Adjusted artificial bee colony (ABC) algorithm for engineering problems

Milan TUBA
Faculty of Computer Science
University Megatrend Belgrade
Bulevar umetnosti 29
SERBIA
tuba@ieee.org

Nebojsa BACANIN
Faculty of Computer Science
University Megatrend Belgrade
Bulevar umetnosti 29
SERBIA
nbacanin@megatrend.edu.rs

Nadezda STANAREVIC
Faculty of Mathematics
University of Belgrade
Studentski trg 16
SERBIA
srna@stanarevic.com

Abstract: In this paper we present a modified algorithm which integrates artificial bee colony (ABC) algorithm with adaptive guidance adjusted for constrained engineering optimization problems. The novel algorithm improves best found solutions in some cases and improves robustness i.e. mean value and variance for number of runs in other cases by improving the algorithm's exploitation/exploration balance. Even though scout bee phase is used for exploration, we introduced adaptive parameter that at different stages of the algorithm narrows search space facilitating faster convergence. We tested our algorithm on four standard engineering benchmark problems. The experimental results show that our modified algorithm can outperform the pure ABC algorithm in most cases.

Key-Words: - Artificial bee colony (ABC), Constrained optimization, Swarm intelligence, Metaheuristic optimization

1 Introduction

Different mathematical programming techniques like linear programming, method of feasible direction, dynamic programming and geometric programming have been used to solve hard optimization problems. These methods do not reach satisfactory results on wide range of optimization problems. Numerous constraints make optimization problems complicated and hence these techniques are not ideal for solving such problems as they tend to converge to a local optimal solution. Nature inspired algorithms have been gaining much popularity in recent years due to the fact that many real-world optimization problems have become increasingly large, complex and dynamic. In many situations development of an exact and good performance algorithm cannot be guaranteed. The size and complexity of the problems nowadays require the development of methods and solutions whose efficiency is measured by their ability to find acceptable results within a reasonable amount of time, rather than an ability to guarantee the optimal solution [1]. These methods use the fitness information instead of the functional derivatives making them more robust and effective.

A branch of nature inspired algorithms which are called swarm intelligence is focused on insect behavior in order to develop some meta-heuristics which can mimic insect's problem solution abilities [2], [3], [4], [5]. Formally, a swarm can be defined as a group of agents which communicate with each other either directly or indirectly. Within these groups, individuals are not aware of the global behavior of the group. Swarm intelligence is a heuristic method that models the population of entities that are able to self-organize and interact among them [6], [7].

The following structure of nature systems can be converted into an appropriate mathematical model to solve complex problems of finding the optimal solution [8]:

1. System contains large numbers of relatively simple participants.
2. System is completely decentralized.
3. System operates in parallel and asynchronously.
4. System uses relatively simple signals.
5. System's desired functionality emerges from the interactions of their participants.

Optimization algorithms are capable of finding optimal solutions for numerous test problems for which exact and analytical methods do not produce

This research is supported by Ministry of Science, Republic of Serbia, Project No. III-44006

optimal solutions within a reasonable computational time. Their ability to provide many near-optimal solutions at the end of an optimization run enables to choose the best solution according to given criteria.

The artificial bee colony (ABC) algorithm is a metaheuristic optimization technique that mimics the process of food foraging of honeybees. Originally the ABC algorithm was developed for continuous function optimization problems, but it can also be successfully applied to various other optimization problems.

A majority of industrial engineering optimization problems are constrained problems. The presence of constraints significantly affects the performance of any optimization algorithm. Michalewicz and Fogel [9] describe the following characteristics that make it difficult to solve an optimization problem in the real world:

1. The number of possible solutions (search space) is too large.
2. The problem is so complicated that, with the aim of obtaining a solution, simplified models of the same problem must be used. Thus, the solution may not be useful.
3. The evaluation function that describes the quality of each solution in the search space varies over time or it has noise.
4. Possible solutions are highly restricted, making it difficult even generating at least one feasible solution (i.e., satisfy the constraints of the problem).

The constrained optimization problem can be represented as the following nonlinear programming problem [10], [11]:

$$\text{minimize } f(x),\ x=(x_1,\ \dots,\ x_n)\ \in R^n \tag{1}$$

where $x \in F \subset S$. The objective function f is defined on the search space $S \in R^n$ and the set $F \subset S$ defines the feasible region. Usually, the search space S is defined as an n-dimensional rectangle in R^n (domains of variables defined by their lower and upper bounds):

$$lb_i \le x_i \le ub_i, \quad 1 \le i \le n \tag{2}$$

the feasible region $F \subset S$ is defined by a set of m additional constraints:

$$g_j(x) \le 0, \text{ for } j = 1, \dots, q$$

$$h_j(x) = 0, \text{ for } j = q + 1, \dots, m. \tag{3}$$

Although the original ABC algorithm is a well-performing optimization algorithm, we have noticed that the solution search method of the ABC algorithm can be improved by better guided exploration. In order to improve the exploration phase we decided to use the information of the global best solution and the current best solution in the process of producing new candidate solution in the scout phase rather than random approach. It should be pointed out that the use of global best solution has also been utilized by DE and PSO algorithms [12], [13].

