Intelligent Automation And Soft Computing, 2019
Copyright © 2019, TSI® Press
Vol. 25, no. 4, 815–825
https://doi.org/10.31209/2019.100000085

A Hybrid GABC-GA Algorithm for Mechanical Design Optimization Problems

Hui Zhi[1,2], Sanyang Liu[1]

[1]School of Mathematics and Statistics, Xidian University, 710126, Xian, China
[2]School of Huaqing and Xi-an University of Architecture and Technology, 710055, Xian, China

ABSTRACT
In this paper, we proposed a hybrid algorithm, which is embedding the genetic operators in the global-best-guided artificial bee colony algorithms called GABC-GA to solve the nonlinear design optimization problems. The genetic algorithm has no memory function and good at find global optimization with large probability, but the artificial bee colony algorithm not have selection, crossover and mutation operator and most significant at local search. The hybrid algorithm balances the exploration and exploitation ability further by combining the advantages of both. The experimental results of five engineering optimization and comparisons with existing approaches show that the proposed approach is outperforms to those typical approaches in terms of the quality of the results solutions in most cases.

KEYWORD: artificial bee colony algorithm; genetic algorithm; global best guided; crossover and mutation operation; mechanical design optimization;

1 INTRODUCTION

MANY optimization problems in science and engineering disciplines can be expressed as Constraint optimization problems (COPs).Without loss of generality, the nonlinear programming (NLP) problem can be formulated as follows:

$$\min f(x)\ \ x=(x_1,x_2,\ldots,x_n)\in S\subseteq R^n \tag{1}$$

where S denotes the search space, which defined as an n -dimensional rectangle in R^n . This rectangle R^n has domains size such that:

$$l(i)\le x_i\le u(i),\ 1\le i\le n.$$

the feasible region S is form by a set of linear or nonlinear constraints as follows:

$$g_j(x)\le 0,\ g_j:R^n\to R,\ j=1,2,\cdots q,$$

$$h_j(x)=0,\ h_j:R^n\to R,\ j=1,2,\cdots p,$$

where q is the number of inequalities, and p is the number of equalities. Usually in COPs, equalities can be replaced by inequalities and thus the problem is composed of inequality constraints only. Accordingly, the non-linear constrained optimization problem can be written as:

$$\begin{aligned}&\min f(x),\ \ x\in S\subseteq R^n\\ &\text{s.t. } g_j(x)\le 0,\ \ j=1,2,\ldots q+p\\ &\quad l(i)\le x_i\le u(i),\quad 1\le i\le n\end{aligned} \tag{2}$$

To solve the COPs problem, many researchers were developed some deterministic methods for solving constraint problems, such as feasible direction approach and generalized gradient descent method. However, due to its limited application and the complexity of constraints, most of the problems like structural optimization problems, economic optimization, and engineering design problems, which inherently involve many difficult and complex requirements to satisfy. These optimization problems can be difficult to solve with traditional mathematical methods. In order to overcome these shortcomings, researchers have proposed many optimization methods to solve these constrained optimization problems, and the meta-heuristic optimization algorithm receives the most concerned. Meta-heuristic optimization algorithm are independent of problems and models when used, and are very efficient and flexible (Baykasoğlu, 2015).

CONTACT Hui Zhi xzzhihui@163.com, zhihui1778@163.com

As a more effective method than traditional mathematical methods, meta-heuristic optimization techniques can accurately explore and discover promising areas in the search space, these methods are well suited for global search. Some of meta-heuristic algorithms developed in recent years which is mainly inspires by natural phenomena and biological behavior, include genetic algorithms, particle swarm optimization (Wang and Yang, 2016), differential evolution (Yu, 2018), artificial bee algorithm (You, et al., 2017), firefly algorithm, grey wolf optimization algorithm, ant colony optimization (Xu, et al., 2016), cuckoo search algorithm, and they have been successful in solve various optimization problems. As for instance, Altalhi (2016) and Coello(2000) using genetic algorithms in engineering design optimization and verify the optimized "best". Yu (2018) presents differential evolution algorithms for constrained multi-objective optimization problems. Xu et. al. (2018) using differential evolution and its various strategies applied for constrained optimization problems. Kim (2010) and Ngo (2017) proposed an efficient PSO algorithm for engineering optimization problems. Liu (2018) proposed a parallel boundary search particle swarm optimization approach for COPs, perform simulation in engineering design problems and indicate the results efficiency. Ouyang (2017) proposed improved PSO for global optimization problems. Ariyasingha and Fernando (2017) used a modified Pareto strength ant colony optimization algorithm to solved multi-objective optimization problems. Tian and Dong (2017) proposed PSO-FWAC algorithm to solve numerical optimization problems. Xie, et al. (2016) proposed a job scheduling algorithm (SFLA) based on particle swarm optimization (PSO) and shuffled frog leaping algorithm. Sun (2013) used an improved ABC algorithm to identify structural systems. Liu, etal. (2018). proposed an ABC algorithm for constrained optimization problems. Wang, et al. (2018) proposed ABC algorithm with multi-search strategy cooperative evolutionary. Kanagaraj, et al. (2014) employed hybrid CS and GA to sloved constrained engineering design optimization. Baykasoğlu (2015) proposed an adaptive (search mechanism and parameter settings) firefly algorithm to solve mechanical design optimization problems. Kohli (2017) introduces the chaotic grey wolf optimization algorithm accelerating global convergence speed and application to constrained optimization problems. In addition to these approaches, Garg (2016) present a hybrid PSO-GA approach for solving the constrained optimization problems. Hsieh (2012) hybridized of PSO and ABC algorithm to predict trends in financial distress. Tsai (2014) combines ABC and bee algorithm to solve the constrained optimization problem. Kanagaraj, et al. (2014) presents an effective hybrid CS and GA for solving engineering design optimization problems. Kıran, et al. (2012) design a novel hybrid algorithm based on PSO and ACO to finding optimal minimum. Lynn and Suganthan (2017) proposed a comprehensive review of population topologies developed for PSO and DE.

As demonstrated in the above literature, the existing research approaches have been successfully applied to various constrained optimization problems. Therefore, in this study, genetic algorithm and global-best-guided artificial bee colony algorithm are combined to solve nonlinear design optimization problems, and a hybrid algorithm named GABC-GA is proposed. In this method, ABC runs in the direction of improving vectors, and genetic operators have used genetic algorithms to modify decision vectors (Garg, 2016). The rest of the main content of this article is described below. Section 2 briefly introduces the algorithm that will be used (GA and ABC) in this paper. Section 3 introduces our designed hybrid algorithm and constraint processing method. In Section 4 we present the design optimization problems to be dealt with, the experimental results and comparison results. The specific conclusions are given in section 5.

