) < f(\overrightarrow{x_i^t}) \\ \qquad \overrightarrow{x_i^t}\ ,\ otherwise \end{cases}. \tag{11}$$

Where $a \neq b \neq c \neq i,\ a, b, c \in [1, N], k \in [1, D], k$ is randomly selected, $f(\overrightarrow{x_i^t})$ denotes the fitness value of the ith individual at t time step. $F \in (0,2)$ is a positive parameter for scaling the difference solutions. $CR \in (0,1)$ is the crossover rate.

2.3. The Feasibility-based Rules

The feasibility-based rules [20] used in this paper are illustrated as follows.

(1) Any feasible solution is superior to any infeasible solution.

(2) Between two feasible solutions, the one having a better objective function value is preferred.

(3) Between two infeasible solutions, the one having a smaller constraint value is preferred.

To summarize, these rules are to choose a solution that lies closer to the feasible region.

3. Hybrid BA with DE strategy

3.1. The Basic Idea of BADE

Bats use echolocation to detect prey and discriminate different types of insects even in the dark. Hence bats are sensitive to the sounds. Bats usually feed on insects, which can emit sound. In a specific habitat, there exist a group of bats, some of which may simultaneously forage for food. Thus bats may be subjected to the noise and interference induced by their prey and their partners. Comprehensive study has suggested that bats seem to be able to discriminate targets by the variations of the Doppler Effect induced by the wing-flutter rates of the targeted insects [21].

As the simplification of the bats' behaviors, the other bats' and insects' interferences for the bats are not considered in the basic BA. To make the virtual bats resemble the realistic bats is the main idea in this work. In this paper, the living environments the bats inhabit would be integrated into the BA.

Through integration of the mutation operator in the DE/rand/1/bin scheme with the BA, the insects' interferences for the bats can be visually simulated as a stochastic decision. The insects' interferences for the bats only exists when rand (0, 1), a uniform random number in [0, 1], is smaller than *CR*. Here *CR* is the crossover rate in DE. Consider three different individuals interfere with the virtual bats. If the interference is strong enough that the virtual bats cannot distinguish the targets by themselves, they will follow the clues suggested by the interference. Otherwise, they will continue searching for their targets using their own strategies.

For simplicity, the swarm's mean velocity is simulated as the other bats' effects on the bats. Figure 1 is a good case to illustrate that it may contribute to help the virtual bats to find the prey by considering the swarm's mean velocity.

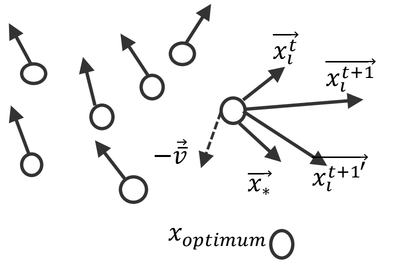

Figure 1. The Swarm's Trajectory

Without considering the swarm's mean velocity, the bat *i* will be located at $\overrightarrow{x_i^{t+1}}$ according to the equations (1), (2) and (3). However, following this direction, bat *i* will be far away from the optimal solution. If the opposite swarm's mean velocity is considered, the situation will be improved. Compared with $\overrightarrow{x_i^{t+1}}$, the new location (solution) $\overrightarrow{x_i^{t+1'}}$ can be closer to the optimal solution than x_i^{t+1}, when we use the equations (1), (12) and (3) to update the new location (solution).

$$v_i^{t+1} = w \times \left(v_i^t - w_2 \times \bar{v}\right) + (x_i^t - x_*) \times f_i, \tag{12}$$

Where $w \in (0,1), w_2 \in (-1,1)$ are two random vectors drawn from a uniform distribution.

Moreover, consider a following situation. Bats would not find a better food than their previous ones during several time step (*G*), and they have eaten up their food. Then they may go to another place to forage for food. For simplicity, this situation can be simulated as follows. All the bats' loudness would be initialized again. Their pulse rates can be temporarily set to a high value, hence the bats can search globally.

Using the criteria above, the virtual bats in the proposed algorithm can be more lifelike than the ones in basic BA, thus helping them escape from the local optima.

3.2. Main Procedure of BADE

The main procedure of the BADE can be described as follows.

Step 1 Initialization

1.1 Initialize *N* bats' positions x_i, velocities v_i $(i \in [1,\cdots,N])$ in a *D*-dimensional space, and initialize the associated parameters, such as frequency f_i, pulse rates r_i, and the loudness A_i.