In this paper, we present enhancements of the artificial bee colony algorithm proposed by Karaboga and Basturk [14]. The organization of the remaining paper is as follows. Section 2 details the original ABC algorithm; Section 3 describes the basic theory of the constrained optimization. In Section 4 our modification is proposed and explained in detail. In Section 5 well-known constrained engineering problems are discussed and in Section 6 comparison experiments on the engineering optimization problems are performed to verify efficiency of our proposed approach over the traditional ABC algorithm. Our conclusions and future work are contained in the final Section 7.

2 Pure ABC algorithm

The foraging behavior, learning, memorizing and information sharing characteristics of honeybees have recently been one of the most interesting research areas in swarm intelligence. In the ABC algorithm, the colony of artificial bees contains three groups of bees: employed bees, onlookers and scouts. The number of employed bees is equal to the number of food sources and an employed bee is assigned to one of the sources. In the ABC algorithm, while onlookers and employed bees carry out the exploitation process in the search space, the scouts control the exploration process. The scouts are characterized by low search costs and a low average in food source quality [14]. In the ABC algorithm, the position of a food source represents a possible solution to the optimization problem and the nectar amount of a food source corresponds to the quality (fitness) of the associated solution. An important difference between ABC and other swarm intelligence algorithms is that in the ABC, the solutions of the problem are represented by the food sources, not by the bees. The food source which is abandoned by the bees is replaced with a new food source by the scouts which involves calculating a new solution at random. The employed bee of an abandoned food source becomes a scout. An onlooker bee chooses a food source depending on the probability value associated with that food source, p_i, calculated by the following expression

$$p_i = \frac{fit_i}{\sum_{n=1}^{SN} fit_n} \quad (4)$$

where fit_i is the fitness value of the solution *i* which is proportional to the nectar amount of the food source in the position *i*.

In order to produce a candidate food solution from the old one in memory, the ABC uses the following expression

$$\upsilon_{i,j} = x_{i,j} + \phi_{i,j}(x_{i,j} - x_{k,j}) \quad (5)$$

where $k \in \{1, 2,.., SN\}$ and $j \in \{1, 2,...,D\}$ are randomly chosen indexes. If a solution cannot be improved further through a predetermined number of cycles, the food source will be abandoned. The value of predetermined number of cycles, *limit*, is an important control parameter of the ABC algorithm [14]. There are three main control parameters used in the ABC: the number of food sources which is equal to the number of employed or onlooker bees (*SN*), the value of *limit*, and the maximum cycle number. There are software systems for ABC algorithm [15], as well as paralelized versions [16].

3 Constrained optimization problems

Constrained optimization problems are encountered in numerous applications. The complexity of the constrained optimization problems steams from the fact that desired solution must satisfy all the constraints. During the initialization phase most of the metaheuristic methods start with solutions that can be outside of the feasible area and it is expected that through the algorithm iterations solutions reach the feasible area. The handling of equality constraints can be considered as one of the main difficult issues in solving constrained optimization problems; their existence makes the feasible space very small compared to the entire search space [10]. To expand the feasible area and provide larger number of eligible solutions in the initial stage of the algorithm, the equality constraints are usually transformed to inequality constraints

$$|h(x)| - \varepsilon \le 0 \quad (6)$$

for some small violation value ε>0 [17]. The consequences of choosing too small or too large tolerance value ε lead to poor performance of the algorithm; the results may be too far from the feasible region.

Hamida and Schoenauer have proposed a strategy for handling equality constraints [18] which combines dynamic adjustment and adaptive adjustment. The idea is to start with a large violation value ε, which provides the exploration of the whole search space S. The ε value is then gradually shrunk along iterations, to approach the optimum region. These adjustments might not provide solutions with high quality; hence some solutions can be slightly infeasible due to ε tolerance .

Parameter ε can be defined as follows:

$$\varepsilon(t+1) = \frac{\varepsilon(t)}{dec} \quad (7)$$

where *t* is the current iteration and *dec* is the decreasing rate value of each iteration [19], [20]. Value of the parameter *dec* must be greater than 1. The idea is to start with a larger search space than the original one, and through the algorithm iterations the tolerance will be reduced with each iteration, thus the constraint violation of current solutions will be lower than those of solutions calculated in the previous iteration [21], [22].

The first proposal to extend the ABC algorithm [14] to constrained problems used a constraint handling technique originally proposed for a genetic algorithm by Deb [23], [24]. Penalty function method is the most common approach in handling constraints. By adding a penalty term to the objective function, a constrained optimization problem is transformed into an unconstrained one. Based on the penalty function method, Deb has developed a constraint handling approach which does not require any penalty parameter. Deb's method uses a tournament selection operator, where two solutions are compared at a time, and the following criteria are always enforced:

1. Any feasible solution is preferred to any infeasible solution,
2. Among two feasible solutions, the one having better objective function value is preferred,
3. Among two infeasible solutions, the one having smaller constraint violation is preferred.