2 OVERVIEW OF THE RELATED ALGORITHM

2.1 Brief introduction to Genetic algorithm

GENETIC algorithm is a adaptive stochastic search algorithm invented by Holland (1975) and based on the survival evolutionary genetics and natural selection. GAs has been widely applied in engineering optimization fields. In most GAs, the model begins with the solution space represented by the initial chromosome population, and the fitness value determines the solution to be good or bad, while using the mutation, crossover and selection methods to obtain a new generation of chromosomes. As the generation increases, all the quality of the chromosomes will increase, and the best generation of the last generation will be recorded as the final solution. The pseudo-code of GA is described in Algorithm 1:

2.2 The Basic artificial bee colony algorithm

Basic artificial bee colony algorithm is a nature inspired swarm intelligence algorithm that simulates the forging behaviors of honey bee swarms. In ABC algorithm, the search process is divided into employed bee stage, onlooker bee stage and scout bee stage. The detailed steps of the basic ABC algorithm are as follows.

Step 1. Set parameter.

The main parameters setting as: the maximum number of iterations (M_{iter}), the size of population (SN)(the sum of numbers of employed and onlooker bees) , the total number of bees (N), D is the problem dimension, the limit parameter ($Limit$) (determine whether the solution needs to be replaced).

Step 2.Initialization

Initial solution population is randomly generated using equation (3), and the fitness value is calculated.

$$x_{ij} = x_{\min j} + \psi_{ij}(x_{\max j} - x_{\min j}) \tag{3}$$

where $i = 1, 2, \cdots, SN$, $j = 1, 2, \cdots, D$, $x_{\min j}$ and $x_{\max j}$ present the bonds of the j-th dimension, and ψ_{ij} is a random number.

Step 3.Employed bee stage

Employed bees generate new candidate solutions by searching for neighbors of food sources. Then calculate fitness value and update the current solution apply greedy selection strategy. The candidate solution can be generated by the follow formula:

$$\overline{x}_{ij} = x_{ij} + \phi_{ij}(x_{ij} - x_{kj}) \tag{4}$$

where $k \in 1, 2, \cdots, SN(k \neq i)$, is a randomly number from SN and ϕ_{ij} is a random number between $[-1,1]$. The parameter set to boundaries by

$$\overline{x}_{ij} = \begin{cases} x_{\min j} & if \quad \overline{x}_{ij} < x_{\min j} \\ x_{\max j} & f \quad \overline{x}_{ij} > x_{\max j} \end{cases} \tag{5}$$

Step 4. Onlooker bee stage

The solution is chosen probabilistically based on its fitness value.

Onlooker bees use equation (4) to search their neighborhood to improve the current solution and the solution is chosen probabilistically according to its fitness value.

$$p_i = \frac{fitness_i}{\sum_{j=1}^{SN} fitness_j} \tag{6}$$

Greedy selection mechanism is applied to update the current solution after a new solution is generated.

Step 5. Scout bee stage

If the food source is not updated after a certain cycle, it is replaced by the new solution randomly generated by Equation (7).

$$x_{newj} = x_{\min j} + \psi_{newj}(x_{\max j} - x_{\min j}) \tag{7}$$

Step 6. Repeat the search process

If the termination condition is not met, repeat above steps, otherwise the algorithm terminates and outputs the result.

2.3 Improved global-best-guided artificial bee colony algorithms

ABC algorithm has good exploration capabilities but weak exploitation capabilities, many literatures have made corresponding research and proposed some modified ABC algorithms. Hsieh (2012) combined together the ABC and PSO to improve the search ability. Ouyang (2017) added a global -best-guided term to search equation to enhance the exploitation performance. Zhong (2016) inspired by the methods of the global-best-guided term proposed a modified ABC algorithm strategy; experiments show that the algorithm has good performance.

Learning from the above algorithms, a modified ABC algorithm based on improved global-best-guided is used in our approach, called GABC. For the reason that, the equation of candidate solution in employed bee stage is modified as following (Zhong, 2016):

$$\overline{x}_{ij} = x_{ij} + \chi_{iter}\varphi_{ij}(g_j - x_{ij}) + \lambda_{iter}\phi_{ij}(x_{ij} - x_{kj}) \tag{8}$$

$$\chi_{iter} = 1 - (\frac{iter}{M_{iter} + \varepsilon_1}) \tag{9}$$

$$\lambda_{iter} = 1 - (\frac{iter}{M_{iter} + \varepsilon_2}) \tag{10}$$

where $\varphi_{ij} \in [0, L]$, L is a non-negative number, $\phi_{ij} \in [-1,1]$, χ_{iter} and λ_{iter} are nonlinear adjusting factors and decrease with iteration increases, $iter$ denotes the current iteration, M_{iter} is the maximum number of iterations, ε_1 and ε_2 are positive number and less than or equal to maximum number of iterations.

Algorithm 1 Pseudo code of the GA algorithm.

Initialize population chromosomes randomly
Objective function:
Evaluate fitness value for all chromosomes
Initial probabilities of crossover and mutation
do
Update chromosomes by crossover and mutation operations
If $p_c > rand$, crossover operation; end if
If $p_m > rand$, mutation operation; end if
Accept the new chromosomes if its fitness increases
Select the best found chromosomes for the next generation
While maximum iterations or minimum error criteria is not met

From equation (8) we can see that when $L=0$, $\lambda_{iter}=1$, the equation becomes the initial one (4). L control the exploitation ability in the search process, it cannot be set too large, otherwise the best solution will be missed. χ_{iter} can adaptively adjusting to balance the ability of exploration and exploitation. $\lambda_{iter}\phi_{ij}$ ensures the self-regulation of randomly generated solutions.