1.2 Evaluate the fitness value of each bat by the objective function $f(x)$ and the constraint value of each bat by the constrained functions.

```
Step 2 Update solutions.
While $t$ < Max number of iterations ($M$)
Step 2.1 Generate offspring (solutions) $x_i^{t+1}$ using the equations (1), (12) and (3).
Step 2.2 If ($rand\ (0,1) > r_i$)
Select a solution among the best solutions
Generate a local solution $x_i^{t+1}$ around the selected best solution using the equation (4).
End if
Evaluate the fitness values and constraint values of the offspring $x_i^{t+1}$.
Step 2.3 If ($rand\ (0,1) < CR$)
Generate offspring $x_i^{t+1'}$using the equation (9).
Evaluate the fitness values and constraint values of the new offspring $x_i^{t+1'}$.
Select the final offspring $x_i^{t+1}$ by comparing the fitness value and constraint value of $x_i^{t+1}$ with those of $x_i^{t+1'}$ according to the feasibility-based rules.
End if
Step 2.4 If ($rand\ (0,1) < A_i$)
If $x_i^t$ is infeasible, but $x_i^{t+1}$ is feasible
Or both $x_i^t$, $x_i^{t+1}$are feasible, but $f\left(x_i^t\right) > f(x_i^{t+1})$
Or both $x_i^t$, $x_i^{t+1}$are infeasible, but constraint value of $x_i^t$ is bigger than that of $x_i^{t+1}$.
Accept the offspring $x_i^{t+1}$as the new solutions.
Increase $r_i$, and reduce $A_i$ using the equations (7) and (8).
End if
Step 2.5 Rank the bats and find the current best $x_*$.
Step 2.6 If $x_*$ does not improve in $G$ generations.
Reinitialize the loudness $A_i$, and set the pulse rates $r_i$, which is a uniform random number between [0.85, 0.9].
$t = t + 1$.
End while
Step 3 Update the best $x_*$.
```

3.3. Computational Complexity of BADE

The computational complexity of the BADE can be estimated. M stands for maximal iteration number. In the BADE, the 'worst-case' complexities are described as follows.

1) In Step 1.1, the time complexity of initializing N bats is $O(2ND + 5N)$.

2) In Step 1.2, evaluating N bats' fitness values and constraint values is $O(ND)$.

3) In Step 2, the time complexity is $O(MN)$.

4) In other steps, the computational complexity is rather simple, and can be neglected.

To summarize, the overall computational complexity of the BADE is $O(MN)$.

4. Validation and Comparison

In order to verify the efficiency of the BADE, and illustrate its superiority over other methods, some well-known benchmark problems and real-world engineering designs are used to investigate the performance of the algorithms. In all case studies, the statistical results have been obtained, based on 100 independent trials for each of the algorithms. The number of iterations is 1,000 in each trial. For a fair comparison, the population size in each algorithm is set to be the same. Table 1 presents the details about the parameters of the BADE, BA and DE.

Table 1. The Parameters' Values of BADE, BA and DE

Algorithm	Parameters
BADE	$\alpha = \gamma = 0.9$, $f_{min} = 0$, $f_{max} = 2$, $A_0 \in [0, 2], r_0 \in [0, 1]$, $w \in (0.1, 1), w_2 \in (-1, 1), G = 20$
BA	$\alpha = \gamma = 0.9$, $f_{min} = 0$, $f_{max} = 2$, $A_0 \in [0, 2], r_0 \in [0, 1]$
DE	$CR = 0.9$, $F = 0.6$

4.1. Benchmark Problems

In this section, a total of nine benchmark problems [22-23] are used to demonstrate the BADE's superiority over the basic BA and DE. These problems are presented as follows.

F1. Minimize $f(\vec{x}) = 5\sum_{i=1}^{4} x_i - 5\sum_{i=1}^{4} x_i^2 - \sum_{i=5}^{13} x_i$

Subject to $g_1(\vec{x}) = 2x_1 + 2x_2 + x_{10} + x_{11} - 10 \leq 0$,
$g_2(\vec{x}) = 2x_1 + 2x_3 + x_{10} + x_{12} - 10 \leq 0$,
$g_3(\vec{x}) = 2x_2 + 2x_3 + x_{11} + x_{12} - 10 \leq 0$,
$g_4(\vec{x}) = -8x_1 + x_{10} \leq 0$,
$g_5(\vec{x}) = -8x_2 + x_{11} \leq 0$,
$g_6(\vec{x}) = -8x_3 + x_{12} \leq 0$,
$g_7(\vec{x}) = -2x_4 - x_5 + x_{10} \leq 0$,
$g_8(\vec{x}) = -2x_6 - x_7 + x_{11} \leq 0$,
$g_9(\vec{x}) = -2x_8 - x_9 + x_{12} \leq 0$.

Where the bounds are $0 \leq x_i \leq 1$ ($i \in [1, \cdots, 9]$), $0 \leq x_i \leq 100$ (i = 10, 11, 12), and $0 \leq x_{13} \leq 1$, the global optimum is at $x^* = (1,1,1,1,1,1,1,1,1,3,3,3,1)$, where $f(x_*) = -15$.

F2. Minimize $f(\vec{x}) = -|(\sum_{i=1}^{n} cos^4(x_i) \; - \; 2\prod_{i=1}^{n} cos^2(x_i))/\sqrt{\sum_{i=1}^{n} ix_i^2}\,|$

Subject to $g_1(\vec{x}) = 0.75 - \prod_{i=1}^{n} x_i \leq 0$
$g_2(\vec{x}) = \sum_{i=1}^{n} x_i - 0.75n \leq 0$

Where n = 20, $0 \leq x_i \leq 10$ ($i \in [1, \cdots, n]$), the optimal solution is $f(x_*) = -0.803619$.

F3. Minimize $f(\vec{x}) = -(\sqrt{n})^n \prod_{i=1}^{n} x_i$

Subject to $h(\vec{x}) = \sum_{i=1}^{n} x_i^2 - 1 = 0$

Where n = 10 and $0 \leq x_i \leq 10$ ($i \in [1, \cdots, n]$), the global optimum is at $x_* = 1/\sqrt{n}$, where $f(x^*) = -1$.

