JOURNAL OF INDUSTRIAL AND
MANAGEMENT OPTIMIZATION
Volume **10**, Number **3**, July **2014**

doi:10.3934/jimo.2014.10.777

pp. **777–794**

SOLVING STRUCTURAL ENGINEERING DESIGN OPTIMIZATION PROBLEMS USING AN ARTIFICIAL BEE COLONY ALGORITHM

Harish Garg

School of Mathematics and Computer Applications
Thapar University Patiala
Patiala - 147004, Punjab, India

(Communicated by Cheng-Chew Lim)

Abstract. The main goal of the present paper is to solve structural engineering design optimization problems with nonlinear resource constraints. Real world problems in engineering domain are generally large scale or nonlinear or constrained optimization problems. Since heuristic methods are powerful than the traditional numerical methods, as they don't requires the derivatives of the functions and provides near to the global solution. Hence, in this article, a penalty guided artificial bee colony (ABC) algorithm is presented to search the optimal solution of the problem in the feasible region of the entire search space. Numerical results of the structural design optimization problems are reported and compared. As shown, the solutions by the proposed approach are all superior to those best solutions by typical approaches in the literature. Also we can say, our results indicate that the proposed approach may yield better solutions to engineering problems than those obtained using current algorithms.

1. **Introduction.** Design optimization can be defined as the process of finding the optimal parameters, which yield maximum or minimum value of an objective function, subject to certain set of specified requirements called constraints. Such problem of optimization is known as constrained optimization problems or nonlinear programming problems. Most design optimization problems in structural engineering are highly nonlinear, involving mixed (discrete and continuous) design variables under complex constraints, which cannot be solved by traditional calculus - based methods and enumerative strategies [35].

In order to solve these type of problems, several heuristic, global optimization as well as meta-heuristic methods exist in the literature. For instance, Ragsdell and Phillips [39] compared optimal results of different optimization methods that are mainly based on mathematical optimization algorithms. These methods are APPROX (Griffith and Stewart's successive linear approximation), DAVID (Davidon-Fletcher-Powell with a penalty function), SIMPLEX (Simplex method with a penalty function), and RANDOM (Richardson's random method) algorithms. Kannan and Kramer [25] combine the augmented Lagrange multiplier

2010 *Mathematics Subject Classification.* Primary: 90C11, 90C30.

Key words and phrases. Artificial bee colony, constraints handling, structural design optimization, nonlinear constraint.

The author would like to thank the Editor-in-Chief and referees for providing very helpful comments and suggestions.

method with Powell's and Fletcher and Reeves Conjugate Gradient method for solving the optimization problems. Sandgren [44] proposed nonlinear branch and bound algorithms based on integer programming to solve the mixed-integer optimization problems. Arora [2] solved the problems by using a numerical optimization technique called a constraint correction at the constraint cost. Although these numerical optimization methods provide a useful strategy to obtain the global optimum (or near to it) for simple and ideal models but they have some disadvantages to handle engineering problems (i.e. complex derivatives, sensitivity to initial values, and the large amount of enumeration memory required). Many real-world engineering optimization problems are highly complex in nature and quite difficult to solve using these methods.

The computational drawbacks of existing numerical methods have forced researchers to rely on heuristic algorithms [32]. Heuristic methods are quite suitable and powerful for obtaining the solution of optimization problems. Although these are approximate methods (i.e. their solution are good, but not provably optimal), they do not require the derivatives of the objective function and constraints. Also, they use probabilistic transition rules instead of deterministic rules. The heuristics technique includes genetic algorithms (GA), simulated annealing (SA), tabu search (TS), particle swarm optimization (PSO), Harmony search (HS), ant colony optimization (ACO) etc. Deb and Goyal [12] presented a combined genetic search technique (GeneAS) which combined binary and real-coded GAs to handle mixed variables. Coello [7], Deb [9], Dimopoulos [13], Hwang and He [24] applied genetic algorithms to solve these mixed-integer engineering design optimization problems. Coelho and Montes [8] proposed a dominance-based selection scheme to incorporate constraints into the fitness function of a genetic algorithm used for global optimization. Tsai [46] proposed a novel method to solve nonlinear fractional programming problems occurring in engineering design and management. Hsu and Liu [22] developed an optimization engine for engineering design optimization problems with monotonicity and implicit constraints. In this, monotonicity of the design variables and activities of the constraints determined by the theory of monotonicity analysis are modeled in the fuzzy proportional-derivative controller optimization engine using generic fuzzy rules.

Montes et al. [37] presented a modified version of the differential evolution algorithm to solve engineering design problems in which a criteria based on feasibility and a diversity mechanism are used to maintain infeasible solution. Zhang et al. [48] proposed an algorithm for constrained optimization problem using differential evolution with dynamic stochastic selection. Omran and Salman [38] presented a new parameter-free meta-heuristic algorithm, named as CODEQ, that is a hybrid of concepts from chaotic search, opposition-based learning, differential evolution and quantum mechanics. Raj et al. [40] presented an efficient real coded evolutionary computational technique which incorporates SA in the selection process of GA for solving mechanical engineering design optimization problems.

He and Whang [17], He et al. [18], Shi and Eberhart[45] applied particle swarm optimization to solve these mixed-integer design optimization problems. Moreover, Coelho [5] presented a quantum-behaved PSO (QPSO) approaches using mutation operator with Gaussian probability distribution while Cagnina et al. [4] introduced a simple constraints particle swarm optimization (SiC-PSO) algorithm to solve constrained engineering optimization problems. Fesanghary et al. [14], Lee and Geem [32] applied harmony search algorithm for these types of problems. Kaveh and

Talatahari [30] developed a hybridized algorithm based on the particle swarm optimization with passive congregation(PSOPC), the ant colony algorithm(ACO), and the harmony search(HS) approach, so-called HPSACO. HPSACO utilizes a PSOPC algorithm as a global search, and the idea of the ACO functions as a local search, and updating the positions of the particles is performed by a pheromone-guided mechanism. The HS-based approach is utilized to handle the boundary constraints. Recently, Gandomi et al. [15, 16] used a cuckoo search and Firefly algorithm for solving mixed continuous/discrete structural optimization problems.

Artificial bee colony (ABC) is one of the meta-heuristic approach proposed by Karaboga in 2005. Because ABCs have the advantages of memory, multi-character, local search and solution improvement mechanism, it is able to discover an excellent optimal solution. Recent studies show that ABC is potentially far more efficient than PSO and GA [26, 27, 28, 29]. In the light of this, the presented paper solve the structural engineering design optimization problems using artificial bee colony and shown that the results are superior to those best solutions by typical approaches in the literature. The rest of the paper is organized as follow: Section 2 deals with the general formulation of optimization problem and method for handling the constraints. In section 3, the artificial bee colony methodology is described along with the local search procedure to improve the solution. The structural design problem are discussed and implemented in Section 4 while conclusions drawn are discussed in Section 5.

2. Engineering optimization problem.

2.1. Structural design optimization problem.

Mechanical design optimization problems can be formulated as a nonlinear programming (NLP) problem. Unlike generic NLP problems which only contain continuous or integer variables, mechanical design optimizations usually involve continuous, binary, discrete and integer variables. The binary variables are usually involved in the formulation of the design problem to select alternative options. The discrete variables are used to represent standardization constraints such as the diameters of standard sized bolts. Integer variables usually occur when the numbers of objects are design variables, such as the number of gear teeth. Considering the mixed variables, the formulation can be expressed as follows:

$$\begin{aligned} \text{Minimize} \quad & f(x) \\ \text{subject to} \quad & h_k(x) = 0 \quad ; \quad k = 1, 2, \ldots, p \\ & g_j(x) \leq 0 \quad ; \quad j = 1, 2, \ldots, q \\ & l_i \leq x_i \leq u_i \quad ; \quad i = 1, 2, \ldots, n \end{aligned} \tag{1}$$

where $x = [x_1, x_2, \ldots, x_n]^T$ denotes the decision solution vectors; f is the objective function; l_i and u_i are the minimum and maximum permissible values for the i^{th} variable respectively; p is the number of equality constraints and q is the number of inequality constraints. Let $S = \{x \mid g_z(x) \leq or = 0, z = 1, 2, \ldots, p{+}q,\ l_i \leq x_i \leq u_i\}$ be the set of feasible solutions and g_z be the set of constraints in the form of equalities and inequalities.