3 THE PROPOSED APPRAACH

3.1 Hybrid GABC-GA algorithm

THE hybrid GABC-GA approach is a improve algorithm, which can find global optimal solutions by repeatedly iterations just like ABC and GA. Therefore, the proposed algorithm begins with an initialization phase in which the initial solutions of the population are randomly generated in the search space using equation (3) and evaluate fitness value of each solution. In employed bee stage, use (8) to generate a new candidate solution and use (5) to limit it to the search space. Through greedy selection method, better solution is chosen, and use (6) to calculate the probability that this solution will be selected by the onlooker bee. In the onlooker bee phase, onlooker chooses one to follow and use equation (8) to search its neighborhood to improve the current solution, calculate the adaptability of the new candidate solution, and retain a solution with higher fitness. During the scout bee phase, if the qualities of the current solution do not improved, it is replaced by a new solution generated arbitrarily by (7). After running the GABC algorithm to generate a new generation, and then GA is applied to each one at this time. Because of this large population size, how to determine the number of solutions in an GABC renewal generation? With regards to this, Harish Garg (2016) uses a evolution method in hybrid PSO-GA algorithms, i.e. in the total population size, the evolution number selected in each PSO generation is effected by GA. Here, we applied its idea to deal with our approach, evolved in each GABC generation the number of GA is defined by GA_N and defined as follow.

$$GA_N = GA_{MaxN} - \left(\frac{GABC_i}{GABC_{MaxI}}\right)^{\alpha} \times (GA_{MaxN} - GA_{MinN}) \tag{11}$$

After choosing the best solution from the population, we applied the selection, crossover, and mutation operators to update the solutions. Population size and maximum iteration numbers of GA changes with the iteration of GABC are set by Garg (2016) as follow equation.

$$GA_{PopS} = GA_{MinPopS} - \left(\frac{GABC_i}{GABC_{MaxI}}\right)^{\alpha} \times (GA_{MaxPopS} - GA_{MinPopS}) \tag{12}$$

$$GA_{MaxI} = GA_{MimI} - \left(\frac{GABC_i}{GABC_{MaxI}}\right)^{\beta} \times (GA_{MaxI} - GA_{MinI}) \tag{13}$$

where GA_N is the current number of individuals, GA_{PopS} is the population size, GA_{MinN} and GA_{MaxN} is the minimum and maximum number of solutions, $GA_{MinPopS}$ and $GA_{ManPopS}$ is the first and last population size, GA_{MimI} and GA_{MimI} is the minimum and maximum number of iteration, $GABC_i$ is the current iteration number in $GABC$, $GABC_{MaxI}$ is the maximum number of iteration in $GABC$, α represents the decreasing rate of GA individuals, β is the increasing rate of maximum iteration. In Figure 1, we have given the flow chart of the proposed hybrid algorithms.

3.2 Constraint handing approach of COPs

Constraint handing mechanism is usually required in constrained non-linear mathematical programming models, therefore, many different ways have been proposed to handing constraints, of which the most popular one is the penalty function. However, the penalty function has a main drawback is when there are too many parameters to adjust; it is very difficult to find the right combination. To overcome this limitation, the effective method was introduced by Kim et al. (2010) is applied for this study. The function proposed by Kim and used by Baykasoglu (2015) is expressed as:

$$\min_{x\in S} L(x) = \begin{cases} \hat{g}(x) = g_{\max}(x), & if\ g_{\max}(x) > 0 \\ \hat{f}(x) = \alpha\tan[f(x)] - \dfrac{\pi}{2}, & otherwise \end{cases} \tag{14}$$

where $g_{\max}(x) = \max_{g_j(x)}[h_1(x), h_2(x), \cdots, h_q(x)]$ and $\alpha\tan[\cdot]$ denotes the inverse tangent. When $\hat{f}(x) < 0$ for any x, and thus $\hat{f}(x) < \hat{h}(x)$ is guaranteed (Baykasoglu, 2015).

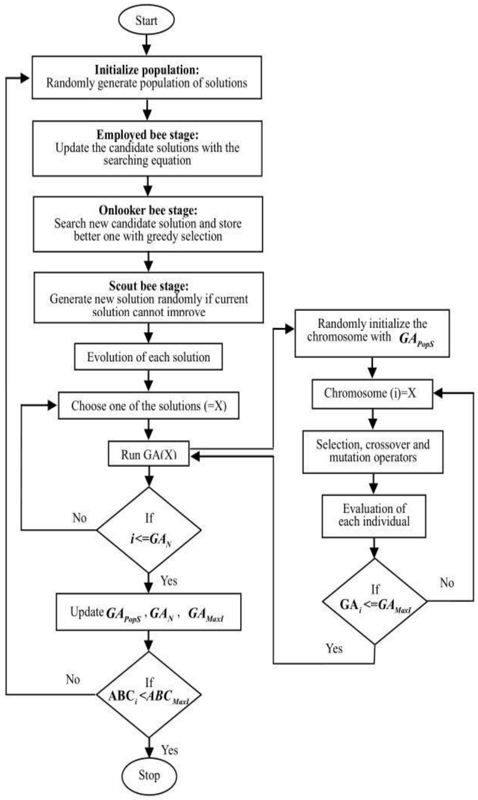

Figure 1 The flow chart of GABC-GA

4 TEST PROBLEMS AND COMPUTATIONAL RESULTS

4.1 Experimental setup

IN order to evaluate the effectiveness and efficiency of the proposed algorithm, we test its performance by selected three mechanical design optimization problems. The tests are conducted with two tests section, Test 1: the results of GABC-GA are compared with basic ABC and GA. The aim of this comparison is to show whether the hybrid approach has improved the overall performance of the method; Test 2: the results of GABC-GA are compared with other algorithms mentioned in the reference. The purpose of doing so is to show the comparison results of the proposed method and other effective methods.

For fair comparisons, 50 independent runs are made and solutions are obtained in maximum number of iterations (1000) or relative error is 10^{-6} . All other parameters of algorithm are setting as $D=20$; $L=1.5$; $crossover\text{-}rate = 0.8$; $mutation\text{-}rate = 0.03$; $GA_{MinPopS}=10$; $GA_{MinN}=1$; $GA_{MinI}=20$; $\alpha=10$; $\beta=15$.

4.2 Mechanical design optimization problems

4.2.1 Test problems 1: pressure vessel design

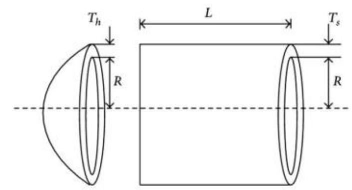

Figure 1 Pressure vessel design problem

The pressure vessel design problem is introduced by Kannan et.al. (1994), as shown in Figure 2. Thickness of shell $T_s(x_1)$, thickness of head $T_h(x_2)$, inner radius $R(x_3)$ and length of the cylindrical section of the vessel $L(x_4)$ four design variables to be consider. T_s and T_h are integer multiples of 0.0625, which are the available thickness of the rolled steel plates; R and L are continuous variables. The mathematical model is described as below.

$$
\begin{aligned}
\min \quad & f(x) = 0.6224x_1x_3x_4 + 1.7781x_2x_3^2 + 3.1661x_1^2x_4 + 19.84x_1^2x_3 \\
\text{s.t.} \quad & g_1(x) = -x_1 + 0.0193x_3 \le 0 \\
& g_2(x) = -x_2 + 0.0095x_3 \le 0 \\
& g_3(x) = -\pi x_3^2x_4 - \frac{4}{3}\pi x_3^3 + 1296000 \le 0 \\
& g_4(x) = x_4 - 240 \le 0 \\
& 0 < x_1, x_2 \le 99 \quad 10 < x_3, x_4 \le 200
\end{aligned}
$$

The results of the proposed algorithm and with other authors have given existing algorithms are presented in Tables 1 and 2, respectively. From Table 1, we can see that the best solution of pressure vessel problem using the design method is x_1=0.81501271, x_2=0.42856436, x_3=42.19546977, x_4=176.63728721 , function value is equal to 6059.71389215 and the standard deviation of the results in 50 independent runs is 0.001783205 . Compared with the results obtained by ABC and GA, it can be seen that GABC-GA is significantly better than the other two methods. Table 2 presents the best solution obtained by different methods published in the literature. The compare results showed that the GABC-GA outperforms most of other state-of-art algorithms.