F4. Minimize $f(\vec{x}) = 3x_1 + 10^{-6}x_1^3 + 2x_2 + 2 \times 10^{-6}x_2^3/3$

Subject to $g_1(\vec{x}) = -(x_1 - 5)^2 - (x_2 - 5)^2 + 100 \leq 0$,
$g_2(\vec{x}) = (x_1 - 6)^2 + (x_2 - 5)^2 - 82.81 \leq 0$.

Where $13 \leq x_1 \leq 100$ and $0 \leq x_2 \leq 100$, the optimal solution is at x_*= (14.095, 0.84296), where $f(x_*) = -6961.8138$.

F5. Minimize $f(\vec{x}) = x_1^2 + x_2^2 + x_1x_2 - 14x_1 - 16x_2 + (x_2 - 10)^2 + 4(x_4 - 5)^2 + (x_5 - 3)^2 + 2(x_6 - 1)^2 + 5x_7^2 + 7(x_8 - 11)^2 + 2(x_9 - 10)^2 + (x_{10} - 7)^2 + 45$

Subject to $g_1(\vec{x}) = 4x_1 + 5x_2 - 3x_7 + 9x_8 - 105 \leq 0$,
$g_2(\vec{x}) = 10x_1 - 8x_2 - 17x_7 + 2x_8 \leq 0$,
$g_3(\vec{x}) = -8x_1 + 2x_2 + 5x_9 - 2x_{10} - 12 \leq 0$,
$g_4(\vec{x}) = 3(x_1 - 2)^2 + 4(x_2 - 3)^2 + 2x_3^2 - 7x_4 - 120 \leq 0$,
$g_5(\vec{x}) = 5x_1^2 + 8x_2 + (x_3 - 6)^2 - 2x_4 - 40 \leq 0$,
$g_6(\vec{x}) = x_1^2 + 2(x_2 - 2)^2 - 2x_1x_2 + 14x_5 - 6x_6 \leq 0$,
$g_7(\vec{x}) = 0.5(x_1 - 8)^2 + 2(x_2 - 4)^2 + 3x_5^2 - x_6 - 30 \leq 0$,
$g_8(\vec{x}) = -3x_1 + 6x_2 + 12(x_9 - 8)^2 - 7x_{10} \leq 0$.

Where $n = 10$ and $-10 \leq x_i \leq 10$ ($i \in [1, \cdots, n]$), the global optimum is at $x_* =$ (2.171996,2.363683,8.773926,5.095984,0.9906548,1.430574,1.321664,9.828726,8.280092,8.375927), where $f(x_*) = 24.3062091$.

F6. Minimize $f(\vec{x}) = \sum_{i=1}^{3} x_i$

Subject to $g_1(\vec{x}) = -1 + 0.0025(x_4 + x_6) \leq 0,$

$g_2(\vec{x}) = -1 + 0.0025(x_5 + x_7 - x_4) \leq 0,$

$g_3(\vec{x}) = -1 + 0.01(x_8 - x_5) \leq 0,$

$g_4(\vec{x}) = -x_1 x_6 + 833.33252 x_4 + 100 x_6 - 83333.333 \leq 0,$

$g_5(\vec{x}) = -x_2 x_7 + 1250 x_5 + x_2 x_4 - 1250 x_4 \leq 0,$

$g_6(\vec{x}) = -x_3 x_8 + 1250000 + x_3 x_5 - 2500 x_5 \leq 0.$

Where$100 \leq x_1 \leq 10000$, $1000 \leq x_i \leq 10000$ (i=2, 3) and $10 \leq x_i \leq 1000$ ($i \in [4, \cdots, 8]$), the optimal solution is at x_* = (579.19, 1360.13, 5109.92, 182.0174, 295.5985, 217.9799, 286.4, 395.5979), where $f(x_*) = 7049.248$.

F7. Minimize $f(\vec{x}) = x_1^2 + (x_2 - 1)^2$

Subject to $h(\vec{x}) = x_2 - x_1^2 = 0$.

Where $-1 \leq x_1, x_2 \leq 1$, the optimal solution is at $x_* = (\pm 1/\sqrt{2}, 0.5)$, where $f(x_*) = 0.75$.

F8. Minimize $f(\vec{x}) = (x_1 - 2)^2 + (x_2 - 1)^2$

Subject to $g_1(\vec{x}) = x_1 - 2x_2 + 1 = 0$

$g_2(\vec{x}) = 0.25 x_1^2 + x_2^2 - 1 \leq 0$

Where$-10 \leq x_1, x_2 \leq 10$, the optimal solution is at $x_* = (0.82288, 0.91144)$,where $f(x_*) =$ 1.3935.

F9. Minimize $f(\vec{x}) = (x_1^2 + x_2 - 11)^2 + (x_1 + x_2^2 - 7)^2$

Subject to $g_1(\vec{x}) = (x_1 - 0.05)^2 + (x_2 - 2.5)^2 - 4.84 \leq 0$

$g_2(\vec{x}) = 4.84 - {x_1}^2 - (x_2 - 2.5)^2 \leq 0$

Where $0 \leq x_1, x_2 \leq 6$, the optimal solution is at $x_* = (2.246826, 2.381865)$, where $f(x_*) =$ 13.59085.