2.2. Constraint handling approach.

The main task while solving the constraint optimization problem is to handle the constraints. In the constrained optimization problem, it is not easy to find the feasible solution of the problem due to the presence of both types of constraints in the form of the equalities and inequalities.

To handle these constraints, many different approaches have been proposed. The most common approach in the EA community is to make use of penalty functions. Despite the popularity of penalty functions, they have several drawbacks out of which the main one is that of having too many parameters to be adjusted and finding the right combination of the same, in order to balance the objective and penalty functions, may not be easy. Also during that the search is very slow and there is no guarantee that the optima will be attained. To overcome this limitation, Deb [10] modified these algorithms using concept of parameter free penalty functions i.e. one attempt to solve an unconstrained problem in a search space S using a modified objective function F such as

$$F(x) = \begin{cases} f(x) & \text{if } x \in S \\ f_w + \sum\limits_{z=1}^{p+q} g_z(x) & \text{if } x \notin S \end{cases} \tag{2}$$

where x are solutions obtained by approaches and f_w is the worst feasible solution in the population.

3. **Artificial bee colony optimization.** The Artificial Bee Colony (ABC) algorithm is a swarm based meta-heuristic algorithm that was introduced by Karaboga in 2005 and its co-authors for optimizing numerical problems [26, 27, 28, 29]. It was inspired by the intelligent foraging behavior of honey bees. In the ABC algorithm, the bees in a colony are divided into three groups: employed bees (forager bees), onlooker bees (observer bees) and scouts. For each food source, there is only one employed bee. That is to say, the number of employed bees is equal to number of food sources. The employed bee of a discarded food site is forced to become a scout for searching new food source randomly. Employed bees share information with the onlooker bees in a hive so that onlooker bee can choose a food source to forager whereas "scouts" are those bees which are currently searching for new food sources in the vicinity of the hive. At the entrance of the hive in an area called the dance floor, the duration of the dance is proportional to the nectar content of the food source currently being exploited by the dancing bee. Onlooker bees which watch numerous dances before choosing a food source tend to choose a food source according to the probability proportional to the quality of that food source. Therefore, the good food sources attract more bees than the bad ones. Whenever a bee, whether it is scout or onlooker, finds a food source it becomes employed. Whenever a food source is exploited fully, all the employed bees associated with it abandon it, and may again become scouts or onlookers. Scout bees can be visualized as performing the job of exploration, whereas employed and onlooker bees can be visualized as performing the job of exploitation.

In this algorithm first stage is the initialization stage in which food source positions are randomly selected by the bees and their nectar amounts (i.e. fitness function) are determined. Then, these bees come into the hive and share the nectar information of the sources with the bees waiting on the dance area within the hive. At the second stage, after sharing the information, every employed bee goes to the food source area visited by her at the previous cycle. Thus the probability p_h of an onlooker bee choose to go the preferred food source at x_h can be defined by $p_h = f_h / \sum\limits_{h=1}^{N} f_h$ where N is the number of food sources and $f_h = f(x_h)$ is the amount of nectar evaluated by its employed bee using eq. (2). After all onlookers have selected their food sources, each of them determines a food source in the neighborhood of his chosen food source and compute its fitness. The best food source among all the neighboring food sources determined by the onlookers associated with a particular food source h itself, will be the new location of the food source h.

After a solution is generated, that solution is improved by using a local search process called greedy selection process carried out by onlooker and employed bees and is given by Eq.(3).

$$v_{hj} = x_{hj} + \phi_{hj}(x_{hj} - x_{kj}) \tag{3}$$

where $k \in \{1, 2, \ldots, N\}$ and $j \in \{1, 2, \ldots, D\}$ are randomly chosen index and D is the number of solutions parameters. Although k is determined randomly, it has to be different from h. ϕ_{hj} is a random number between [-1, 1] and v_h is the solution in the neighborhood of $x_h = (x_{h1}, x_{h2}, \ldots, x_{hD})$. Except for selected parameter j, all other parametric value of v_h are same as that of x_h i.e. $v_h = (x_{h1}, x_{h2}, \ldots, x_{h(j-1)}, v_{hj}, x_{h(j+1)}, ..., x_{hD})$. It controls the production of neighbour food sources around x_{hj} and represents the comparison of two food positions visually by a bee. As can be seen from eq. (3), as the difference between the parameters of the x_{hj} and x_{kj} decreases, the perturbation on the position x_{hj} gets decreased, too. Thus, as the search approaches the optimum solution in the search space, the step length is adaptively reduced. If the resulting value falls outside the acceptable range for parameter j, it is set to the corresponding extreme value in that range.

If a food source is tried/foraged at a given number of explorations without improvement then a new food source will be searched out by its associated bee and it becomes a scout i.e. if position of food source cannot be improved further through a predetermined number of cycles, called "limits", then it is abandoned. Assume that abandonment source is x_h and $j \in \{1, 2, \ldots, D\}$ then the scout discovers a new food source to be replaced with randomly generated food source x_h within its domain $[x_{\min}, x_{\max}]$ as follow:

$$x_{hj} = x_{\min j} + rand(0, 1)(x_{\max,j} - x_{\min,j}) \tag{4}$$

So this randomly generated food source is equally assigned to this scout and changing its status from scout to employed and hence other iteration of ABC algorithm begins until the termination condition is not satisfied. The pseudo code of the ABC algorithm is given in Algorithm 1 and the details are given hereafter.

4. **Numerical examples.** In order to validate the proposed algorithm, several examples taken from the optimization literature will be used to show the way in which the proposed approach works. These examples have linear and nonlinear constraints, and have been previously solved using a variety of other techniques, which is useful to determine the quality of the solutions produced by the proposed approach. The presented algorithm is implemented in Matlab (MathWorks) and the program has been run on a T6400 @ 2GHz Intel Core(TM) 2 Duo processor with 2GB of Random Access Memory(RAM). In order to eliminate stochastic discrepancy, in each example, 30 independent runs are made which involves 30 different initial trial solutions with randomly generated a population/colony size(CS) is set to $20 \times D$, and the maximum iteration or cycle number is 500 in the algorithm for solving the problems. The other parameters for the ABC algorithm is limit $= (CS \times D)/2$ and ABC adopts $5000 \times D$ function evaluations in each run, where D is dimension of the problem. The termination criterion has been set either limited to

Algorithm 1 Pseudo code of the ABC algorithm

1: Objective function: $f(\mathbf{x}), \quad \mathbf{x} = (x_1, x_2, \dots, x_D)$; as given in eq. (2)
2: Generate an initial bee populations (solutions) x_h, $h = 1, 2, \dots, N$ where $x_h = (x_{h1}, x_{h2}, \dots, x_{hD})$ where N is the number of employed bees which are equal to onlooker bees;
3: Evaluate fitness value at each x_h using equation (2) i.e. $f_h = f(x_h)$
4: Initialize cycle=1
5: For each employed bee
   1. Produce new food source position v_{hj} in the neighborhood of x_{hj} using eq. (3).
   2. Evaluate the fitness value at new source v_{hj}
   3. If new position is better than previous position then memorizes the new position.
6: End For.
7: Calculate the probability values $p_h = \frac{f_h}{\sum_{h=1}^{N} f_h}$ of the solution x_h by means of their fitness values using eq. (2).
8: For each onlooker bee
   1. Produce the new populations v_h of the onlookers from the populations x_h, selected by depending on p_h by applying the roulette wheel selection process and evaluate them;
   2. Apply the greedy selection process for the onlookers between x_h and v_h using eq. (3).
   3. If new position is better than previous position, then memorizes the new position.
9: End For
10: If there is any abandoned solution i.e. if employed bee becomes scout then replace its position with a new random source positions
11: Memorize the best solution achieved so far
12: cycle = cycle + 1
13: If termination criterion is satisfied then stop otherwise go to step 5

a maximum number of generations or to the order of relative error equal to 10^{-6}, whichever is achieved first.