In order to adapt the ABC algorithm Karaboga has accepted Deb's constrained handling method instead of the selection process (greedy selection) of the ABC algorithm. Pseudo-code [2] for the ABC algorithm for constrained optimization problems is:

1. Initialize the population of solutions
2. Evaluate the population

3. cycle=1
4. repeat
5. Produce new solutions for the employed bees by using Eq. (8) and evaluate them

$$\upsilon_{i,j} = \begin{cases} x_{i,j} + \phi * (x_{i,j} - x_{k,j}), & R_j < MR \\ x_{i,j} & otherwise \end{cases} \tag{8}$$

6. Apply selection process based on Deb's method
7. Calculate the probability values $P_{i,j}$ for the solutions $x_{i,j}$ using fitness of the solutions and the constraint violations (CV) by Eq. (9)

$$pi = \begin{cases} \left(0.5 + \left(\frac{fitness_i}{\sum_{i=1}^{SN} fitness_i}\right) * 0.5\right) & if\ \ solution\ \ is\ \ feasible \\ \left(1 - \frac{CV}{\sum_{i=1}^{SN} CV}\right) * 0.5 & if\ \ solution\ \ is\ \ infeasible \end{cases} \tag{9}$$

where CV is defined by Eq. (10)

$$CV = \sum_{1}^{q} g_j(x) \ + \ \sum_{q+1}^{m} h_j(x) \tag{10}$$

8. For each onlooker bee, produce a new solution υ_i by (4) in the neighborhood of the solution selected depending on p_i and evaluate it
9. Apply selection process between υ_i and x_i based on Deb's method
10. Determine the abandoned solutions by using "limit" parameter *fr* the scout. If it exists, replace it with a new randomly produced solution by step 5

$$x_i^j = x_{\min}^j + rand(0,1) * (x_{\max}^j - x_{\min}^j) \tag{11}$$

11. Memorize the best solution achieved so far
12. cycle = cycle+1
13. until cycle = MCN

Some recent research on modifications to the ABC algorithm for constrained problems is in [25], [26], [27], [28].

4 Adjusted ABC algorithm

Two major components of any metaheuristic algorithms are exploitation and exploration. Exploration means to explore the search space on the global scale, while exploitation focuses on the search in a local region. It is well known that both exploration and exploitation should be well balanced in any population-based optimization algorithm [29], [30].

In the ABC algorithm, the process of replacing abandoned food source is simulated by randomly producing a new solution, as defined by Eq. (5). Also, new solutions in the scout phase of the ABC algorithm are not based on the information of previous solutions or the global best solution. In practice, we noticed that after a certain number of cycles, solutions will approach the optimum value, hence the use of random solution given by Eq. (5) will be a step backwards. Combination of global and local search is one of the main aspects in the research on constrained optimization problems [29]. Inspired by original proposal in SAVPSO [13] and related works [19], [21], [22], [31] we modified the solution search equation by applying the global best solution and limited solution to guide the search of scout in order to improve the exploration. Latest addition to this research are [32], [33]. To handle constraints, in SAVPSO authors adopt their proposed dynamic-objective constraint-handling method.

The following three characteristics of the feasible region, which can be considered as some kind of knowledge about the feasible region, are responsible for the impact on the search behavior of the particles [13]:

1. The position of the feasible region with respect to the search space;
2. The connectivity and the shape of the feasible region;
3. The ratio $|F|/|S|$ of feasible region to the search space.

According to the characteristics above, in SAVPSO, the swarm is manipulated according to the following self-adaptive velocity equations:

$$v_{id}(t+1) = \omega|p_{i'd}(t) - p_{id}(t)| sign(v_{id}(t)) + \tag{12}$$
$$+ r(p_{id}(t) - x_{id}(t)) + (1-r)(p_{gd}(t) - x_{id}(t))$$
$$x_{id}(t+1) = x_{id}(t) + v_{id}(t+1)$$

where $r \in U[0, 1]$, $i' \in intU[1, N]$, ω is a scaling parameter, and $sign(v_{id}(t))$ is the sign of $v_{id}(t)$. The self-adaptive velocity formula consists of three parts. The first part is velocity of the particle. The second part is the "cognitive" part which represents personal thinking of itself - learning from its own flying experience. The third part is the "social" part which represents the collaboration among particles - learning from group flying experience [13].

In the original ABC algorithm in the scout phase a new solution is generated by using random approach, thus it is very difficult to generate a new solution that could be placed in the promising region of the search space. Our modified algorithm uses a different approach based on proposal utilized in [13]. Instead of generating a random solution based on Eq. (5), the scout will generate a new solution depending on the algorithm's stage. In the early stages of the algorithm's progress (less than 75%) a scout will generate a new solution by adding the global experience information ($x_{best,j}$ - *the best global food source*) to Eq. (5). A new solution will be generated by using information about the food source that is abandoned, the best global food source and a randomly chosen food source as stated in Eq. (13):

$$\upsilon_{i,j} = x_{i,j} + \phi*(x_{i,j} - x_{k,j}) + (1-\phi)*(x_{i,j} - x_{best,j}) \quad (13)$$

When algorithm reaches the final stage of its execution (more than 75%), a scout will generate a new solution by adding the limited solution x_{lim} to Eq. (13).

$$\upsilon_{i,j} = x_{i,j} + \phi*(x_{i,j} - x_{\lim,j}) + (1-\phi)*(x_{i,j} - x_{best,j}) \quad (14)$$

The limited solution is calculated as follows. For each x_{limi} where $i \in \{1, 2,...,D\}$ we determine the $minx_j$ and the $maxx_j$ where $j \in \{1, 2,...,SN\}$ as lower and upper bounds in the current cycle for x_{lim}. Then, we generate x_{limi} randomly between new lower and upper bounds for each cycle.