4.2.2 Test problems 2: tension/compression spring design

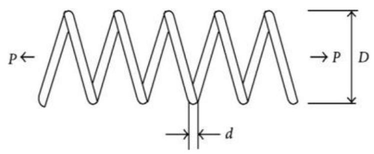

Figure 2 **Tension/compression spring design problem**

Tension/compression spring design problem was first described by Belegundu, as shown in Figure 3. The design variables are the wire diameter $d(x_1)$, the mean coil diameter $D(x_2)$ and the number of active coils $N(x_3)$. The mathematical model is described as below.

$$\min \quad f(x) = (x_3+2)x_2x_1^2$$

$$\text{s.t.} \quad g_1(x) = 1 - \frac{x_2^3x_3}{71785x_1^4} \le 0$$

$$g_2(x) = \frac{4x_2^2 - x_1x_2}{12566(x_2x_1^3 - x_1^4)} + \frac{1}{5108x_1^2} - 1 \le 0$$

$$g_3(x) = 1 - \frac{140.45x_1}{x_2^2x_3} \le 0$$

$$g_4(x) = \frac{x_1+x_2}{1.5} - 1 \le 0$$

$$0.05 < x_1 \le 2, \quad 0.25 < x_2 \le 1.3, \quad 2 < x_3 \le 15$$

In this experiment, the GABC-GA algorithm is used to solve Tension/compression spring problem. The results of the proposed algorithm and with other authors existing algorithms are presented in Tables 3 and 4, respectively. Table 3 showed the best solution is obtained by the proposed approach is x_1=0.0516789124, x_2=0.3567321179, x_3=11.2872132513 with corresponding function value is equal to 0.0126654528 and the standard deviation of the results in 50 independent runs is $1.23167\,\mathrm{e}-06$. Compared with the results obtained by ABC and GA, it can be seen that the present approach is significantly better than the other two methods. Moreover, Table 4 presents the best solution obtained by different methods published in the literature. The compare results showed that the GABC-GA outperforms most of other algorithms so far.

4.2.3 Test problems 3: welded beam design

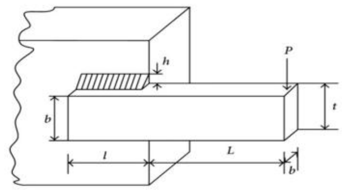

Figure 3 **Welded beam design problem**

Welded beam design problem introduced the optimization of welded beam which is a minimize cost design of the fabrication shown in Figure 4. The design variables to be consider are the thickness of the weld $h(x_1)$, the length of the welded joint $l(x_2)$, the width of the beam $t(x_3)$ and the thickness of the beam $b(x_4)$ .The mathematical programming problem is given as below.

$$\min \quad f(x) = 1.1047x_1^2x_2 + 0.04811x_3x_4(14.0 + x_2)$$

$$\text{s.t.} \quad g_1(x) = \tau(x) - \tau_{\max} \le 0$$

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

$$g_3(x) = x_1 - x_4 \le 0$$

$$g_4(x) = 0.10471x_1^2 + 0.04811x_3x_4(14.0 + x_2) - 5.0 \le 0$$

$$g_5(x) = 0.125 - x_1 \le 0$$

$$g_6(x) = \delta(x) - \delta_{\max} \le 0$$

$$g_7(x) = P - P_C(X) \le 0$$

$$0.1 < x_i \le 10 \quad (i = 1,2,3,4)$$

$$where \quad \tau(x) = \sqrt{(\tau')^2 + (\tau'')^2 + \frac{2\tau'\tau''x_2}{2R}}$$

$$\tau' = \frac{p}{\sqrt{2}x_1x_2} \quad \tau'' = \frac{MR}{J} \quad M = P(L + \frac{x_2}{2})$$

$$R = \sqrt{(\tau')^2 + (\tau'')^2 + (\frac{x_1 + x_3}{2})^2}$$

$$J = 2\{\sqrt{2}x_1x_2[\frac{x_2^2}{4} + (\frac{x_1+x_3}{2})^2]\}$$

$$\sigma(x) = \frac{6PL}{x_4x_3^2} \quad \delta(x) = \frac{4PL^3}{Ex_4x_3^3}$$

$$P_C(x) = \frac{4.013E\sqrt{x_3^2x_4^6/36}}{L^2}(1 - \frac{x_3}{2L}\sqrt{\frac{E}{4G}})$$

In this experiment, the results of the proposed algorithm and with other authors existing algorithms are presented in Tables 5 and 6, respectively. From Table 5, we can see that the best solution is obtained by the proposed approach for Welded beam design problem is x_1=0.2057298, x_2=3.4704891, x_3=9.0366240, x_4=0.2057303, and function value is equal to 1.7248537 and the standard deviation of the

Table 1. The best solution obtained by ABC, GA and GABC-GA for pressure vessel design problem.

Algo.	$x_1(T_s)$	$x_2(T_h)$	$x_3(R)$	$x_4(L)$	$f(best)$	$f(worst)$	$f(avg.)$	$f(std.)$
ABC(Akay and Karaboga)	0.812500	0.4375	42.098446	176.636596	6059.714736	-	6245.308144	2.05e+02
GA(Coello)	0.8125	0.4375	40.097398	176.654047	6059.946341	-	-	-
GABC-GA	0.81501271	0.42856436	42.19546977	176.63728721	6059.71389215	6059.72345316	6059.71389457	0.001783205

Table 2. The best solution obtained by different methods for pressure vessel design problem.