4.1. **Constrained optimization problem.**

4.1.1. *Himmelblau's nonlinear optimization problem.* Before solving the structural engineering problems, the ABC was benchmarked using a well-known problem, namely, Himmelblau's problem. This problem has originally been proposed by Himmelblau [20] and it has been widely used as a benchmark nonlinear constrained optimization problem. In this problem, there are five positive design variables $X = [x_1, x_2, x_3, x_4, x_5]$, six nonlinear inequality constraints, and ten boundary conditions. The problem can be stated as follow:

$$\text{Minimize } f(X) = 5.3578547x_3^2 + 0.8356891x_1x_5 + 37.293239x_1 - 40792.141$$

$$
\begin{aligned}
\text{subject to } & 0 \leq g_1(X) \leq 92 \\
& 90 \leq g_2(X) \leq 110 \\
& 20 \leq g_3(X) \leq 25 \\
\text{where } g_1(X) =& 85.334407 + 0.0056858 x_2 x_5 + \mathbf{0.0006262} x_1 x_4 - 0.0022053 x_3 x_5 \\
g_2(X) =& 80.51249 + 0.0071317 x_2 x_5 + 0.0029955 x_1 x_2 - 0.0021813 x_3^2 \\
g_3(X) =& 9.300961 + 0.0047026 x_3 x_5 + 0.0012547 x_1 x_3 + 0.0019085 x_3 x_4 \\
& 78 \leq x_1 \leq 102\ ; \quad 33 \leq x_2 \leq 45\ ; \quad 27 \leq x_3, x_4, x_5 \leq 45
\end{aligned}
$$

This problem was originally proposed by Himmelblau [20], and it has been used before as a benchmark by using several other methods such as GA [10, 21], harmony search algorithm [32], and PSO [18], Cuckoo search [16], simplex search [34] etc. Also, some of the authors [7, 14, 23, 38, 45] have tested their algorithm in another variation of this problem (named as version II), where a parameter 0.0006262 (typeset bold in the constraint g_1) has been taken as 0.00026. The presented algorithm has been tested on both the versions and compares the best solution of the problem with previous best solution reported by them in Table 1. It has been observed from the Table 1 that the solutions gave by [7, 14, 18, 32, 38, 45] are infeasible as they violates the g_1 constraints. The ABC method could obtain the better solution in the variable version I is

$$X = [78.00, 33.00, 29.99516951, 45.00, 36.77574688]$$

with the objective function value -30665.56680546 while in version II, the optimal solution is

$$X = [78.00, 33.00, 27.07097927, 45, 44.96902388]$$

with corresponding function value is -31025.57569195.

Table 1. Optimal results for Himmelblau's nonlinear optimization problem (NA means not available)

Version	Method	Design variables					$f(X)$	Constraints		
		x_1	x_2	x_3	x_4	x_5		$0\leq g_1 \leq 92$	$90\leq g_2 \leq 110$	$20\leq g_3 \leq 25$
I	Himmelblau [20]	NA	NA	NA	NA	NA	-30373.9490	NA	NA	NA
	Homaifar et al. [21]	80.39	35.07	32.05	40.33	33.34	-30005.700	91.65619	99.53690	20.02553
	Deb [10]	NA	NA	NA	NA	NA	-30665.539	NA	NA	NA
	Lee and Geem [32]	78.00	33.00	29.995	45.00	36.776	-30665.500	92.00004[a]	98.84051	19.99994[a]
	He et al. [18]	78.00	33.00	29.995256	45.00	36.7758129	-30665.539	93.28536[a]	100.40478	20.00000
	Dimopoulos [13]	78.00	33.00	29.995256	45.00	36.775813	-30665.54	92.00000	98.84050	20.00000
	Gandomi et al. [16]	78.00	33.00	29.99616	45.00	36.77605	-30665.233	91.99996	98.84067	20.0003
	Mehta and Dasgupta [34]	78.00	33.00	29.995256	45.00	36.775813	-30665.538741	NA	NA	NA
	Present study	78.00	33.00	29.99516951	45.00	36.77574688	-30665.566806	91.999998	98.840473	20.000000
II	Shi and Eberhart [45]	78.00	33.00	27.07099	45.00	44.969	-31025.561	93.28533[a]	100.40473	19.99997[a]
	Coello [6]	78.5958	33.01	27.6460	45.00	45.0000	-30810.359	91.956402	100.54511	20.251919
	Coello [7]	78.0495	33.007	27.081	45.00	44.94	-31020.859	93.28381[a]	100.40786	20.00191
	Hu et al. [23]	78.00	33.00	27.070997	45.00	44.9692425	-31025.5614	92	100.404784	20
	Fesanghary et al. [14]	78.00	33.00	27.085149	45.00	44.925329	-31024.3166	93.27834[a]	100.39612	20.00000
	Omran and Salman [38]	78.00	33.00	27.0709971	45.00	44.9692425	-31025.55626	93.28536[a]	100.40478	20.00000
	Present study	78.00	33.00	27.07097927	45.00	44.96902388	-31025.575692	91.999974	100.404731	20.000002

[a] violate constraints

The best, the average, the worst and the standard deviation of objective function values obtained by 30 runs are reported in Table 2. Based on the above simulation results and comparisons, it can be concluded that ABC is of superior searching quality and robustness for this problem. Moreover, the worst solution obtained by the ABC method is still better than the best one obtained by the other methods. The time elapsed for one execution of the program are 0.958 s and 0.409 s under version I and version II respectively.

Table 2. Statistical results for the Himmelblau's problem (NA means not available)

Version	Method	Best	Median	Mean	Worst	Std
I	Deb [10]	-30665.537	-30665.535	NA	-29846.654	NA
	Lee and Geem [32]	-30665.500	NA	NA	NA	NA
	He et al. [18]	-30665.539	NA	-30643.989	NA	70.043
	Dimopoulos [13]	-30665.54	NA	NA	NA	NA
	Gandomi et al. [16]	-30665.2327	NA	NA	NA	11.6231
	Mehta and Dasgupta [34]	-30665.538741	NA	NA	NA	NA
	Present study	-30665.56680546	-30665.49388961	-30665.40461198	-30664.62469625	0.2383866
II	Coello [7]	-31020.859	-31017.21369099	-30984.24070309	-30792.40773775	73.633536
	Hu et al. [23]	-31025.56142	NA	-31025.56142	NA	0
	Fesanghary et al. [14]	-31024.3166	NA	NA	NA	NA
	Omran and Salman [38]	-31025.55626	NA	-31025.5562644829	NA	NA
	Present study	-31025.57569195	-31025.5612911	-31025.55841263	-31025.49205458	0.0153528

4.2. **Structural optimization problems.**

4.2.1. *Design of pressure vessel.* A compressed air storage tank with a working pressure of 2000psi and a maximum volume of 750ft^3. A cylindrical vessel is capped at both ends by hemispherical heads as shown in Fig. 1. Using rolled steel plate, the shell is made in two halves that are joined by two longitudinal welds to forms a cylinder. The objective is to minimize the total cost, including the cost of material, forming and welding [25]. There are four design variable associated with it namely as thickness of the pressure vessel, $T_s = x_1$, thickness of the head, $T_h = x_2$, inner radius of the vessel, $R = x_3$, and length of the vessel without heads, $L = x_4$ i.e. the variables vectors are given (in inches) by $X = (T_s, T_h, R, L) = (x_1, x_2, x_3, x_4)$. Then, the mathematical model of the problem is summarized as

$$\begin{aligned}
\text{Minimize } f(X) &= 0.6224x_1x_3x_4 + 1.7781x_2x_3^2 + 3.1661x_1^2x_4 + 19.84x_1^2x_3 \\
\text{s.t. } g_1(X) &= -x_1 + 0.0193x_3 \leq 0 \\
g_2(X) &= -x_2 + 0.00954x_3 \leq 0 \\
g_3(X) &= -\pi x_3^2x_4 - \frac{4}{3}\pi x_3^3 + 1296000 \leq 0 \\
g_4(X) &= x_4 - 240 \leq 0
\end{aligned}$$