The proposed modification will increase the capabilities of the ABC algorithm to produce new solutions located near the boundaries of the feasible region or if the best solution is feasible in the promising region by choosing direction based on the best global food source. By adding the limited solution to Eq. (13) algorithm performs fine tuning in the global best solution area. This modification does not change the computational complexity of the algorithm.

5 Engineering optimization problems

In the case of the complex engineering problem, the structure of the problem is often unknown. The quality of the certain parameter setting can often only be evaluated by experiments or simulations. In order to study the performance of solving the real-world engineering design problems, the proposed method is applied to 4 well-known constrained engineering problems: Pressure vessel, tension/ compression spring, speed reducer and welded beam. The number of linear and nonlinear inequality constraints of the problems is given in Table 1.

For constrained optimization problems, no single parameter (number of linear, nonlinear, active function, number of variables) is proved to be significant as a major measure of difficulty of the problem.

Problem	LI	NI
Pressure vessel	3	1
Tension/comp. spring	1	3
Speed reducer	4	7
Welded beam	2	5

Table 1: Number of linear and nonlinear inequality constraints

The pressure vessel problem is to design a compressed air storage tank with a working pressure of 3000 psi and a minimum volume of 750 ft^3. A cylindrical vessel (Fig. 1) is capped at both ends by hemispherical heads. Using rolled steel plate, the shell is made in two halves that are joined by the longitudinal welds to form a cylinder. The objective is to minimize the total cost of material, forming and welding of a cylindrical vessel. The four design variables are x_1 (thickness of the shell), x_2 (thickness of the head), x_3 (inner radius R) and x_4 (length of the cylindrical section of the vessel, not including the head). x_1 and x_2 are to be in integral multiples of 0.0625 inch which are the available thicknesses of rolled steel plates. The radius x_3 and the length x_4 are continuous variables.

Problem 1: the pressure vessel problem

min $f(X) = 0.6224x_1x_3x_4 + 1.7781x_2x_3^2 + 3.1661x_1^2 x_4 + 19.84x_1^2x_3$

subject to

$g_1(X) : -x_1 + 0.0193x_3 \leq 0$

$g_2(X) : -x_2 + 0.00954 \leq 0$

$g_3(X) : -\pi x_3^2x_4 - 4/3\pi x_3^3 + 1296000 \leq 0$

$g_4(X) : x_4 - 240 \leq 0$

where $X = (x_1, x_2, x_3, x_4)^T$. The ranges of the design parameters are $0 \leq x_1, x_2 \leq 99$, $10 \leq x_3, x_4 \leq 200$. Best solution: $x^* = (0.8125, 0.4375, 42.098446, 176.636596)$ where $f(x^*) = 6059.714335$.

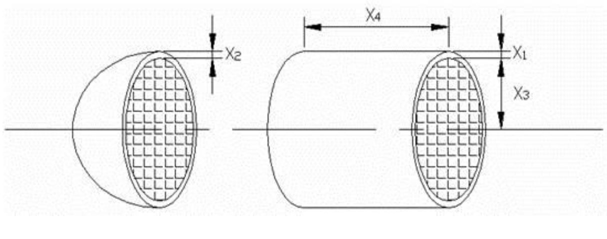

Fig. 1: Pressure vessel design

The tension/compression spring problem deals with minimizing of the weight of the tension/ compression spring subject to constraints on the minimum deflection, shear stress, surge frequency, diameter and design variables. This problem has a nonlinear objective function, a linear and three nonlinear inequality constraints. There are three continuous variables: the wire diameter x_1, the mean coil diameter x_2, and the number of active coils x_3.

Problem 2: The tension/compression spring problem

$$\min f(X) = (N + 2)Dd^2$$

subject to

$$g_1(X) : \frac{1 - D^3N}{71785d^4} \leq 0$$

$$g_2(X) : \frac{4D^2 - Dd}{12566(Dd^3 - d^4)} + \frac{1}{5108d^2} - 1 \leq 0$$

$$g_3(X) : 1 - \frac{140.45d}{ND^2} \leq 0$$

$$g_4(X) : \frac{D+d}{1.5} - 1 \leq 0$$

where $X = (d, D, N)^T$, $0.05 \leq d \leq 2.0$, $0.25 \leq D \leq 1.3$, $2.0 \leq N \leq 15.0$ Best solution is f (x*)= 0.012665, where x*= (0.051690,0.356750,11.287126).