Table 2 presents the statistical results obtained by the BADE, BA and DE. It clearly shows that the proposed algorithm outperforms the BA and DE for solving all these problems in terms of optimization accuracy and robustness.

As illustrative examples, two complex problems are chosen to show the convergence performance of the three algorithms. Note that different strategies of the vertical axis are used in Figure 2 and Figure 3 to obtain a visible comparison result. In Figure 3, a logarithmic (base 10) scale is used for the vertical axis, while the vertical axis in Figure 2 is divided by the optimal value of the problem F1.

It can be figured out that BADE can yield a better convergence performance than that of the BA and DE. Thus the proposed algorithm successfully enhances the bat algorithm's performance.

Table 2. Optimization Results Obtained by BADE, BA and DE

Problem	Algorithm	Best	Mean	Worst	Std.
F1	BA	-14.895416	-14.123807	-13.281171	0.0326253
(Optimum	BADE	-15.000000	-15.000000	-15.000000	0
-15)	DE	-15.000000	-14.309998	-5.000000	0.215554
F2	BA	-0.415492	-0.288233	-0.188259	0.00511553
(Optimum	BADE	-0.803551	-0.774184	-0.627821	0.00301232
-0.803619)	DE	-0.802466	-0.763126	-0.622109	0.00377503
F3	BA	-0.146877	-0.006375	-0.000000	0.002418

(Optimum	BADE	-1.000000	-0.999989	-0.999587	4.25983e-6
-1)	DE	-0.9999998	-0.9999910	-0.9999398	1.17517e-6
F4	BA	-6955.586103	-6861.925196	-6747.332675	3.97447
(Optimum	BADE	-6961.813876	-6961.813876	-6961.813876	3.72944e-13
-6961.81388)	DE	-6961.813876	-5636.952741	-550.011046	298.533
F5	BA	31.202667	37.812975	44.376195	0.256523
(Optimum	BADE	24.306516	24.309069	24.325009	0.000328386
24.306)	DE	24.307619	24.452326	25.078375	0.0281264
F6	BA	7641.352525	15886.42796	29999.28351	511.12
(Optimum	BADE	7049.330285	7050.769000	7098.552165	0.508031
7049.3307)	DE	7049.66928	7380.71174	11010.0000	79.1739
F7	BA	0.750021	0.752844	0.756825	0.000175376
(Optimum	BADE	0.75	0.75	0.75	1.12144e-18
0.75)	DE	0.75	0.750704	0.778339	0.0003717
F8	BA	1.397637	1.455812	1.534232	0.00340042
(Optimum	BADE	1.393465	1.393465	1.393465	1.11424e-16
1.3935)	DE	1.393465	1.394360	1.454966	0.000675
F9	BA	13.598136	13.684625	13.864751	0.00569281
(Optimum	BADE	13.590842	13.590842	13.590842	1.14132e-15
13.59085)	DE	13.590842	15.558862	67.422811	0.784452