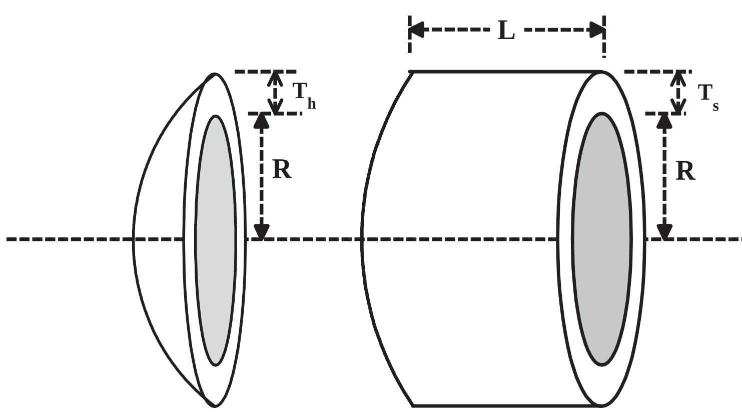

Figure 1. Design of pressure vessel problem

This structural optimization problem has been solved by many researchers within the following variable region:

Region I: $1 \times 0.0625 \leq x_1, x_2 \leq 99 \times 0.0625$; $\quad 10 \leq x_3, x_4 \leq 200$

For instance, the approaches applied to this problem includes genetic adaptive search [11], an augmented Lagrangian multiplier approach [25], a branch and bound technique [44], a GA-based co-evolution model [7], a feasibility-based tournament selection scheme [8], a co-evolutionary particle swarm optimization [17], an evolution strategy [36], improved ant colony optimization [31], hybrid algorithm based on particle swarm optimization with passive congregation [30], cuckoo search [16], quantum behaved PSO [5] etc. The optimal solution obtained by ABC approach in the variable region I is

$$X = (0.778197751897, 0.384665697936, 40.321054550108, 199.980236777701)$$

with corresponding function value equal to

$$f(X) = 5885.403282809389$$

and constraints

$$[g_1, g_2, g_3, g_4] = [-0.0000013991, -0.0000028375, -1.1418297244, -40.0197632223].$$

Table 3. Comparison of the best solution for pressure vessel design problem found by different methods

Region	Method	Design variables				Cost
		x_1	x_2	x_3	x_4	$f(X)$
I	Sandgren [44]	1.125000	0.625000	47.700000	117.701000	8129.1036
	Kannan and Kramer [25]	1.125000	0.625000	58.291000	43.690000	7198.0428
	Deb and Gene [11]	0.937500	0.500000	48.329000	112.67900	6410.3811
	Coello [7]	0.812500	0.437500	40.323900	200.000000	6288.7445
	Coello and Montes [8]	0.812500	0.437500	42.097398	176.654050	6059.946
	He and Wang [17]	0.812500	0.437500	42.091266	176.746500	6061.0777
	Montes and Coello [36]	0.812500	0.437500	42.098087	176.640518	6059.7456
	Kaveh and Talatahari [30]	0.812500	0.437500	42.103566	176.573220	6059.0925
	Kaveh and Talatahari [31]	0.812500	0.437500	42.098353	176.637751	6059.7258
	Zhang and Wang [47]	1.125000	0.625000	58.290000	43.6930000	7197.7000
	Cagnina et al. [4]	0.812500	0.437500	42.098445	176.6365950	6059.714335
	Coelho [5]	0.812500	0.437500	42.098400	176.6372000	6059.7208
	He et al. [18]	0.812500	0.437500	42.098445	176.6365950	6059.7143
	Lee and Geem [32]	1.125000	0.625000	58.278900	43.75490000	7198.433
	Montes et al. [37]	0.812500	0.437500	42.098446	176.6360470	6059.701660
	Hu et al. [23]	0.812500	0.437500	42.098450	176.6366000	6059.131296
	Gandomi et al. [16]	0.812500	0.437500	42.0984456	176.6365958	6059.7143348
	Akay and Karaboga [1]	0.812500	0.437500	42.098446	176.636596	6059.714339
	Present study	0.778197751897	0.384665697936	40.321054550108	199.980236777701	5885.403282809389
II	Dimopoulos [13]	0.75	0.375	38.86010	221.36549	5850.38306
	Mahdavi et al. [33]	0.75	0.375	38.86010	221.36553	5849.76169
	Hedar and Fukushima [19]	0.7683257	0.3797837	39.8096222	207.2255595	5868.764836
	Gandomi et al. [15]	0.75	0.375	38.86010	221.36547	5850.38306
	Present study	0.727595830354	0.359655288904	37.699135991646	239.999805551413	5804.448670820886

The comparison of results are shown in Table 3 under the Region I while their corresponding statistical simulation results are summarized in Table 4. The result obtained by using ABC algorithm are better optimized than any other earlier solutions reported in the literature. It has been notified from the Table 4 that the worst solution found by ABC algorithm is better than any of the solution produced by any of the other techniques.

In the Region I, the bound variable x_4 with an upper bound of 200 has been used to obtain the best solution. By using this domain, the fourth constraints is automatically satisfied. So in order to investigate the whole of the constrained problem domain the upper limit of the variable x_4 have extended to 240 i.e. $10 \leq x_4 \leq 240$ [13] i.e. range of decision variables becomes

$$1 \times 0.0625 \leq x_1, x_2 \leq 99 \times 0.0625 \; ; \quad 10 \leq x_3 \leq 200 \; ; \quad 10 \leq x_4 \leq 240$$

In this region i.e. region II, the researchers Dimopoulos [13], Gandomi et al. [15], Hedar and Fukushima [19], Mahdavi et al. [33] have previously solved the problem by various approaches. Table 3 shows their corresponding best solution vectors along with the best solution obtained from ABC algorithm under Region II. The ABC algorithm again shows better result than other methods. From the statistical simulation results, summarized in Table 4, it can be seen that the average searching quality of ABC is better than those of other methods, and even the worst solution found by ABC is better than the best solutions found by [13, 15, 19, 33]. In addition, the standard deviation of the results by ABC in 30 independent runs is very small. The time elapsed for one execution of the program are 0.575 s and 0.584 s corresponding to a region I and region II.