Ref	$x_1(T_s)$	$x_2(T_h)$	$x_3(R)$	$x_4(L)$	$f(best)$	$f(worst)$	$f(avg.)$	$f(std.)$	*Fes*
Aguirre et al.(2007)	0.8125	0.4375	42.098446	176.636596	6059.714335	-	6071.013366	15.101157	350,000
Akay and Karaboga(2012)	0.812500	0.4375	42.098446	176.636596	6059.714736	-	6245.308144	2.05e+02	30,000
Baykasoglu (2012)	0.8125	0.4375	42.09754674	176.64838674	6059.83905683	6823.60245024	6149.72760669	210.77	20,000
Baykasoglu (2015)	0.8125	0.4375	42.09844611	176.63658942	6059.71427196	6090.52614259	6064.33605261	11.28785324	50,000
Brajevic and Tuba (2013)	0.8125	0.4375	42.098446	176.636596	6059.714335	-	6192.116211	204	15,000
Cagnina (2008)	0.8125	0.4375	42.098445	176.690000	6059.714335	-	6092.0498	12.1725	24,000
Chun(2013)	0.8125	0.4375	42.09844560	176.63659584	6059.714335	6090.52620169	6060.33057699	4.35745530	300,000
Coello(2000)	0.812500	0.4375	40.3239	200.0000	6288.7445	-	-	-	900,000
Coello and Montes(2001)	0.8125	0.4375	40.097398	176.654047	6059.946341	-	-	-	80,000
Gandomi et al. (2013)	0.8125	0.4375	42.0984456	176.6365958	6059.7143348	6318.95	6179.13	137.223	20,000
Gandomi et al. (2013)	0.8125	0.4375	42.0984456	176.6365958	6059.714	6495.347	6447.736	502.693	15,000
Garg(2016)	0.7781686	0.3846491	40.3196187	200,0000	5885.332773	5885.486467	5885.382053	0.049080	20,000
He et al. (2004)	0.8125	0.4375	42.0984456	176.63659584	6059.714355	-	-	-	30,000
He and Wang(2007)	0.8125	0.4375	42.091266	176.746500	6061.0777	6363.8041	6147.1332	86.4545	200,000
Kanan and Kramer (1994)	1.123000	0.625000	58.291000	43.690000	7198.0428	-	-	-	20,000
Kaveh and Talatahari (2013)	0.8125	0.4375	42.098353	176.637751	6059.7258	6150.1289	6081.7812	67.2418	-
Kim et al. (2010)	0.8125	0.4375	42.0984456	176.63659584	6059.714355	6060.074434	6059.727721	0.065870503	100,000
Liu et al. (2010)	-	-	-	-	6059.714335	6059.714335	6059.714335	1.0e-10	42,100
Maruta et al. (2009)	0.8125	0.4375	42.0984456	176.63659584	6059.714355	7332.841508	6358.156992	372.71	40,000
Mezura and Coello (2005)	0.8125	0.4375	42.098446	176.636596	6059.7143	-	6379.938037	210	30,000
Montes and Coello (2008)	0.8125	0.4375	42.098087	176.640518	6059.7456	7332.8798	6850.0049	426.0000	30,000
Parsopoulos et al. (2005)	-	-	-	-	6154.7	9387.77	8016.37	745.869	5000
Tomassetti (2010)	0.8125	0.4375	42.098446	176.636596	6059.714337	-	-	-	200,000
Xu et al. (2013)	0.8125	0.4375	42.09844560	176.6365958	6059.714335	6059.7183	6059.7145	0.0007	30,000
Yildiz (2009)	0.8125	0.4375	42.09844559	176.6366	6059.7144	6156.5700	6097.4460	35.7810	30,000
Present study	0.81501271	0.42856436	42.19546977	176.63728721	6059.71389215	6059.72345316	6059.71389457	0.001783205	20,000

Table 3. The best solution obtained by ABC, GA and GABC-GA for tension/compression spring design problem.

Algo.	$x_1(d)$	$x_2(D)$	$x_3(N)$	$f(best)$	$f(worst)$	$f(avg.)$	$f(std.)$
ABC(Akay and Karaboga)	0.051749	0.358179	11.203763	0.01665	-	0.012709	0.012813
GA(Coello)	0.051480	0.351661	11.632201	0.0127047834	0.01282208	0.01276920	3.94e-05
GABC-GA	0.0516789124	0.3567321179	11.2872132513	0.0126654528	0.012665231	0.0126652119	1.23167 e-06

Table 4. The best solution obtained by different methods for tension/compression spring design problem.

Ref	$x_1(d)$	$x_2(D)$	$x_3(N)$	$f(best)$	$f(worst)$	$f(avg.)$	$f(std.)$	*Fes*
Aguirre et al. (2007)	0.05168908	0.35671831	11.28893209	0.012665	-	0.012665	0	350,000
Akay and Karaboga (2012)	0.051749	0.358179	11.203763	0.012665	-	0.012709	0.012813	30,000
Baykasoglu (2012)	0.0516929296	0.3568108568	11.2835059488	0.0126652296	0.0140793687	0.0128750789	0.0002966889	20,000
Baykasoglu (2015)	0.0516674837	0.3561976945	11.3195613646	0.0126653049	0.0128058	0.0126770446	0.0127116883	50,000
Brajevic and Tuba (2013)	0.051691	0.356769	11.285988	0.012665	-	0.012683	0.00000331	15,000
Cagnina (2008)	0.051583	0.354190	11.438675	0.012665	-	0.0131	0.00041	24,000
Coello(2000)	0.051480	0.351661	11.632201	0.0127047834	0.01282208	0.01276920	3.94e-05	900,000
Coello and Becerra(2004)	0.050000	0.317395	14.031795	0.0127210	-	-	-	2500
Gandomi et al. (2013)	0.05169	0.35673	11.2885	0.01266522	0.0168954	0.01350052	0.001420272	20,000
He et al. (2004)	0.05169040	0.35674999	11.28712599	0.0126652812	-	-	-	15,000
He and Wang(2007)	0.051728	0.357644	11.244543	0.0126747	0.012924	0.012730	0.000051985	200,000
Hu et al. (2003)	0.051466369	0.35138394	11.60865920	0.0126661409	-	0.012718975	0.008644	10,000
Kim et al. (2010)	0.0516890615	0.3567177493	11.2889651961	0.0126652328	0.01266523	0.01266523	1.05055e-14	100,000
Liu et al. (2010)	-	-	-	0.012665233	0.012665304	0.012665244	1.2e-08	24,950
Mahdavi et al. (2007)	0.05115483	0.34987116	12.0764321	0.0126706	-	-	-	30,000
Maruta et al. (2009)	0.0516885495	0.3567054307	11.2896874780	0.0126652329	0.01461170	0.01275760	0.000269863	40,000
Mezura and Coello(2005)	0.052836	0.384942	9.807729	0.012689	-	0.013165	0.00039	30,000
Ouyang et al. (2017)	0.05167009418	0.3562614538	11.315774166	0.0126652499	0.012665843	0.012665618	9.69175e-06	100,000
Parsopoulos et al. (2005)	-	-	-	0.013120	0.0503651	0.0229478	0.00720571	5000
Ray and Liew(2003)	0.0521602170	0.3681586950	10.6484422590	0.01671727	0.01671727	0.01292267	0.000592	30,000
Tomassetti(2010)	0.051644	0.355632	11.35304	0.012665	-	-	-	200,000
Xu et al. (2013)	0.051689061	0.356717727	11.28896667	0.012665233	0.01266524	0.01266523	1.5386e-09	30,000
Yildiz(2009)	0.051690402	0.3567500	11.2871200	0.01266527	0.012708	0.012673	6.24e-06	30,000
Present study	0.0516789124	0.3567321179	11.2872132513	0.0126654528	0.012665231	0.0126652119	1.23167 e-06	20,000