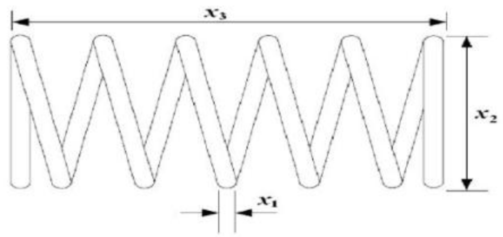

Fig. 2: Tension/compression spring

The aim of the **speed reducer design** is to minimize the weights of the speed reducer subject to constraints on bending stress of the gear teeth, surface stress, transverse deflections of the shafts and stresses in the shafts. The design of the speed reducer, is considered with the face width x_1, module of teeth x_2, number of teeth on pinion x_3, length of the first shaft between bearings x_4, length of the second shaft between bearings x_5, diameter of the first shaft x_6, and diameter of the first shaft x_7. All variables are continuous except x_3 that is integer. Speed reducer problem has seven nonlinear and four linear constraints.

Problem 3: Speed reducer

$$\min f(X) = 0.7854x_1x_2^2(3.3333x_3^2 + 14.9334x_3 - 43.0934) - 1.508x_1(x_6^2 + x_7^2) + 7.4777(x_6^3 + x_7^3)$$

subject to

$$g_1(x): \frac{27}{x_1x_2^2x_3} \leq 0$$

$$g_2(x): \frac{397.5}{x_1x_2^2x_3^2} - 1 \leq 0$$

$$g_3(x): \frac{1.93x_4^3}{x_2x_3x_6^4} - 1 \leq 0$$

$$g_4(x): \frac{1.93x_5^3}{x_2x_3x_7^4} - 1 \leq 0$$

$$g_5(x): \frac{((\frac{745x_4}{x_2x_3})^2 + 16.9*10^6)^{1/2}}{110x_6^3} - 1 \leq 0$$

$$g_6(x): \frac{((\frac{745x_4}{x_2x_3})^2 + 157.5*10^6)^{\frac{1}{2}}}{85x_7^3} - 1 \leq 0$$

$$g_7(x): \frac{x_2x_3}{40} - 1 \leq 0$$

$$g_8(x): \frac{5x_2}{x_1} - 1 \leq 0$$

$$g_9(x): \frac{x_1}{12x_2} - 1 \leq 0$$

$$g_{10}(x): \frac{1.5x_6 + 1.9}{x_4} - 1 \leq 0$$

$$g_{11}(x): \frac{1.1x_7 + 1.9}{x_5} - 1 \leq 0$$

where the bounds are: $2.6 \leq x_1 \leq 3.6$, $0.7 \leq x_2 \leq 0.8$, $17 \leq x_3 \leq 28$, $7.3 \leq x_4 \leq 8.3$, $7.8 \leq x_5 \leq 8.3$, $2.9 \leq x_6 \leq 3.9$, and $5.0 \leq x_7 \leq 5.5$. Best solution is f (x*)= 2996.348165, where x*= (3.5, 0.7, 17, 7.3, 7.8, 3.350214, 5.286683).

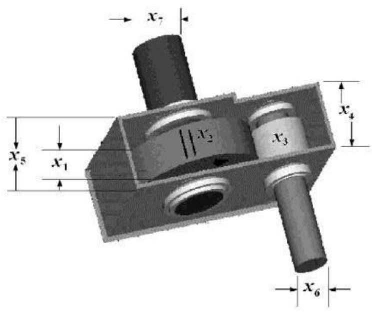

Fig. 3: Speed reducer

Welded beam design problem is a standard test problem for constrained design optimization. The problem aims to minimize the cost of beam subject to constraints on shear stress, τ, bending stress in the beam, σ, buckling load on the bar, Pc, end deflection of the beam, δ, and side constraints. Welded beam design is illustrated in Fig. 4. This problem consists of a nonlinear objective function, five nonlinear and two linear inequality constraints. The solution is located on the boundaries of the feasible region.

Problem 4: Welded beam

$\min f(X) = 1.10471 x_1 2\, x_2 + 0.04811 x_3 x_4 (14.0 + x_2)$
subject to:
$g_1(x)$: $\tau(x) - \tau_{max} \leq 0$
$g_2(x)$: $\sigma(x) - \sigma_{max} \leq 0$
$g_3(x)$: $x_1 - x_4 \leq 0$
$g_4(x)$: $0.10471 x_1^2 + 0.04811 x_3 x_4 (14.0 + x_2) - 5.0 \leq 0$
$g_5(x)$: $0.125 - x_1 \leq 0$
$g_6(x)$: $\delta(x) - \delta_{max} \leq 0$
$g_7(x)$: $P - P_c(x) \leq 0$

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

$\tau' = \frac{P}{\sqrt{2}x_1 x_2}$, $\tau'' = \frac{MR}{J}$,

$M = P\left(L + \frac{x_2}{2}\right)$, $R = \sqrt{\frac{x_2^2}{4} + \left(\frac{x_1 + x_3}{2}\right)^2}$

$$J = 2\left\{\frac{x_1 x_2}{\sqrt{2}}\left[\frac{x_2^2}{12} + \left(\frac{x_1 + x_3}{2}\right)^2\right]\right\}$$

$\sigma(x) = \frac{6PL}{x_4 x_3^2}$, $\delta(x) = \frac{4PL^3}{E x_3^3 x_4}$

$$P_c = \frac{4.013E\sqrt{\frac{x_3^2 x_4^6}{36}}}{L^2}\left(1 - \frac{x_3}{2L}\sqrt{\frac{E}{4G}}\right)$$

where P=6000 lb., L=14 in, δmax = 0.25 in, $E = 30 \times 10^6$ psi, G = 12 × 106 psi, τ_{max} = 1, 3600 psi , σ_{max} =
3,0000 psi, $X = (x1, x2, x3, x4)^T$, $0.1 \leq x_1, x_4 \leq 2.0$, $0.1 \leq x_2, x_3 \leq 10$.