TABLE 4. Statistical results of different methods for pressure vessel (NA means not available)

Region	Method	Best	Mean	Worst	Std Dev	Median
I	Sandgren [44]	8129.1036	N/A	N/A	N/A	NA
	Kannan and Kramer [25]	7198.0428	N/A	N/A	N/A	NA
	Deb and Gene [11]	6410.3811	N/A	N/A	N/A	NA
	Coello [7]	6288.7445	6293.8432	6308.1497	7.4133	NA
	Coello and Montes [8]	6059.9463	6177.2533	6469.3220	130.9297	NA
	He and Wang [17]	6061.0777	6147.1332	6363.8041	86.4545	NA
	Montes and Coello [36]	6059.7456	6850.0049	7332.8798	426.0000	NA
	Kaveh and Talatahari [31]	6059.7258	6081.7812	6150.1289	67.2418	NA
	Kaveh and Talatahari [30]	6059.0925	6075.2567	6135.3336	41.6825	NA
	Gandomi et al. [16]	6059.714	6447.7360	6495.3470	502.693	NA
	Cagnina et al. [4]	6059.714335	6092.0498	NA	12.1725	NA
	Coelho [5]	6059.7208	6440.3786	7544.4925	448.4711	6257.5943
	He at al. [18]	6059.7143	6289.92881	NA	305.78	NA
	Akay and Karaboga [1]	6059.714339	6245.308144	NA	205	NA
	Present study	5885.403282809389	5887.557024096123	5895.126804460902	2.745290297634486	5886.149289006167
II	Dimopoulos [13]	5850.38306	N/A	N/A	N/A	NA
	Mahdavi et al. [33]	5849.7617	N/A	N/A	N/A	NA
	Hedar and Fukushima [19]	5868.764836	6164.585867	6804.328100	257.473670	NA
	Gandomi et al. [15]	5850.38306	5937.33790	6258.96825	164.54747	NA
	Present study	5804.448670820886	5805.473914033477	5811.977127837280	1.411462164114731	5805.073797973411

4.2.2. *Welded beam design problem.* The welded beam structure, shown in Fig. 2 taken from Rao [41], is a practical design problem that has been often used as a benchmark problem. The objective is to find the minimum fabricating cost of the welded beam subject to constraints on shear stress (τ), bending stress in the beam (θ), buckling load on the bar (P_c), end deflection of the beam (δ), and side constraints. There are four design variables associated with this problem namely, thickness of the weld $h = x_1$, length of the welded joint $l = x_2$, width of the beam $t = x_3$ and thickness of the beam $b = x_4$ i.e. the decision vector is $X = (h, l, t, b) = (x_1, x_2, x_3, x_4)$. For this particular problem, there are several models available in the literature, all of them are vary with respect to number of constraints and the way in which constraints are defined. In the present study, the proposed algorithm has been used on all these versions.

Version I:

The mathematical formulation of the objective function $f(X)$ which is the total fabricating cost mainly comprised of the set-up, welding labor, and material cost,

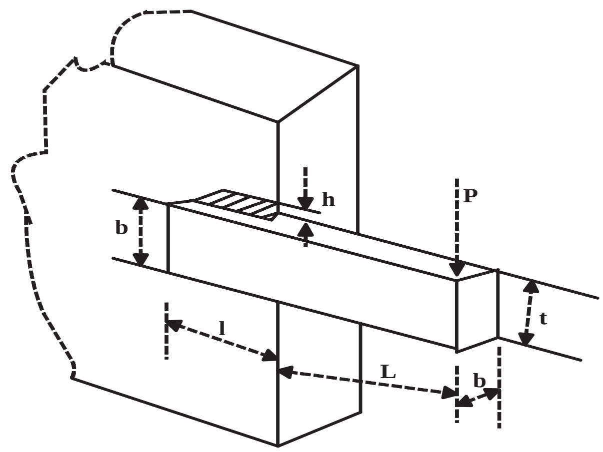

FIGURE 2. Design of Welded beam problem

is as follows:

$$
\begin{aligned}
\text{Minimize } f(X) &= 1.10471x_1^2x_2 + 0.04811x_3x_4(14 + x_2) \\
\text{s.t. } g_1(X) &= \tau(X) - \tau_{\max} \le 0 \\
g_2(X) &= \sigma(X) - \sigma_{\max} \le 0 \\
g_3(X) &= x_1 - x_4 \le 0 \\
g_4(X) &= 0.125 - x_1 \le 0 \\
g_5(X) &= \delta(X) - 0.25 \le 0 \\
g_6(X) &= P - P_c(X) \le 0 \\
0.1 \le x_1 &\le 2\ ; \quad 0.1 \le x_2 \le 10\ ; \quad 0.1 \le x_3 \le 10\ ; \quad 0.1 \le x_4 \le 2
\end{aligned}
$$

where τ is the shear stress in the weld, $\tau_{\max}$ is the allowable shear stress of the weld (= 13600 psi), σ the normal stress in the beam, $\sigma_{\max}$ is the allowable normal stress for the beam material (= 30000 psi), P_c the bar buckling load, P the load (= 6000lb), and δ the beam end deflection.

The shear stress τ has two components namely primary stress (τ_1) and secondary stress (τ_2) given as

$$
\tau(X) = \sqrt{\tau_1^2 + 2\tau_1\tau_2\left(\frac{x_2}{2R}\right) + \tau_2^2} \quad ; \quad \tau_1 = \frac{P}{\sqrt{2}x_1x_2} \quad ; \quad \tau_2 = \frac{MR}{J},
$$

where

$$
M = P\left(L + \frac{x_2}{2}\right) \quad ; \quad J(X) = 2\left\{\frac{x_1x_2}{\sqrt{2}}\left[\frac{x_2^2}{12} + \left(\frac{x_1 + x_3}{2}\right)^2\right]\right\}
$$

are known as moments and polar moment of inertia respectively while the other terms associated with the model are as follows

$$R = \sqrt{\frac{x_2^2}{4} + \left(\frac{x_1 + x_3}{2}\right)^2} \quad ; \quad \sigma(X) = \frac{6PL}{x_4 x_3^2}$$

$$\delta(X) = \frac{4PL^3}{E x_3^3 x_4} \quad ; \quad P_c(X) = \frac{4.013\sqrt{\frac{EGx_3^2x_4^6}{36}}}{L^2}\left(1 - \frac{x_3}{2L}\sqrt{\frac{E}{4G}}\right)$$

$$G = 12 \times 10^6 psi, \quad E = 30 \times 10^6 psi, \quad P = 6000lb, \quad L = 14in$$

Deb [9], Hwang and He [24] have solved this problem using GA - based methods. Ragsdell and Phillips [39] compared optimal results of different optimization methods that are mainly based on mathematical optimization algorithms. These methods are APPROX (Griffith and Stewart's successive linear approximation), DAVID (Davidon - Fletcher - Powell with a penalty function), SIMPLEX (Simplex method with a penalty function), and RANDOM (Richardson random method) algorithms. Lee and Geem [32] and He et al. [18] applied harmony search and particle swarm optimization algorithm respectively for solving this problem.

In this version, the best solutions obtained by the ABC approach are listed in Table 5, under the version I section, along with the best solutions gave by the other authors [9, 18, 24, 32, 34, 39, 42, 48] while the statistical simulation results are summarized in Table 6. It has been clearly seen from the Table 6, after 30 independent runs, that values of the best, mean, worst, standard deviation and the median obtained by ABC are the best when compared with respect to other algorithms. Moreover, the worst solution found is $f(X) = 2.38146999$, which is better than any of the solution produced by any of the other techniques. The time elapsed for one execution of the program is 0.748 s. The best results obtained by ABC is $f(X) = 2.38099617$ corresponding to $X = [x_1, x_2, x_3, x_4] = [0.24436198, 6.21767407, 8.29163558, 0.24436883]$ and constraints $[g_1(X), g_2(X), \ldots, g_6(X)]$ =[-0.10024432, -1.17019903, -0.00000684, -0.11936198, -0.23424176, -0.07175578]

Table 5. Comparison of the best solution for Welded beam found by different methods (NA means not available)