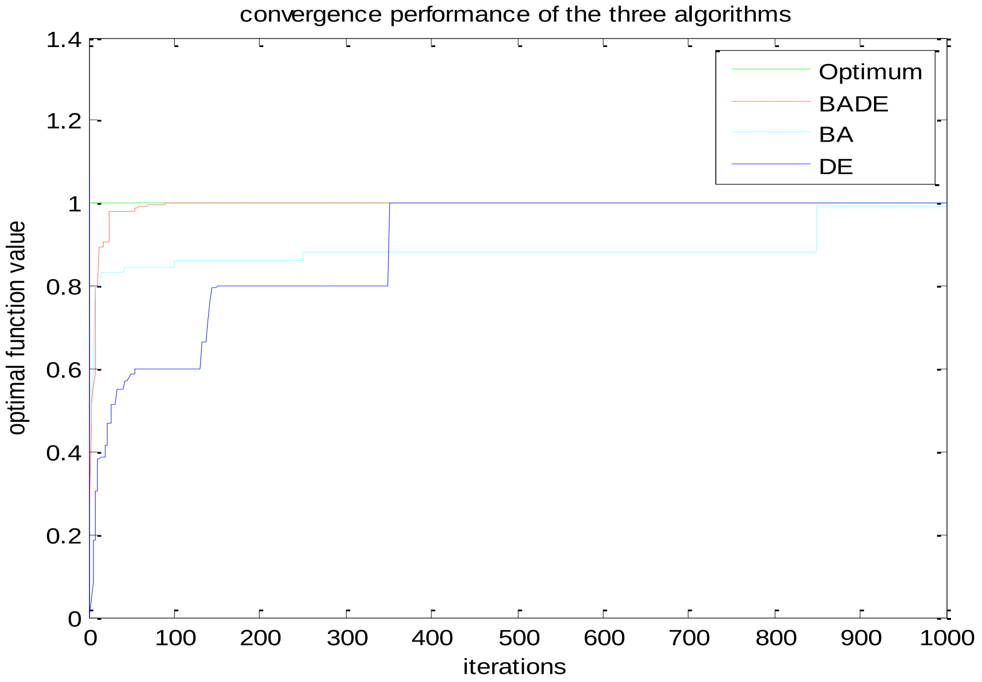

Figure 2. Convergence Curves of F1

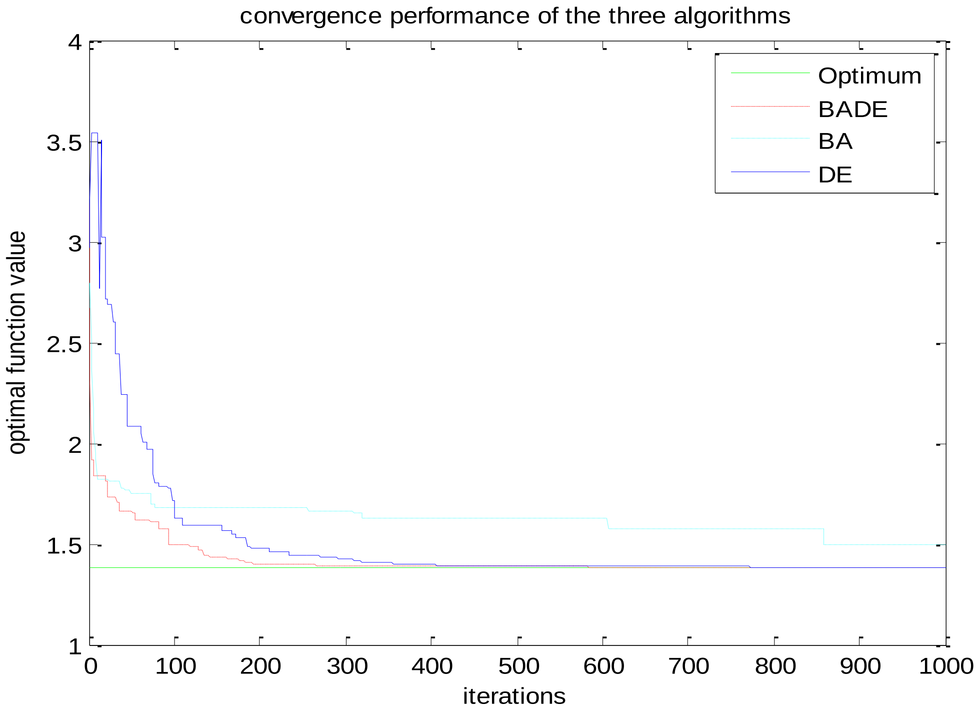

Figure 3. Convergence Curves of F5

4.2. Applications of BADE in Engineering Design

(1) Cantilever beam. The structure of a cantilever beam with square cross section [4] is shown in Figure 4. The design variables are the heights (or widths) of the different beam elements. The beam is rigidly supported at node 1. There exist five nodes. A given vertical force acts at node 6. Here, the thickness (*t*), which is set to 2/3, is held fixed.

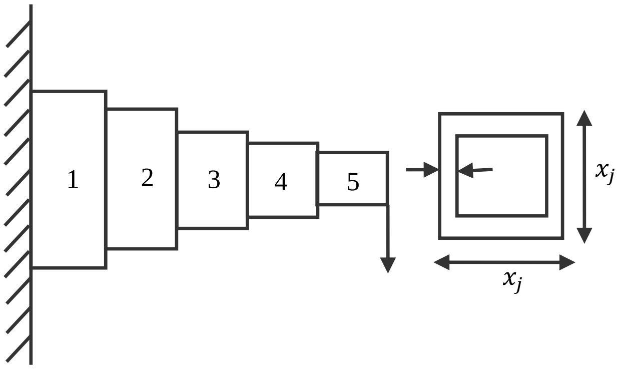

Figure 4. Cantilever Beam

This design problem can be described as follows.

Minimize $f(\vec{x}) = 0.0624 \sum_{i=1}^{5} x_i$

Subject to $g_1(\vec{x}) = 61/x_1^3 + 37/x_2^3 + 19/x_3^3 + 7/x_4^3 + 1/x_5^3 - 1 \leq 0$

Table 3 shows the optimization results obtained by BADE and other algorithms. It clearly shows that the solution achieved by the BADE is slightly better than that of others. The constraint value is g_* = -7.12208e-13, which means the best solution achieved by BADE is feasible.

Table 3. Optimization Results of the Cantilever Beam Design

	Chickermane *et al.* [24]			Gandomi et al. [4]	BADE
	MMA	GCA(I)	GCA(II)		
x_1	6.010	6.010	6.010	6.0089	6.015810
x_2	5.300	5.300	5.300	5.3049	5.309177
x_3	4.490	4.490	4.490	4.5023	4.494092
x_4	3.490	3.490	3.490	3.5077	3.501817
x_5	2.150	2.150	2.150	2.1504	2.152765
Worst	N/A	N/A	N/A	N/A	1.33995636
Mean	N/A	N/A	N/A	N/A	1.33995636
Std.	N/A	N/A	N/A	N/A	7.4604e-11
Best	1.340	1.340	1.340	1.33999	1.33995636

(2) Welded beam design. The schematic of the welded beam [3], shown in Figure 5, is described by four design variables, which are *h* (x_1), *l* (x_2), *t* (x_3), and *b* (x_4). This design is to find a minimum total fabricating cost, subject to the constraints on the shear stress τ, bending stress in the beam σ, buckling load on the bar P_c, and deflection of the beam δ, and side constraints.