Table 5. The best solution obtained by ABC, GA and GABC-GA for welded beam design problem

Algo.	$x_1(h)$	$x_2(l)$	$x_3(t)$	$x_4(b)$	$f(best)$	$f(worst)$	$f(avg.)$	$f(std.)$
ABC(Akay and Karaboga)	0.205730	3.470489	9.036624	0.205730	1.724852	-	1.741913	3.1e-02
GA(Coello)	0.202369	3.544214	9.048210	0.205730	1.728024	1.782143	1.748831	0.012926
GABC-GA	0.2057298	3.4704891	9.0366240	0.2057303	1.7248537	1.7259673	1.7248893	2.65e-09

Table 6. The best solution obtained by different methods for welded beam design problem

Ref	$x_1(h)$	$x_2(l)$	$x_3(t)$	$x_4(b)$	$f(best)$	$f(worst)$	$f(avg.)$	$f(std.)$	*Fes*
Aguirre et al. (2007)	0.205730	3.470489	9.036624	0.205730	1.724852	-	1.724881	1.2e-05	350,000
Akay and Karaboga (2012)	0.205730	3.470489	9.036624	0.205730	1.724852	-	1.741913	3.1e-02	30,000
Baykasoglu (2012)	0.205730	3.470488	9.036624	0.205730	1.724852	1.724852	1.724852	0.000000	20,000
Baykasoglu(2015)	0.205730	3.470489	9.036624	0.205730	1.724852	1.724852	1.724852	0.000000	50,000
Brajevic and Tuba (2013)	0.205730	3.470489	9.036624	0.205730	1.724852	-	1.724853	1.7e-06	15,000
Cagnina (2008)	0.205729	3.470488	9.036624	0.205729	1.724852	-	2.0574	0.2154	32,000
Coello(2000)	0.202369	3.544214	9.048210	0.205730	1.728024	1.782143	1.748831	0.012926	900,000
Coello and Becerra(2004)	0.2057	3.4705	9.0366	0.2057	1.7248523	-	-	-	2500
Coello and Montes(2001)	0.205986	3.471328	9.020224	0.206480	1.728226	-	-	-	80,000
Gandomi et al. (2013)	0.2015	3.562	9.0414	0.2057	1.7312	2.3455793	1.8786560	0.2677989	20,000
Garg(2016)	0.2057296	3.2531200	9.0366239	0.2057296	1.6952471	1.6952471	1.6952471	2.192e-09	20,000
He and Wang(2007)	0.202369	3.544214	9.048210	0.205723	1.728024	1.782134	1.748831	0.012926	200,000
Hu et al. (2003)	0.20573	3.47049	9.03662	0.20573	1.72485	-	1.72485	0	10,000
Kaveh and Talatahari (2010)	0.205700	3.471131	9.036683	0.205731	1.724918	1.729752	1.775961	0.009200	-
Kim et al. (2010)	0.205730	3.470489	9.036624	0.205730	1.724852	1.724852	1.724852	0.000000	50,000
Liu et al. (2010)	-	-	-	-	1.7248531	1.7248811	1.7248579	4.1e-06	33,000
Mahdavi et al. (2007)	0.20573	3.47049	9.03662	0.20573	1.7248	-	-	-	200,000
Maruta et al. (2009)	0.205730	3.470489	9.036624	0.20573	1.724852	1.813471	1.724871	0.0136371	40,000
Mezura and Coello (2005)	0.205730	3.470489	9.036624	0.205730	1.724852	-	1.7776	0.088	30,000
Montes and Coello(2008)	0.199742	3.612060	9.037500	0.206080	1.73730	1.994851	1.813290	0.70500	80,000
Ouyang et al. (2017)	0.205730	3.470489	9.036624	0.205730	1.724852	1.724852	1.724852	4.76378e-09	120,000
Parsopoulos et al. (2005)	0.2407	6.4851	8.2399	0.2497	2.4426	-	-	-	5000
Ray and Liew(2003)	-	-	-	-	2.3854374	6.3996785	3.0025883	0.96	33,095
Tomassetti(2010)	0.205729	3.470489	9.036624	0.205730	1.724852	-	-	-	200,000
Yildiz(2009)	0.205730	3.470489	9.036624	0.205730	1.7248	1.75322	1.73418	0.00510	20,000
Present study	0.2057298	3.4704891	9.0366240	0.2057303	1.7248537	1.7259673	1.7248893	2.65e-09	20,000

results in 50 independent runs is $2.65e-09$. Compared with the results obtained by ABC and GA, it can be seen that GABC-GA is significantly better than the other two methods. Table 6 presents the best solution obtained by different methods published in the literature. The compare results showed that the GABC-GA outperforms the reported results in the most of other literature, which demonstrated the proposed algorithm is more reliable than the other published approach.

5 CONCLUSION

IN this work, we present a new hybrid approach to solve nonlinear design optimization problems. In the proposed approach, it is mainly to combine the advantages of the genetic algorithm and the global optimal guided artificial bee colony algorithm named GABC-GA. Here, the ABC algorithm is responsible for the local search of the problem, while the GA algorithm performs a global search through selection, crossover, and mutation operations; thereby balance the exploration and exploitation ability of the algorithm. From the results of the engineering design constrained optimization problem, the GABC-GA algorithm has superior performance to the ABC and GA algorithms since the proposed algorithm uses different selection operators together: greedy choice, probability selection and random selection, and genetic operation methods. Comparison with other state-of-art approaches, in most cases, the proposed GABC-GA algorithm proves to be effective for constrained optimization problems. The simulation results also show the statistics results for each problem. It can be seen that our approach is recommended for solving constrained optimization problems.

6 ACKNOWLEDGMENT

THE authors would like to thank the editors and anonymous reviewers for their valuable comments and suggestions that have greatly improved the earlier versions of our papers.