Version	Method	Design variables				$f(X)$
		x_1	x_2	x_3	x_4	
I	Ragsdell and Phillips [39]	0.245500	6.196000	8.273000	0.245500	2.385937
	Rao [41]	0.245500	6.196000	8.273000	0.245500	2.3860
	Deb [9]	0.248900	6.173000	8.178900	0.253300	2.433116
	Deb [10]	NA	NA	NA	NA	2.38119
	Ray and Liew [42]	0.244438276	6.2379672340	8.2885761430	0.2445661820	2.3854347
	Lee and Geem [32]	0.2442	6.2231	8.2915	0.2443	2.38
	Hwang and He [24]	0.223100	1.5815	12.84680	0.2245	2.25
	Mehta and Dasgupta [34]	0.24436895	6.21860635	8.29147256	0.24436895	2.3811341
	Present study	0.24436198	6.21767407	8.29163558	0.24436883	2.38099617
II	Coello [7]	0.208800	3.420500	8.997500	0.210000	1.748309
	Coello and Montes [8]	0.205986	3.471328	9.020224	0.206480	1.728226
	Hu et al. [23]	0.20573	3.47049	9.03662	0.20573	1.72485084
	Hedar and Fukushima [19]	0.205644261	3.472578742	9.03662391	0.2057296	1.7250022
	He and Wang [17]	0.202369	3.544214	9.048210	0.205723	1.728024
	Dimopoulos [13]	0.2015	3.5620	9.041398	0.205706	1.731186
	Mahdavi et al. [33]	0.20573	3.47049	9.03662	0.20573	1.7248
	Montes et al. [37]	0.205730	3.470489	9.036624	0.205730	1.724852
	Montes and Coello [36]	0.199742	3.612060	9.037500	0.206082	1.73730
	Cagnina et al. [4]	0.205729	3.470488	9.036624	0.205729	1.724852
	Fesanghary et al. [14]	0.20572	3.47060	9.03682	0.20572	1.7248
	Kaveh and Talatahari [30]	0.205729	3.469875	9.036805	0.205765	1.724849
	Kaveh and Talatahari [31]	0.205700	3.471131	9.036683	0.205731	1.724918
	Gandomi et al. [15]	0.2015	3.562	9.0414	0.2057	1.73121
	Mehta and Dasgupta [34]	0.20572885	3.47050567	9.03662392	0.20572964	1.724855
	Akay and Karaboga [1]	0.205730	3.470489	9.036624	0.205730	1.724852
	Present study	0.20572450	3.25325369	9.03664438	0.20572999	1.69526388

Version II:

In the literature, several authors [4, 7, 8, 13, 14, 15, 17, 19, 23, 30, 31, 33, 36, 37] have tried their algorithm on the another version of the welded beam design problem. In this version, the constraints namely deflection $\delta(X)$, buckling load $P_c(X)$ and polar moment of inertia $J(X)$ have taken along with another constraint $g_7(X)$ as

$$\begin{aligned} g_7(X) &= 0.10471x_1^2 + 0.04811x_3x_4(14 + x_2) - 5 \leq 0 \\ \delta(X) &= \frac{6PL^3}{Ex_3^3x_4} \quad ; \quad P_c(X) = \frac{4.013E\sqrt{\frac{x_3^2x_4^6}{36}}}{L^2}\left(1 - \frac{x_3}{2L}\sqrt{\frac{E}{4G}}\right) \\ J(X) &= 2\left\{\sqrt{2}x_1x_2\left[\frac{x_2^2}{4} + \left(\frac{x_1 + x_3}{2}\right)^2\right]\right\} \end{aligned}$$

A comparison of the present work with the previous studies is presented in Table 5 under Version II section. From the table, it has been seen that the present algorithm based on ABC performs much better in comparison to other algorithms as optimal function value is lower than the previous studies. The statistically result, after 30 independent runs, in terms of the best, median, mean, worst and the standard deviation obtained for the best objective value by ABC approach are given in Table 6. It shows that mean from the 30 runs performed is $f(X) = 1.69530842$ with a standard deviation of 2.836×10^{-5}. Also the worst solution found in this version is better than any of the solutions produced by any other techniques. The best solution reported by ABC algorithm on this version is $f(X) = 1.69526388$ corresponding to decision variable $X = [0.20572450, 3.25325369, 9.03664438, 0.20572999]$ and constraints $g_1(X), \ldots\ldots, g_7(X)$]=[-0.17975428, -0.18697948, -0.00000549, -3.45240767, -0.08072450, -0.22831066, -0.03957707]. Thus the ABC algorithm provides the best results. The time elapsed for one execution of the program is 0.869 s.

Table 6. Statistical results of different methods for welded beam design problem (NA means not available)

Version	Method	Best	Mean	Worst	Std-dev	Median
I	Ragsdell and Phillips [39]	2.385937	NA	NA	NA	NA
	Rao [41]	2.3860	NA	NA	NA	NA
	Deb [9]	2.433116	NA	NA	NA	NA
	Deb [10]	2.38119	NA	NA	NA	NA
	Ray and Liew [42]	2.3854347	3.2551371	6.3996785	0.9590780	3.0025883
	Lee and Geem [32]	2.38	NA	NA	NA	NA
	Hwang and He [24]	2.25	2.26	2.28	NA	NA
	Mehta and Dasgupta [34]	2.381134	2.3811786	2.3812614	NA	2.3811641
	Present study	2.38099617	2.38108932	2.38146999	1.01227E-4	2.38107233
II	Coello [7]	1.748309	1.771973	1.785835	0.011220	NA
	Coello and Montes [8]	1.728226	1.792654	1.993408	0.07471	NA
	Dimopoulos [13]	1.731186	NA	NA	NA	NA
	He and Wang [17]	1.728024	1.748831	1.782143	0.012926	NA
	Hedar and Fukushima [19]	1.7250022	1.7564428	1.8843960	0.0424175	NA
	Montes et al. [37]	1.724852	1.725	NA	1E-15	NA
	Montes and Coello [36]	1.737300	1.813290	1.994651	0.070500	NA
	Cagnina et al. [4]	1.724852	2.0574	NA	0.2154	NA
	Kaveh and Talatahari [31]	1.724918	1.729752	1.775961	0.009200	NA
	Kaveh and Talatahari [30]	1.724849	1.727564	1.759522	0.008254	NA
	Gandomi et al. [15]	1.7312065	1.8786560	2.3455793	0.2677989	NA
	Mehta and Dasgupta [34]	1.724855	1.724865	1.72489	NA	1.724861
	Akay and Karaboga [1]	1.724852	1.741913	NA	0.031	NA
	Present study	1.69526388	1.69530842	1.69537060	2.836238E-5	1.69530879

4.2.3. *Tension/compression string design problem.* This problem is described by Arora [2] and Belegundu [3] and it consists of minimizing the weight of a tension/compression spring (as shown in Fig. 3) subject to constraints on minimum deflection, shear stress, surge frequency, limits on outside diameter and on design variables. The design variables are the mean coil diameter(x_1), the wire diameter (x_2) and the number of active coil (x_3). The mathematical formulation of this problem can be described as follow:

$$
\begin{aligned}
\text{Minimize } f(X) &= (x_3+2)x_2x_1^2 \\
\text{s.t. } g_1(X) &= 1 - \frac{x_2^3x_3}{71785x_1^4} \leq 0 \\
g_2(X) &= \frac{4x_2^2 - x_1x_2}{12566(x_2x_1^3 - x_1^4)} + \frac{1}{5108x_1^2} - 1 \leq 0 \\
g_3(X) &= 1 - \frac{140.45x_1}{x_2^2x_3} \leq 0 \\
g_4(X) &= \frac{x_1+x_2}{1.5} - 1 \leq 0 \\
0.05 \leq x_1 &\leq 2\ ; \quad 0.25 \leq x_2 \leq 1.3\ ; \quad 2 \leq x_3 \leq 15
\end{aligned}
$$