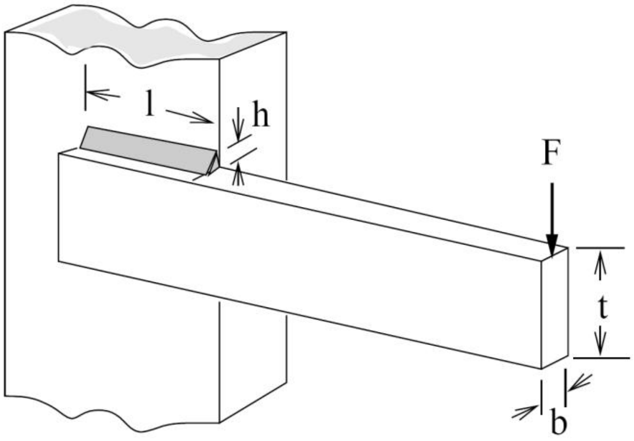

Figure 5. Welded Beam Design

The problem can be explained as follows.

Minimize $f(\vec{x}) = 1.10471x_1^2x_2 + 0.04811x_3x_4(x_2 + 14)$

Subject to $g_1(\vec{x}) = \tau(x) - \tau_{max} \leq 0$

$g_2(\vec{x}) = \sigma(x) - \sigma_{max} \leq 0$

$g_3(\vec{x}) = x_1 - x_4 \leq 0$

$g_4(\vec{x}) = \delta(x) - \delta_{max} \leq 0$

$g_5(\vec{x}) = P - P_c(x) \leq 0$

Where $\tau(\vec{x}) = \sqrt{(\tau')^2 + 2\tau'\tau''x_2/(2R) + (\tau'')^2}$, $\tau' = P/(\sqrt{2}x_1x_2), \tau'' = MR/J, M = P(L + 0.5x_2)$, $R = \sqrt{0.25x_2^2 + 0.25(x_1 + x_3)^2}$, $J = 2\{\sqrt{2}x_1x_2[x_2^2/12 + 0.25(x_1 + x_3)^2]\}$, $\sigma(\vec{x}) = 6PL/(x_3^2x_4)$, $\delta(\vec{x}) = 4PL^3/(Ex_3^3x_4)$, $P_c(\vec{x}) = 4.013E/L^2\sqrt{\frac{x_3^2x_4^6}{36}}(1 - x_3/(2L)\sqrt{E/(4G)})$, P=6000lb, L=14in, $\delta_{max} = 0.25\ in$, E=30,000,000psi, G=12,000,000, $\tau_{max} = 13{,}600psi$, $\sigma_{max} = 30{,}000psi$, $0.125 \leq x_1 \leq 5, 0.1 \leq x_2, x_3 \leq 10, 0.1 \leq x_4 \leq 5$.

Table 4. Optimization Results of Welded Beam Design

	Coello, et al [3]	Coello, et al. [25]	He, et al. [1]	Zhang, et al. [6]	Cuevas, et al. [5]	BADE
x_1	0.205986	0.205700	0.202369	N/A	N/A	0.205730
x_2	3.471328	3.470500	3.544214	N/A	N/A	3.470489
x_3	9.020224	9.036600	9.048210	N/A	N/A	9.036624
x_4	0.206480	0.205700	0.205723	N/A	N/A	0.205730
Worst	1.993408	3.179709	1.782143	1.724853	1.7993318	1.724852
Mean	1.792654	1.971809	1.748831	1.724852	1.7464616	1.724852
Std.	0.074713	0.443131	0.012926	2.1e−7	0.02572985	2.22e-16
Best	1.728226	1.724852	1.728024	1.724852	1.7248523	1.724852

Table 4 summarizes a comparison of the results obtained by the BADE and other optimization methods [1, 3, 5-6, 25]. BADE can yield a better result than the results reported by Coello *et al.* [3] and He *et al.* [1], and has a comparable optimization accuracy with that reported by Coello *et al.* [25], Zhang *et al.* [6] and Cuevas *et al.* [5]. However, BADE performs more stable than all the methods. The best solution obtained by BADE is feasible. The constraint values are g_* = (0, 0, -5.55112e-17, -0.23554, -2.72848e-12). It can be draw the conclusion that BADE outperforms all the methods in these literatures for solving this problem.

(3) Coil compression spring design. The coil compression spring [26] (as shown in Figure 6) is to minimize the weight of the compression spring, subject to the constraints on the minimum deflection, shear stress, surge frequency, diameter and design variables. The design variables are the mean coil diameter *D (x1)*, the wire diameter *d (x2)*, and the number of active coils *N (x3)*.