This paper is supported by the National Natural Science Foundation of China (61877046). Natural Science Special Plan of school of Huaqing and Xi'an University of Architecture and Technology(17KY01).

7 REFERENCES

Aguirre, A. H., Mu, A. E., Diharce, E. V., & Rionda, S. B. (2007). Copso: constrained optimization via pso algorithm. Center for.

Akay, B., & Karaboga, D. (2012).Artificial bee colony algorithm for large-scale problems and engineering design optimization. Journal of Intelligent Manufacturing, 23(4), 1001-1014.

Altalhi, A. H., Al-Ghamdi, A. M., Ullah, Z., & Saleem, F. (2016). Developing a framework and algorithm for scalability to evaluate the performance and throughput of crm systems. Intelligent Automation & Soft Computing, 23(1), 4.

Ariyasingha, I. D. I. D., & Fernando, T. G. I.. (2017). A modified Pareto strength ant colony optimization algorithm for the multi-objective optimization problems. IEEE International Conference on Information & Automation for Sustainability. IEEE.

Askarzadeh, A. (2016). A novel metaheuristic method for solving constrained engineering optimization problems: crow search algorithm. Computers & Structures, 169, 1-12.

Babu, B. V., & Munawar, S. A. (2007). Differential evolution strategies for optimal design of shell-and-tube heat exchangers. Chemical Engineering Science, 62(14), 3720-3739.

Baykasoglu, A. (2012). Design optimization with chaos embedded great deluge algorithm. Applied Soft Computing, 12(3), 1055-1067.

Baykasoğlu, A., Ozsoydan, F. B., Baykasoğlu, A., Ozsoydan, F. B., Baykasoğlu, A., & Ozsoydan, F. B. (2015).Adaptive firefly algorithm with chaos for mechanical design optimization problems. Applied Soft Computing, 36(C), 152-164.

Babu, B. V., & Munawar, S. A. (2007). Differential evolution strategies for optimal design of shell-and-tube heat exchangers. Chemical Engineering Science, 62(14), 3720-3739.

Brajevic, I., & Tuba, M. (2013). An upgraded artificial bee colony (ABC) algorithm for constrained optimization problems. Journal of Intelligent Manufacturing, 24(4), 729-740.

Cagnina, L. C., Esquivel, S. C., & Coello, C. A. C. (2008).Solving engineering optimization problems with the simple constrained particle swarm optimizer. Informatica, 32(3).

Chun, S., Kim, Y. T., & Kim, T. H. (2013). A diversity-enhanced constrained particle swarm optimizer for mixed integer-discrete-continuous engineering design problems. Advances in Mechanical Engineering, 2013, (2013-5-20), 2013(6), 575-583.

Coello, C.A.C. (2000).Use of a self -adaptive penalty approach for engineering optimization problems, Comput. Ind. 41 113–127.

Coello, C. A. C. (2001). Use of dominance-based tournament selection to handle constraints in genetic algorithms.

Coello, C. A. C., & Becerra, R. L. (2004).Efficient evolutionary optimization through the use of a cultural algorithm. Engineering Optimization, 36(2), 219-236.

Gandomi, A. H., Yang, X. S., & Alavi, A. H. (2013). Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems. Engineering with Computers, 29(2), 245-245.

Gandomi, Hossein, A., Yang, Xin-She, Alavi, &Hossein, A., et al. (2013). Bat algorithm for constrained optimization tasks. Neural Computing & Applications, 22(6), 1239-1255.

Garg, H. (2014). Solving structural engineering design optimization problems using an artificial bee colony algorithm. Journal of Industrial & Management Optimization, 10(3), 777-794.

Garg, H. (2016). A hybrid pso-ga algorithm for constrained optimization problems. Applied Mathematics & Computation, 274(11), 292-305.

He, Q., & Wang, L. (2007). An effective co-evolutionary particle swarm optimization for constrained engineering design problems. Engineering Applications of Artificial Intelligence, 20(1), 89-99.

He, Q., & Wang, L. (2015). A hybrid particle swarm optimization with a feasibility-based rule for constrained optimization. Applied Mathematics & Computation, 186(2), 1407-1422.

Herskovits, J. (1986). A two-stage feasible directions algorithm for nonlinear constrained optimization. Springer-Verlag New York, Inc.

He, S., Prempain, E., Wu, Q.H., (2004). An improved particle swarm optimizer for mechanical design optimization problems, Eng. Optimiz. 36 (5) 585–605.

Hsieh, T. J., Hsiao, H. F., & Yeh, W. C. (2012). Mining financial distress trend data using penalty guided support vector machines based on hybrid of particle swarm optimization and artificial bee colony algorithm. Elsevier Science Publishers B. V.

Hu, X., Eberhart, R. C., & Shi, Y. (2003).Engineering optimization with particle swarm. Swarm Intelligence Symposium, 2003. Sis '03.Proceedings of the (pp.53-57).IEEE.

Jian, J. (1996). A new generalized reduced gradient method for nonlinear inequality constrained optimization problems. Journal of Mathematical Study (4).

Kanagaraj, G., Ponnambalam, S. G., Jawahar, N., &Nilakantan, J. M. (2014). An effective hybrid cuckoo search and genetic algorithm for constrained engineering design optimization. Engineering Optimization, 46(10), 1331-1351.

Kannan, B. K. (1994). An augmented lagrange multiplier based method for mixed integer discrete continuous optimization and its applications to mechanical design. Transactions of Asme Journal of Mechanical Design, 116(2), 405-411.

Kaveh, A., Talatahari, S., (2010). An improved ant colony optimization for constrained engineering design problems, Engineering Computations, 27 1155-182.

Kim, T. H., Maruta, I., &Sugie, T. (2010). A simple and efficient constrained particle swarm optimization and its application to engineering design problems. ARCHIVE Proceedings of the Institution of Mechanical Engineers Part C Journal of Mechanical Engineering Science 1989-1996 (vols 203-210), 224(224), 389-400.

Kıran, M. S., Gündüz, M., & ÖmerKaanBaykan. (2012). A novel hybrid algorithm based on particle swarm and ant colony optimization for finding the global minimum. Applied Mathematics & Computation, 219(4), 1515-1521.

Kohli, M., &Arora, S. (2017).Chaotic grey wolf optimization algorithm for constrained optimization problems. Journal of Computational Design & Engineering.

Liu, F., Sun, Y., Wang, G. G., & Wu, T. (2018). An artificial bee colony algorithm based on dynamic penalty and lévy flight for constrained optimization problems. Arabian Journal for Science & Engineering (1), 1-20.

Luo, J., Wang, Q., & Xiao, X. (2013). A modified artificial bee colony algorithm based on converge-onlookers approach for global optimization. Applied Mathematics & Computation, 219(20), 10253-10262.