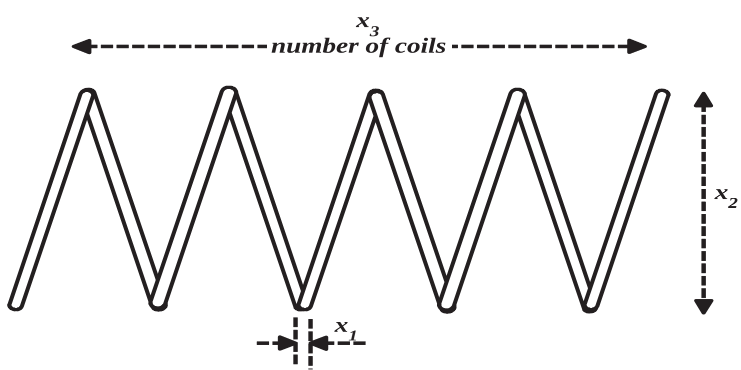

FIGURE 3. Design of the Tension/compression string problem

This problem has been solved by Belegundu [3] using eight different mathematical optimization techniques (only the best results are shown). Arora [2] solved this problem using a numerical optimization technique called a constraint correction at the constant cost. Coello [7] and Coello and Montes [8] solved this problem using GA-based method. Additionally, He and Wang [17] utilized a co-evolutionary particle swarm optimization (CPSO). Montes and Coello [36] used various evolution strategies to solve this problem. Table 7 presents the best solution of this problem obtained using the ABC algorithm and compares with the solutions reported by other researchers, and their correspondingly statistical simulation results are shown in Table 8. The best results obtained by ABC is

$$f(X) = 0.0126652327883$$

corresponding to

$$X = [x_1, x_2, x_3] = [0.051689156131, 0.356720026419, 11.288831695483]$$

and constraints

$$
\begin{aligned}
[g_1(X), \ldots, g_4(X)] \quad = \quad & [-2.5313084961 \times 10^{-13}, -5.7553961596 \times 10^{-13}, \\
& -4.0537846722, -0.7277291363]
\end{aligned}
$$

Table 7. Comparison of the best solution for tension/compression string design problem by different methods

Method	Design variables			$f(X)$
	x_1	x_2	x_3	
Belegundu [3]	0.05	0.315900	14.25000	0.0128334
Arora [2]	0.053396	0.399180	9.185400	0.0127303
Coello [7]	0.051480	0.351661	11.632201	0.01270478
Ray and Saini [43]	0.050417	0.321532	13.979915	0.013060
Coello and Montes [8]	0.051989	0.363965	10.890522	0.0126810
Ray and Liew [42]	0.0521602170	0.368158695	10.6484422590	0.01266924934
Hu et al. [23]	0.051466369	0.351383949	11.60865920	0.0126661409[a]
He et al. [18]	0.05169040	0.35674999	11.28712599	0.0126652812[a]
Hedar and Fukushima [19]	0.05174250340926	0.35800478345599	11.21390736278739	0.012665285
Raj et al. [40]	0.05386200	0.41128365	8.68437980	0.01274840
Tsai [46]	0.05168906	0.3567178	11.28896	0.01266523
Mahdavi et al. [33]	0.05115438	0.34987116	12.0764321	0.0126706
Montes et al. [37]	0.051688	0.356692	11.290483	0.012665
He and Wang [17]	0.051728	0.357644	11.244543	0.0126747
Cagnina et al. [4]	0.051583	0.354190	11.438675	0.012665
Zhang et al. [48]	0.0516890614	0.3567177469	11.2889653382	0.012665233
Montes and Coello [36]	0.051643	0.355360	11.397926	0.012698
Omran and Salman [38]	0.0516837458	0.3565898352	11.2964717107	0.0126652375
Keveh and Talatahari [31]	0.051865	0.361500	11.00000	0.0126432[a]
Coelho [5]	0.051515	0.352529	11.538862	0.012665
Akay and Karaboga [1]	0.051749	0.358179	11.203763	0.012665
Present study	0.051689156131	0.356720026419	11.288831695483	0.0126652327883

[a] infeasible solution as they violate one of the constraint set

From Table 7, it can be seen that the best feasible solution obtained by ABC is better than the best solutions found by other techniques. It has been observed through the calculation that the solutions gave by Hu et al. [23], Kaveh and Talatahari [31] and He et al. [18] are infeasible as they violated one of constraint set. In addition, as shown in Table 8, the average searching quality of ABC is superior to those of other methods. Moreover, the standard deviation of the results by ABC in 30 independent runs for this problem is the smallest. The time elapsed for one execution of the program is 0.463 s.

Table 8. Statistical results of different methods for tension/compression string (NA means not available)

Method	Best	Mean	Worst	Std Dev	Median
Belegundu [3]	0.0128334	NA	NA	NA	NA
Arora [2]	0.0127303	NA	NA	NA	NA
Coello [7]	0.01270478	0.01276920	0.01282208	3.9390×10^{-5}	0.01275576
Ray and Saini [43]	0.0130600	0.015526	0.018992	NA	NA
Coello and Montes [8]	0.0126810	0.012742	0.012973	5.9000×10^{-5}	NA
Ray and Liew [42]	0.01266924934	0.012922669	0.016717272	5.92×10^{-4}	0.012922669
Hu et al. [23]	0.0126661409	0.012718975	NA	6.446×10^{-5}	NA
He et al. [18]	0.0126652812	0.01270233	NA	4.12439×10^{-5}	NA
He and Wang [17]	0.0126747	0.012730	0.012924	5.1985×10^{-5}	NA
Zhang et al. [48]	0.012665233	0.012669366	0.012738262	1.25×10^{-5}	NA
Hedar and Fukushima [19]	0.012665285	0.012665299	0.012665338	2.2×10^{-8}	NA
Montes et al. [37]	0.012665	0.012666	NA	2.0×10^{-6}	NA
Montes and Coello [36]	0.012698	0.013461	0.164850	9.6600×10^{-4}	NA
Cagnina et al. [4]	0.012665	0.0131	NA	4.1×10^{-4}	NA
Kaveh and Talatahari [31]	0.0126432	0.012720	0.012884	3.4888×10^{-5}	NA
Omran and Salman [38]	0.0126652375	0.0126652642	NA	NA	NA
Coelho [5]	0.012665	0.013524	0.017759	0.001268	0.012957
Akay and Karaboga [1]	0.012665	0.012709	NA	0.012813	NA
Present study	0.0126652327883	0.0126689724845	0.012710407729	9.429426×10^{-6}	0.012665314728

5. **Conclusion.** This paper presents the penalty guided artificial bee colony to solve various structural engineering design optimization problems which include pressure vessel design, welded beam design, compression string design. In these optimization problems, the objective is to minimize the cost of the design subject to various nonlinear constraints. To evaluate the performance of ABC algorithm, numerical experiments are conducted and compared to other optimization methods, especially meta-heuristic algorithm-based optimization methods. As demonstrated in the tables, the best solutions found by our ABCs are all better than the well-known best solutions found by other heuristic methods in each problem, i.e. the proposed method achieves the global solution or finds a near-global solution in each problem tested. To demonstrate the effectiveness and robustness of the algorithm compared to other optimization methods, simulations results are also conducted for each problems in terms of mean, median, worst, best and standard deviation. The corresponding results show that the ABC algorithm may yield better solutions than those obtained using other meta-heuristic algorithms. Moreover, the standard deviations of design cost by proposed approach are pretty low, and it further implies that the approach seems reliable to solve the engineering design optimization problems. Thus it is concluded from the analysis the ABC algorithm is a global search algorithm that can be easily applied to various engineering optimization problems.