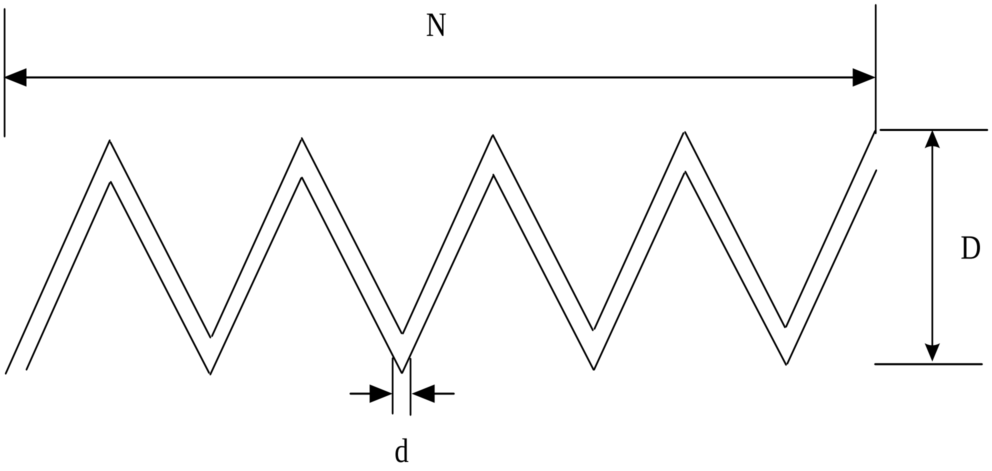

Figure 6. Coil Compression Spring Design

This problem can be formulated as follows.

Minimize $f(\vec{x}) = x_1^2 x_2 (x_3 + 2)$

Subject to $g_1(\vec{x}) = 1 - x_2^3 x_3 / (71782 x_1^4) \leq 0$

$g_2(\vec{x}) = (4x_2^2 - x_1 x_2)/(12566(x_1^3 x_2 - x_1^4)) + 1/(5108 x_1^2) - 1 \leq 0$

$g_3(\vec{x}) = 1 - 140.45 \mathrm{x}_1/(x_2^2 x_3) \leq 0$

$g_4(\vec{x}) = (x_1 + x_2)/1.5 - 1 \leq 0$

Where $0.25 \leq x_1 \leq 1.3, 0.05 \leq x_2 \leq 1.3, 2 \leq x_3 \leq 15.$

Table 5. Optimization Results of Coil Compression Spring Design

	He, et al. [1]	Gao, et al. [26]	Zhang, et al. [6]	Ali, et al. [27]	Cuevas, et al. [5]	BADE
x_1	0.051728	0.051678	N/A	0.05169280	N/A	0.051689
x_2	0.357644	0.356436	N/A	0.35680768	N/A	0.356718
x_3	11.244543	11.306441	N/A	11.28369666	N/A	11.288966
Worst	0.012924	N/A	0.012769	N/A	0.01286792	0.0126656
Mean	0.012730	N/A	0.012680	N/A	0.01276489	0.0126652
Std.	5.1985e-5	N/A	2.7e−5	N/A	0.000093	3.9551e-9
Best	0.0126747	0.0126665	0.012665	0.01266523	0.01266523	0.0126652

Table 5 shows the statistical results obtained by BADE and several other methods [1, 5-6, 26-27]. According to the comparison results, the proposed algorithm outperforms the methods reported by He *et al.* [1] and Gao *et al.* [26] in terms of optimization accuracy and robustness. Though the results reported by Zhang *et al.* [6], Ali *et al.* [27] and Cuevas *et al.* [5] have comparable optimization accuracy with that of BADE, BADE shows its superiority over these methods in terms of robustness. The best result achieved by BADE is feasible. The constraint values are g_* = (0, 0, -4.05379, -0.727729).

5. Conclusion

A novel hybrid bat algorithm, BADE, is proposed in this paper to enhance the performance of the basic BA. Using the mutation operator in the DE within the BA framework, the insects' interferences for bats can be effectively simulated. The other bats' effects on the bats can be simulated by adding the swarm's mean velocity to the velocity equation in BA. Having considered the interference induced by other bats and insects, the virtual bats in BADE can be more lifelike than those in BA. Experimental results on a total of nine benchmark problems and three engineering design problems demonstrate that the BADE perform more efficient and robust than the basic BA, DE, and a few other methods. How to use BADE to solve multi-objective optimization problems is worth doing in the future.

Acknowledgements

This research work was funded by the New Academic Staffs Program of Shanghai Maritime University under Grant GK2013089. X. Z. Gao's research was also funded by the Academy of Finland under Grants 135225, Finnish Funding Agency for Technology and Innovation (TEKES), and the Program for Professor of Special Appointment (Eastern Scholar) at the Shanghai Institutions of Higher Learning. The authors also gratefully acknowledge the useful suggestions of the reviewers and editors, which have processed and improved the presentation

References

[1] Q. He and L. Wang, "An effective co-evolutionary particle swarm optimization for constrained engineering design problems", Engineering Applications of Artificial Intelligence, vol. 20, **(2007)**, pp. 89-99.

[2] S. Das and P. N. Suganthan, "Differential evolution: a survey of the state-of-the-art", IEEE Transactions on Evolutionary Computation, vol. 15, no. 1, **(2011)**, pp. 4-31.

[3] C. A. C. Coello and E. M. Montes, "Constraint-handling in genetic algorithm through the use of dominance-based tournament selection", Advanced Engineering Informatics, vol. 16, **(2002)**, pp. 193-203.

[4] A. H. Gandomi, X. S. Yang and A. H. Alavi, "Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems", Engineering with Computers, vol. 29, **(2013)**, pp. 17-35.