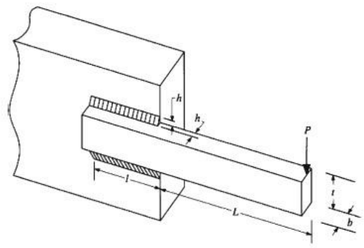

Fig. 4: Welded beam problem

6 Parameter settings and results

The performance of our modified algorithm was compared with the original ABC algorithm [34], particle swarm optimization (PSO) [35] and the evolution strategy [36]. We performed 30 independent runs per problem. Our algorithm used the same parameters' values as the original ABC algorithm: Swarm Size = 30, Maximum cycle number = 1000, Modification rate = 0.9.

The values of the algorithm-specific control parameters are listed in Table 2. The best and mean values with standard deviations are reported in Table 3. Comparisons show that our algorithm outperforms or performs similarly to three state-of-the-art approaches in terms of the quality of the resulting solutions. From the results, it can be concluded that our adjusted algorithm is a promising ABC modification for optimizing constrained engineering problems. Tables 3, 4, 5 and 6 show the solution vectors of the best solution reached by our algorithm and the values of the constrains for each of the problems tested.

PSO		**$(\mu+\lambda)$ES**		**ABC**		ourABC	
SS	30	μ	15	CSabc	30	CSabc	30
MNG	1,000	λ	100	MCN	1,000	MCN	1,000
ω	0.8	S_r	0.97	MR	0.9	MR	0.9
c_1	0.5	MGN	300	SPP	400	Limit	MCN/(2*CSabc)
c_2	0.5	LR	$\tau = (\sqrt{2\sqrt{n}})^{-1}$ $\tau' = (\sqrt{2n})^{-1}$	Limit	CSabc*D*5		
		MSS	$\sigma_i(0) = 0.4(\Delta x_i / \sqrt{n})$				

Table 2. The values of the control parameters of the algorithms: SS Swarm size, MGN Maximum generation number, ω inertia, c_1 cognitive component, c_2 social component, S_r Selection ratio, LR Learning rate, MSS Mutation step size, CSabc Colony size, MCN Maximum cycle number, SPP Scout production period, MR Modification rate

Problem	Stats.	PSO	$(\mu + \lambda)$-ES	ABC	ourABC
Pressure vessel	Best	6059.714	6059.701	6059.714	6059.714
	Mean	6289.928	6379.938	6245.308	6218.515
	St. Dev	3.1E+02	2.1E+02	2.05.E+02	1.9 E+02
Ten/comp. spring	Best	0.0126	0.0126	0.0126	0.0126
	Mean	0.012	0.013	0.0127	0.0127
	St. Dev	4.1E-05	3.9E-04	1.28E-4	2.8E-4
Speed reducer	Best	NA	2996.348	2997.058	2996.783
	Mean	NA	2996.348	2997.058	2996.783
	St. Dev	NA	0.000	0.000	0.000
Welded beam	Best	NA	1.724	1.724	1.724
	Mean	NA	1.777	1.741	1.763
	St. Dev	NA	0.088	0.031	0.033

Table 3. Statistical results of the PSO, $(\mu + \lambda)$-ES, ABC and ourABC algorithms

	Best Solution
x_1	3.4999
x_2	0.6999
x_3	17.0000
x_4	7.3000
x_5	7.8000
x_6	3.3502
x_7	5.2872
$g_1(x)$	-0.0739
$g_2(x)$	-0.1979
$g_3(x)$	-0.4991
$g_4(x)$	-0.9015
$g_5(x)$	0.0000
$g_6(x)$	0.0000
$g_7(x)$	-0.7025
$g_8(x)$	0.0001
$g_9(x)$	-0.5833
$g_{10}(x)$	-0.0513
$g_{11}(x)$	-0.0106
$f(x)$	2996.783

Table 4. Parameter and constraint values of the best solutions obtained for pressure speed reducer

	Best Solution
x_1	0.8125
x_2	0.4375
x_3	42.0985
x_4	176.6366
$g_1(x)$	0.0000
$g_2(x)$	-0.03588
$g_3(x)$	-0.00003
$g_4(x)$	-63.3634
$f(x)$	6059.714

Table 5. Parameter and constraint values of the best solutions obtained for pressure vessel problem

	Best Solution
x_1	0.051749
x_2	0.3581
x_3	11.2015
$g_1(x)$	-0.0002
$g_2(x)$	0.0000
$g_3(x)$	-4.0618
$g_4(x)$	-0.7246
$f(x)$	0.0127