REFERENCES

[1] B. Akay and D. Karaboga, Artificial bee colony algorithm for large-scale problems and engineering design optimization, *Journal of Intelligent Manufacturing,* **23** (2012), 1001–1014.
[2] J. S. Arora, *Introduction to Optimum Design*, McGraw-Hill, New York, 1989.
[3] A. D. Belegundu, *A Study of Mathematical Programming Methods for Structural Optimization*, PhD thesis, Department of Civil and Environmental Engineering, University of Iowa, Iowa, USA, 1982.
[4] L. C. Cagnina, S. C. Esquivel and C. A. C. Coello, Solving engineering optimization problems with the simple constrained particle swarm optimizer, *Informatica,* **32** (2008), 319–326.
[5] L. S. Coelho, Gaussian quantum-behaved particle swarm optimization approaches for constrained engineering design problems, *Expert Systems with Applications,* **37** (2010), 1676–1683.
[6] C. A. C. Coello, Treating constraints as objectives for single-objective evolutionary optimization, *Engineering Optimization,* **32** (2000), 275–308.
[7] C. A. C. Coello, Use of a self -adaptive penalty approach for engineering optimization problems, *Computers in Industry,* **41** (2000), 113–127.
[8] C. A. C. Coello and E. M. Montes, Constraint- handling in genetic algorithms through the use of dominance-based tournament selection, *Advanced Engineering Informatics,* **16** (2002), 193–203.
[9] K. Deb, Optimal design of a welded beam via genetic algorithms, *AIAA Journal,* **29** (1991), 2013–2015.
[10] K. Deb, An efficient constraint handling method for genetic algorithms, *Computer Methods in Applied Mechanics and Engineering,* **186** (2000), 311–338.
[11] K. Deb and A. S. Gene, A robust optimal design technique for mechanical component design, (Eds. D. Dasgupta, Z. Michalewicz,), *Evolutionary Algorithms in Engineering Applications,* Springer, Berlin, (1997), 497–514.
[12] K. Deb and M. Goyal, A combined genetic adaptive search (GeneAS) for engineering design, *Computer Science and Informatics,* **26** (1986), 30–45.
[13] G. G. Dimopoulos, Mixed-variable engineering optimization based on evolutionary and social metaphors, *Computer Methods in Applied Mechanics and Engineering,* **196** (2007), 803–817.
[14] M. Fesanghary, M. Mahdavi, M. Minary-Jolandan and Y. Alizadeh, Hybridizing harmony search algorithm with sequential quadratic programming for engineering optimization problems, *Computer Methods in Applied Mechanics and Engineering,* **197** (2008), 3080–3091.
[15] A. H. Gandomi, X. S. Yang, and A. H. Alavi, Mixed variable structural optimization using firefly algorithm, *Computers & Structures,* **89** (2011a), 2325–2336.

[16] A. H. Gandomi, X. S. Yang and A. Alavi, Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems, *Engineering with Computers,* **29** (2003), 17–35.
[17] Q. He and L. Wang, An effective co - evolutionary particle swarm optimization for constrained engineering design problems, *Engineering Applications of Artificial Intelligence,* **20** (2007), 89–99.
[18] S. He, E. Prempain and Q. H. Wu, An improved particle swarm optimizer for mechanical design optimization problems, *Engineering Optimization,* **36** (2004), 585–605.
[19] A. R. Hedar and M. Fukushima, Derivative - free filter simulated annealing method for constrained continuous global optimization, *Journal of Global Optimization,* **35** (2006), 521–549.
[20] D. M. Himmelblau, *Applied Nonlinear Programming*, McGraw-Hill, New York, 1972.
[21] A. Homaifar, S. H. Y. Lai and X. Qi, Constrained optimization via genetic algorithms, *Simulation,* **62** (1994), 242–254.
[22] Y. L. Hsu and T. C. Liu, Developing a fuzzy proportional derivative controller optimization engine for engineering design optimization problems, *Engineering Optimization,* **39** (2007), 679–700.
[23] X. H. Hu, R. C. Eberhart and Y. H. Shi, Engineering optimization with particle swarm, *Proceedings of the 2003 IEEE Swarm Intelligence Symposium,* (2003), 53–57.
[24] S. F. Hwang and R. S. He, A hybrid real-parameter genetic algorithm for function optimization, *Advanced Engineering Informatics,* **20** (2006), 7–21.
[25] B. K. Kannan and S. N. Kramer, An augmented lagrange multiplier based method for mixed integer discrete continuous optimization and its applications to mechanical design, *Transactions of the ASME, Journal of Mechanical Design,* **116** (1994), 405–411.
[26] D. Karaboga, *An Idea Based on Honey Bee Swarm for Numerical Optimization*, Technical report, TR06, Erciyes University, Engineering Faculty, Computer Engineering Department 2005.
[27] D. Karaboga and B. Akay, A comparative study of artificial bee colony algorithm, *Applied Mathematics and Computation,* **214** (2009), 108–132.
[28] D. Karaboga, B. Gorkemli, C. Ozturk and N. Karaboga, A comprehensive survey: Artificial bee colony (abc) algorithm and applications, *Artificial Intelligence Review* ,(2012), 1–37.
[29] D. Karaboga and C. Ozturk, A novel clustering approach: Artificial bee colony (ABC) algorithm, *Applied Soft Computing,* **11** (2011), 652–657.
[30] A. Kaveh and S. Talatahari, Engineering optimization with hybrid particle swarm and ant colony optimization, *Asian journal of civil engineering (building and housing),* **10** (2009), 611–628.
[31] A. Kaveh and S. Talatahari, An improved ant colony optimization for constrained engineering design problems, *Engineering Computations,* **27** (2010), 155–182.
[32] K. S. Lee and Z. W. Geem, A new meta-heuristic algorithm for continuous engineering optimization: harmony search theory and practice, *Computer Methods in Applied Mechanics and Engineering,* **194** (2005), 3902–3933.
[33] M. Mahdavi, M. Fesanghary and E. Damangir, An improved harmony search algorithm for solving optimization problems, *Applied Mathematics and Computation,* **188** (2007), 1567–1579.
[34] V. K. Mehta and B. Dasgupta, A constrained optimization algorithm based on the simplex search method, *Engineering Optimization,* **44** (2012), 537–550.
[35] Z. Michalewicz, *Genetic Algorithms + Data Structures = Evolution Programs*, Springer - Verlag, Berlin, 1994.
[36] E. M. Montes and C. A. C. Coello, An empirical study about the usefulness of evolution strategies to solve constrained optimization problems, *International Journal of General Systems,* **37** (2008), 443–473.
[37] E. M. Montes, C. A. C. Coello, J. V. Reyes and L. M. Davila, Multiple trial vectors in differential evolution for engineering design, *Engineering Optimization,* **39** (2007), 567–589.
[38] M. G. H. Omran and A. Salman, Constrained optimization using CODEQ, *Chaos, Solitons & Fractals,* **42** (2009), 662–668.
[39] K. M. Ragsdell and D. T. Phillips, Optimal design of a class of welded structures using geometric programming, *ASME Journal of Engineering for Industries,* **98** (1976), 1021–1025.
[40] K. H. Raj, R. S. Sharma, G. S. Mishra, A. Dua and C. Patvardhan, An evolutionary computational technique for constrained optimisation in engineering design, *Journal of the Institution of Engineers India Part Me Mechanical Engineering Division,* **86** (2005), 121–128.

[41] S. S. Rao, *Engineering Optimization: Theory and Practice,* 3rd edition, John Wiley & Sons, Chichester, 1996.

[42] T. Ray and K. M. Liew, Society and civilization : An optimization algorithm based on the simulation of social behavior, *IEEE Transactions on Evolutionary Computation,* **7** (2003), 386–396.

[43] T. Ray and P. Saini, Engineering design optimization using a swarm with an intelligent information sharing among individuals, *Engineering Optimization,* **33** (2001), 735–748.

[44] E. Sandgren, Nonlinear integer and discrete programming in mechanical design, *Proceedings of the ASME Design Technology Conference,* F.L. Kissimine, 1988, 95–105.

[45] Y. Shi and R. C. Eberhart, A modified particle swarm optimizer, *IEEE International Conference on Evolutionary Computation,* Piscataway, NJ: IEEE Press, (1998), 69–73.

[46] J. Tsai, Global optimization of nonlinear fractional programming problems in engineering design, *Engineering Optimization,* **37** (2005), 399–409.

[47] C. Zhang and H. P. Wang, Mixed-discrete nonlinear optimization with simulated annealing, *Engineering Optimization,* **21** (1993), 277–291.

[48] M. Zhang, W. Luo and X. Wang, Differential evolution with dynamic stochastic selection for constrained optimization, *Information Sciences,* **178** (2008), 3043–3074.

Received June 2012; 1st revision November 2012; 2nd revision June 2013.

E-mail address: harishg58iitr@gmail.com