Liu, H., Cai, Z., and Wang, Y., (2010). Hybridizing particle swarm optimization with differential evolution for constrained numerical and engineering optimization. Applied Soft Computing, 10 629-640.

Liu, Z., Li, Z. , Zhu, P. , & Chen, W. . (2018). A parallel boundary search particle swarm optimization algorithm for constrained optimization problems. Structural and Multidisciplinary Optimization.

Lynn, N., Ali, M. Z., & Suganthan, P. N. (2017). Population topologies for particle swarm optimization and differential evolution. Swarm & Evolutionary Computation, S2210650217308805.

Mahdavi, M., Fesanghary, M., &Damangir, E. (2007).An improved harmony search algorithm for solving optimization problems. appl math comput. Applied Mathematics & Computation, 188(2), 1567-1579.

Maruta, I., Kim, T. H., &Sugie, T. (2009). Fixed-structure ∞ controller synthesis: a meta-heuristic approach using simple constrained particle swarm optimization. Automatica, 45(2), 553-559.

Mezura-Montes, E., &Coello, C. A. C. (2005).Useful infeasible solutions in engineering optimization with evolutionary algorithms. , 3789, 652-662.

Mezura-Montes, E. & Carlos A. CoelloCoello. (2008). An empirical study about the usefulness of evolution strategies to solve constrained optimization problems. International Journal of General Systems, 37(4), 443-473.

Mirjalili, S., & Lewis, A. (2016).The whale optimization algorithm. Advances in Engineering Software, 95, 51-67.

Ngo, etal. (2017). The Extraordinary Particle Swarm Optimization and Its Application in Constrained Engineering Problems. International Conference on Harmony Search Algorithm. Springer Singapore.

Ouyang, H. B., Gao, L. Q., Li, S., & Kong, X. Y. (2017). Improved global-best-guided particle swarm optimization with learning operation for global optimization problems. Applied Soft Computing, 52(C), 987-1008.

Parsopoulos, K.E., Vrahatis, M.N., (2005). Unified particle swarm optimization for solving constrained engineering optimization problems, in: L. Wang, K. Chen, Y.S. Ong (Eds.), Advances in Natural Computation, Springer, Berlin, pp. 582–591.

Ray, T., & Liew, K. M. (2003). Society and civilization: an optimization algorithm based on the simulation of social behavior. IEEE Transactions on Evolutionary Computation, 7(4), 386-396.

Sandgren, E. (1990). Nonlinear integer and discrete programming in mechanical design optimization. Journal of Mechanical Design, 112(2), 223-229.

Storn, R., & Price, K. (1997).Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces. Journal of Global Optimization, 11(4), 341-359.

Sun, H., Luş, H., &Betti, R. (2013). Identification of structural models using a modified artificial bee colony algorithm. Computers & Structures, 116(1), 59-74.

Tang, C. M. (2012). Strongly sub-feasible direction method for constrained optimization problems with nonsmooth objective functions ☆. European Journal of Operational Research, 218(1), 28-37.

Tian, & Dongping. (2017). Particle swarm optimization with chaos-based initialization for numerical optimization. Intelligent Automation & Soft Computing, 1-12.

Tomassetti, G. (2010). A cost-effective algorithm for the solution of engineering problems with particle swarm optimization. Engineering Optimization, 42(5), 471-495.

Tsai, H. C. (2014). Integrating the artificial bee colony and bees algorithm to face constrained optimization problems. Information Sciences, 258(3), 80-93.

Wang, H. C., & Yang, C. T. (2016). Enhanced particle swarm optimization with self-adaptation based on fitness-weighted acceleration coefficients. Intelligent Automation & Soft Computing, 22(1), 14.

Wang, Z. G., et al. (2018). Artificial bee colony algorithm with multi-search strategy cooperative evolutionary. Control & Decision, 33(2), 235-241.

Xie, X., Liu, R., Cheng, X., Hu, X., & Ni, J. (2016). Trust-driven and pso-sfla based job scheduling algorithm on cloud. Intelligent Automation & Soft Computing, 22(4), 6.

Xu, B. etal. (2018). Adaptive differential evolution with multi-population-based mutation operators for constrained optimization. Soft Computing.

Xu, W., Geng, Z., Zhu, Q., & Gu, X. (2013). A piecewise linear chaotic map and sequential quadratic programming based robust hybrid particle swarm optimization. Information Sciences, 218(1), 85-102.

Xu, X., Liang, T., Wang, G., Wang, M., & Wang, X.. (2016). Self-adaptive pcnn based on the aco algorithm and its application on medical image segmentation. Intelligent Automation and Soft Computing, 23(2), 1-8.

Yang, X., &Gandomi, A. H. (2012). Bat algorithm: a novel approach for global engineering optimization. Engineering Computations, 29(5), 464-483.

Yang, X. S. (2010). Firefly algorithm, stochastic test functions and design optimisation. International Journal of Bio-Inspired Computation, 2(2), 78-84(7).

Yildiz, A. R. (2009). A novel particle swarm optimization approach for product design and manufacturing. International Journal of Advanced Manufacturing Technology, 40(5-6), 617-628.

You, X., He, X., & Han, X. (2017). A novel solution to the cognitive radio decision engine based on improved multi-objective artificial bee colony algorithm and fuzzy reasoning. Intelligent Automation & Soft Computing, 23(4), 1-9.

Yu, X., et al. (2018). Differential evolution mutation operators for constrained multi-objective optimization. Applied Soft Computing, 67, S1568494618301509.

Zhang, R., Song, S., & Wu, C. (2013). A hybrid artificial bee colony algorithm for the job shop scheduling problem. International Journal of Production Economics, 141(1), 167-178.

Zhong, F., Li, H., & Zhong, S. (2016). A modified ABC algorithm based on improved-global-best-guided approach and adaptive-limit strategy for global optimization. Applied Soft Computing, 6(C), 469-486.

8 DISCLOSURE STATEMENT

NO potential conflict of interest was reported by the author.

9 NOTES ON CONTRIBUTORS

Hui Zhi was born in Xinzhou, Shanxi. P.R.China. She received his master's degree in Applied Mathematics from college of science at Xi'an University of Architecture and Technology. His research interest fields include intelligent computing, nonlinear optimization problem, artificial intelligence and big data analysis.

Sangyang Liu is a professor in Applied Mathematics at Xidian University. He received his Ph.D. degree in Applied Mathematics from Xi'an Jiaotong University in Xi'an, P.R. China. His research interests are nonlinear analysis, information network and scientific computing. He has made many outstanding contributions in the research of this professional field.