Table 6. Parameter and constraint values of the best solutions for tension/compression spring problem

	Best Solution
x_1	0.2057
x_2	3.4705
x_3	9.0366
x_4	0.2057
$g_1(x)$	0.0000
$g_2(x)$	0.0000
$g_3(x)$	0.0000
$g_4(x)$	-3.4329
$g_5(x)$	-0.0807
$g_6(x)$	-0.2355
$g_7(x)$	0.0000
$f(x)$	1.724

Table 7. Parameter and constraint values of the best solutions obtained for welded beam problem

7 Conclusion

A new method is introduced in this paper, which improves the performance of the ABC algorithm by incorporating adaptive scout behavior modification. The approach obtains competitive results on 4 well-known constrained engineering problems. From the comparative study our modified algorithm has shown its potential to handle various constrained problems and its performance is better or similar to the original ABC algorithm, so we can conclude that this mechanism does improve the robustness of the ABC.

The results obtained in this paper provide an understanding and improvement of the search mechanism of the ABC algorithms for the constrained optimization problems. Several directions need to be explored in the future work like evaluating the modifications of the improved algorithm on the other optimization problems and performing a more detailed statistical analysis of the performance of our proposed approach. Also, a constriction factor from the PSO algorithm incorporated in our adjusted ABC algorithm may introduce further performance improvement.

References:

[1] Raymond Chiong, *Nature-Inspired Algorithms for Optimisation*, ISBN 978-3-642-00266-3, Vol. 193, p. 536, Springer, 2009.

[2] Karaboga D, Basturk B, Artificial bee colony (ABC) optimization algorithm for solving constrained optimization problems, *Advances in Soft Computing: Foundations of Fuzzy Logic and Soft Computing*, Vol. 4529/2007, No. 0302-9743, 2007, pp.789–798.

[3] N. Gomez, L. F. Mingo, J. Bobadilla, F. Serradilla, J. A. C. Manzano: Particle Swarm Optimization models applied to Neural Networks using the R language, *WSEAS Transactions on Systems*, Vol. 9, Issue 2, 2010, pp. 192-202

[4] R. M. Idris, A. Khairuddin and M.W. Mustafa, Optimal Allocation of FACTS Devices in Deregulated Electricity Market Using Bees Algorithm, *WSEAS Transactions on Power Systems*, Vol. 5, Issue 2, Apr 2010, pp. 108-119.

[5] A.V. Baterina, C. Oppus, Image Edge Detection Using Ant Colony Optimization, *WSEAS Transactions On Signal Processing*, Vol. 6, No. 2, 2010, pp.58-67.

[6] Milan Tuba: Swarm Intelligence Algorithms Parameter Tuning, Proceedings of the American Conference on Applied Mathematics (AMERICAN-MATH'12), pp. 389-394, Harvard, Cambridge, USA, January 2012

[7] L. Jiann-Horng, H. Li-Ren: Chaotic bee swarm optimization algorithm for path planning of mobile robots, *Proceedings of the 10th WSEAS international conference on evolutionary computing*, pp. 84-89, 2009.

[8] E. Ridge, D. Kudenko, D. Kazakov, E. Curry, Moving Nature-Inspired Algorithms to Parallel, Asynchronous and Decentralised Environments, *Proceeding of the 2005 Conference on Self-Organization and Autonomic Informatics*, Vol. 135, pp. 35-49, 2005

[9] Michalewicz Z, Fogel D B, *How to Solve It: Modern Heuristics*, Springer, 2004.

[10] E. Mezura-Montes (Ed.), *Constraint-Handling in Evolutionary Optimization*, Springer-Verlag. Studies in Computational Intelligence, Vol. 198, p.264, 2009.

[11] Mezura-Montes Efren; Coello Coello Carlos A, An empirical study about the usefulness of evolution strategies to solve constrained optimization problems, *International Journal of General Systems*, Vol, 37, pp. 443-473, 2008

[12] T. Jayabarathi, S. Chalasani, Z. A. Shaik, Hybrid Differential Evolution and Particle Swarm Optimization Based Solutions to Short Term Hydro Thermal Scheduling, *WSEAS Transactions On Power Systems*, Vol. 2, No. 11, 2007, pp. 245-254

[13] Haiyan Lu, Weiqi Chen, Self-adaptive velocity particle swarm optimization for solving constrained optimization problems, *Journal of global optimization*, Vol. 41, No. 3, 2008, pp. 427-445.

[14] Karaboga D, Basturk B, A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. *J. of Global Optimization*, Vol. 39, No. 0925-5001, 2007, pp. 459-471.

[15] Nebojsa Bacanin, Milan Tuba, and Ivona Brajevic: Performance of object-oriented software system for improved artificial bee colony optimization, International Journal of Mathematics and Computers in Simulation, Vol. 5, Issue 2, 2011, pp. 154-162

[16] Milos Subotic, Milan Tuba and Nadezda Stanarevic: Different approaches in parallelization of the artificial bee colony algorithm, International Journal of Mathematical Models and Methods in Applied Sciences, Vol. 5, Issue 4, 2011, pp. 755-762

[17] Coello C. A. C. Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: a survey of the state of the art, *Computer Methods in Applied Mechanics and Engineering*, Vol. 191, pp. 1245-1287, 2002.