[5] E. Cuevas and M. Cienfuegos, "A swarm optimization algorithm inspired in the behavior of the social-

spider", Expert Systems with Applications, vol. 41, **(2014)**, pp. 412-425.

[6] G. X. Zhang, J. X. Cheng, M. Gheorghe and Q. Meng, "A hybrid approach based on differential evolution and tissue membrane systems for solving constrained manufacturing parameter optimization problems", Applied Soft Computing, vol. 13, **(2013)**, pp. 1528-1542.

[7] J. Lampinen and I. Zelinka, "On stagnation of the differential evolution algorithm", Proceedings of 6th International Mendel Conference on Soft computing, **(2000)** June 07-09, Brno, Czech Republic.

[8] A. K. Qin, V. L. Huang, and P. N. Suganthan, "Differential evolution algorithm with strategy adaptation for global numerical optimization", IEEE Transactions on Evolutionary Computation, vol. 13, no. 2, **(2009)**, pp. 398-417.

[9] X. S. Yang, "A new metaheuristic bat-inspired algorithm", Nature Inspired Cooperative Strategies for Optimization (NICSO 2010), vol. 284, **(2010)**, pp. 65-74.

[10] X. S. Yang, "Bat algorithm: literature review and applications", International Journal of Bio-Inspired Computation, vol. 5, no. 3, **(2013)**, pp. 141-149.

[11] J. W. Zhang and G. G. Wang, "Image matching using a bat algorithm with mutation", Applied Mechanics and Materials, vol. 203, **(2012)**, pp. 88-93.

[12] Y. G. Li and J. P. Peng, "An improved bat algorithm and its application in multiple UCAVs", Applied Mechanics and Materials, vol. 442, **(2014)**, pp. 282-286.

[13] S. Mishra, K. Shaw and D. Mishra, "A New Meta-heuristic Bat Inspired Classification Approach for Microarray Data", Procedia Technology, vol. 4, **(2012)**, pp. 802-806.

[14] X. S. Yang, Nature-inspired optimization algorithm, Elsevier, **(2014)**.

[15] X. W. Wang, W. Wang and Y. Wang, "An Adaptive Bat Algorithm", Proceedings of 9th International Conference on Intelligent Computing Theories and Technology, **(2013)**, pp. 216-223.

[16] L. L. Li and Y. Q. Zhou, "A novel complex-valued bat algorithm", Neural Computing and Applications, **(2014)**.

[17] A. H. Gandomi and X. S. Yang, "Chaotic bat algorithm", Journal of Computational Science, vol. 5, **(2014)**, pp. 224-232.

[18] G. H. Liu, H. Y. Huang, S. M. Wang and Z. X. Chen, "An Improved Bat Algorithm with Doppler Effect for Stochastic Optimization", International Journal of Digital Content Technology and its Applications, vol. 6, no. 21, **(2012)**, pp. 326-336.

[19] R. Storn and K. V. Price, "Differential Evolution: a simple and Efficient Heuristic for Global Optimization over Continuous Spaces", Journal of Global Optimization, vol. 11, **(1997)**, pp. 341-359.

[20] G. T. Pulido and C. A. C. Coello, "A constraint handling mechanism for particle swarm optimization", Congress on Evolutionary Computation, **(2004)** June 19-23, Portland Marriott Downtown, Portland.

[21] J. D. Altringham, Bats: Biology and Behavior, Oxford University Press, Oxford **(2006)**.

[22] T. P. Runarsson and X. Yao, "Stochastic ranking for constrained evolutionary optimization", IEEE Transactions on Evolutionary Computation, vol. 4, no. 3, **(2000)**, pp. 284-294.

[23] M. Jaberipour and E. Khorram, "Two improved harmony search algorithms for solving engineering optimization problems", Commun Nonlinear Sci Numer Simulat, vol. 15, **(2010)**, pp. 3316-3331.

[24] H. Chickermane and H. C. Gea, "Structural optimization using a new local approximation method", International Journal for Numerical Methods Engineering, vol. 39, no. 5, **(1996)**, pp. 829-846.

[25] C. A. C. Coello and R. L. Becerra, "Efficient evolutionary optimization through the use of a cultural algorithm", Engineering Optimization, vol. 36, no. 2, **(2004)**, pp. 219-236.

[26] X. Z. Gao, X. Wang, S. J. Ovaska and H. X, "A modified harmony search method in constrained optimization", International Journal of Innovative Computing, Information and Control, vol. 6, no. 9, **(2010)**, pp. 4235-4247.

[27] M. Z. Ali and N. H. Awad, "A novel class of niche hybrid Cultural Algorithms for continuous engineering optimization", Information Sciences, vol. 267, **(2014)**, pp. 158–190.

Authors

X. B. Meng (1988-), M. S. Student. He is studying at Shanghai Maritime University. His research interests focus on soft computing and its application.

X .Z. Gao (1972-), Prof. Gao is working at Aalto University. His research interests focus on soft computing and its application.

Yu Liu (1980-), Dr. Yu Liu is working at Chengdu Green Energy and Green Manufacturing R&D Center. His present research interests include: Precision Mechatronics, Additive Manufacturing, Printing Technique, Scanning Probe Microscopy and Soft Computing.