[18] Hamida S B, Schoenauer M., ASCHEA: new results using adaptive segregational constraint handling. *Proceedings of the 2002 Congress on Evolutionary Computation*, pp. 884-889, DOI: 10.1109/CEC.2002.1007042 , 2002

[19] Efren Mezura-Montes, Mauricio Damian-Araoz and Omar Cetina-Dominguez, Smart Flight and Dynamic Tolerances in the Artificial Bee Colony for Constrained Optimization, in Proceedings of the IEEE Congress on Evolutionary Computation (CEC'2010), pp. 4118-4125, 2010.

[20] Efren Mezura-Montes, Carlos A. Coello, Adding a diversity mechanism to a simple evolution strategy to solve constrained optimization problems, *In Proceedings of the Congress on Evolutionary Computation 2003 (CEC'2003)*, Vol. 1, pages 6–13. 2003.

[21] Guopu Zhu, Sam Kwong, Gbest-guided artificial bee colony algorithm for numerical function optimization, *Applied Mathematics and Computation*, Vol. 217, Nr. 7, DOI: 10.1016/j.amc.2010.08.049, pp. 3166-3173, 2010.

[22] Efren Mezura-Montes, Omar Cetina-Dominguez, Exploring Promising Regions of the Search Space with the Scout Bee in the Artificial Bee Colony for Constrained Optimization, *Proceedings of the Artificial Neural Networks in Engineering Conference (ANNIE'2009)*, Vol. 19, pp.: 253--260, ASME Press Series, Intelligent Engineering Systems Through Artificial Neural Networks, 2009.

[23] Deb K, An Efficient Constraint-handling Method for Genetic Algorithms, *Computer Methods in Applied Mechanics and Engineering*, Vol. 186, No. 0045-7825, 2000, pp. 311-338.

[24] Deb K. Optimization for Engineering Design, Algorithms and Examples 4th. Edition, Chapter 1: Introduction, pp. 1–30, Prentice-Hall India, 2000.

[25] Nebojsa Bacanin, Milan Tuba: Artificial Bee Colony (ABC) Algorithm for Constrained Optimization Improved with Genetic Operators, Studies in Informatics and Control, Vol. 21, Isuue 2, June 2012, pp.

[26] Ivona Brajevic and Milan Tuba: An upgraded artificial bee colony algorithm (ABC) for constrained optimization problems, Journal of Intelligent Manufacturing, ISSN: 0956-5515, published Online First, Jan 2012, DOI: 10.1007/s10845-011-0621-6

[27] Nadezda Stanarevic, Milan Tuba, and Nebojsa Bacanin: Modified artificial bee colony algorithm for constrained problems optimization, International Journal of Mathematical Models and Methods in Applied Sciences, Vol. 5, Issue 3, 2011, pp. 644-651

[28] Ivona Brajevic, Milan Tuba, Milos Subotic: Performance of the improved artificial bee colony algorithm on standard engineering constrained problems, International Journal of Mathematics and Computers in Simulation, Vol. 5, Issue 2, 2011, pp. 135-143

[29] Mezura-Montes, Efren; Miranda-Varela, Mariana Edith; Gomez-Ramon, Rubi del Carmen, *Differential evolution in constrained numerical optimization: An empirical study*, Information Sciences, vol. 180, 2010, pp. 4223-4262

[30] Milan Tuba: Artificial Bee Colony (ABC) Algorithm Exploitation and Exploration Balance, Proceedings of the 1st Internatonal Conference on Computing, Information Systems and Communications (CISCO '12), 2012, Singapore, pp.252-257

[31] R. M. Gamot, A. Mesa: Particle swarm optimization: Tabu search approach to constrained engineering optimization problems, *WSEAS Transactions on Mathematics*, Vol.7, 2008, pp. 666-675.

[32] W. Gao, S. Liu and L. Huang: A global best artificial bee colony algorithm for global optimization, *Journal of computational and applied mathematics*, Vol. 236, issue 11, pp. 2741-2753, 2012.

[33] K. Diwold, A. Aderhold, A. Scheidler and M. Middendorf: Performance evaluation of artificial bee colony optimization and new selection schemes, *Memetic computing*, vol. 3, issue 3, pp. 149-162, 2011.

[34] Akay B, Karaboga D, Artificial bee colony algorithm for large-scale problems and engineering design optimization, *Journal of Intelligent Manufacturing*, pp. 1-14, 2010

[35] Parsopoulos, K., Vrahatis, M., Unified particle swarm optimization for solving constrained engineering optimization problems, *Advances in natural computation*, Vol. 3612, 2005, pp. 582–59.

[36] Mezura-Montes, E., Coello C. A. Useful infeasible solutions in engineering optimization with evolutionary algorithms, *In Proceedings of the 4th Mexican International Conference on Artificial Intelligence*, Vol. 3789, 2005, pp. 652–662