An Improved Artificial Bee Colony Algorithm for Linear Least Squares and Constrained Optimization Problems

Zahraa Tariq Mohammed Taher
Statistics and information Department,
Faculty of Computer Science& Mathematics,
Mosul University, Iraq.

ABSTRACT
This paper proposes An Improved Artificial Bee Colony (Deb's-ABC) algorithm for solving constrained optimization problems and Nonnegative linear least squares problems. The proposed approach introduces different methods Based upon new search mechanism to balance exploration and exploitation abilities, generating initial population by using the orthogonal initialization method for achieving initial population that spread regularly over the feasible solution and to enhance the global convergence. In addition, we relax the Deb's rules by replacing the feasible solutions with the approximate feasible solutions in Deb's rules because some infeasible solutions with better objective function value and small violation may carry more important information than some feasible solutions .This algorithm is tested on several benchmark functions. Experimental results compared with a standard ABC and other algorithms show that the proposed algorithm is efficient and competitive algorithm for solving constrained optimization problems.

General Terms
ABC Algorithm, Deb's rules ,LLS,COP .

Keywords
constrained optimization Problems ,Artificial Bee Colony, Swarm Intelligent, Least Square Error .

1. INTRODUCTION
Constrained optimization problem consist of engineering design ,economics, structural optimization and VLSI design[1].There are two types of optimization problems which have combined (continuous and discrete) design variables, nonlinear objective functions and nonlinear constrains[2].The aim of this problem is to find parameter vector $\vec{x}$ that minimizes an objective function f($\vec{x}$) [3].

The objective function f is represented on a search space S ,which is defined as a n-dimensional rectangle in R^n ($S \subseteq R^n$).$F \subseteq S$ is called feasible region and is defined by a set of m additional constraints ($m \geq 0$) and $\vec{x}$ is defined on feasible space ($\vec{x} \in F \in S$).

Where :

$X=(x_1,\ldots..,x_n) \in R^n$

The variable domains are limited by their lower and upper bounds:

$l_i \leq x_i \leq u_i, \ 1 \leq i \leq n$

$g_j(x) \leq 0, \quad for \quad j=1,2,\ldots.q$

$h_j \leq 0, \quad for \quad j=q+1, \ldots.. \ m$

As well as, The algorithm of least squares (LLS) is a serious,simple and effective method for solving the approximate functions of over complex systems.Naturally,It shows when one would like to estimate values of parameters of a statistical model from measured data, which are subject to errors[4].As well as, LLS is a derivative based algorithm and the cost function can be represented by reduces the squares of errors and also frequently used in the fitting function to get the linear relationship between input and output [5], [6][7].There are two types of Least squares problems: linear least squares and non-linear least squares, which depend on whether or not the residuals are linear in all unknowns [7]. Without loss of generality, the Nonnegative Linear Least Squares (NLLS) problem can be formulated as follows:

$$\min_{x \geq 0} f(x) = \frac{1}{2}\|Ax-b\|^2 = \frac{1}{2}(Ax-b)^T(Ax-b) \ldots\ldots (1)$$

Where $A \epsilon R^{m \times n}$, $m \geq n, rank(A) = n, b \epsilon R^m$

In the last time, many researchers solved constraints optimization problem and linear least squares by using traditional expensive methods .these methods suffered from slow convergence and frequently converging to local optimum solution. In a recent decade, Swarm Intelligence (SI) algorithms have shown considerable success in solving many constraints optimization problems,LLS and attracted more attention in recent years. For examples of these algorithms[8]: Genetic Algorithm (GA) [9], Particle Swarm Optimization (PSO) [10],differential evolution(DE)[11], and Artificial Bee Colony (ABC) algorithm [12] and so on.

(ABC) algorithm depend on the foraging behaviour of honey bees for optimization problems [5].Many researchers have compared the performance of the ABC algorithm with the other population algorithms such as Genetic Algorithm (GA), ant colony algorithm, Particle Swarm Optimization (PSO) on unconstrained and constrained problems. Results have been shown that the ABC algorithm is better than other heuristic algorithms for solving unconstrained and constrained optimization problems and LLS problems.

In this paper, our approach is to applied an improved Artificial Bee Colony Algorithm (Deb's-ABC) on constraints and LLs optimization problems , by using new search mechanism to balance exploration and exploitation abilities, generating initial population by using the orthogonal initialization method [13] for achieving initial population that scattered uniformly over the feasible solution and to enhance the global convergence when producing initial population. In addition, we relax the Deb's rules by replacing the feasible solutions with the approximate feasible solutions in Deb's rules because some infeasible solutions with better objective function value and small violation may carry more useful information than some feasible solutions .This algorithm is tested on several benchmark functions where the numerical results show that the proposed (Deb's-ABC) algorithm has an outstanding performance for the optimization problems.

2. LITERATURE REVIEW

Swarm Intelligence is a field of Artificial Intelligence that is used to model the collective intelligent behaviour of social swarms in nature[13] .It is used for solving constraints and inconstraints problems in a way that is inspired by the behaviour of real swarms or insect colonies like: ant colony, PSO Algorithm, evolutionary algorithm and Bee colony algorithm. Many researchers used swarm intelligent algorithms for solving complex problems and others improved swarm intelligent algorithms inorder to produce an efficient results for solving constraints and inconstraints problems .

In 2010, Leticia C. , Susana C. and Carlos A. present particle swarm optimization algorithm for solving general constrained optimization problems by allowing the boundary between the feasible and unfeasible regions to be explored because unfeasible solutions play an important role when trying to solve problems with active constraints so, they used shake mechanism when the percentage of unfeasible individuals is higher than 10% .This algorithm provide better diversity maintenance and better exploration of constrained search spaces rather than standard PSO algorithm[14]. In 2011 ,Dervis K. and Bahriye A. modified ABC algorithm by using Deb's rules instead of selection mechanism in order to cope with the constraints and introducing a probabilistic selection that assigns probability values to feasible solutions based on their fitness values and infeasible solutions based on their violations .Results show that improved ABC algorithm is efficiently for solving constraints optimization problems [3].

In 2013 , Bingqin Qiao and others propose a hybrid particle swarm optimization algorithm based on the multiplier penalty function to solve constrained optimization problems .This hybrid algorithm used the multiplier penalty function values as the fitness of particles and also made the particle swarm track the best particle and fly to the global best[15].In 2014, Xiangyu Kong and others ,produced a hybrid artificial bee colony algorithm for solving nonnegative linear least squares problems.This algorithm used orthogonal initialization method to generate the initial swarmand a new search mechanism called (DE/best) is designed to balance the exploration and exploitation abilities. Numerical results demonstrate that the proposed algorithm is better than other algorithms for global optimization problems and nonnegative linear least squares problems[7]. In 2015, Yaosheng Liang and others produced an improved artificial bee colony (I-ABC) algorithm for COPs .This paper used a selection strategy based on rank selection and designed a search mechanism using the information of the best-so-far solution to balance the exploration and the exploitation at different stages ,as well as, introducing the approximate feasible solutions to suitably utilize the information of the infeasible solutions with better objective function value and small violation which called Deb's rules .The numerical results show that the I-ABC algorithm has an outstanding performance for the constraints optimization problems [16] .In 2015,A. J. Umbarkar and Others used a new in the area of evolutionary algorithms for solving optimization problems called Dual Population Genetic Algorithm. It solves constraints optimization problems by applying Maximum Constrains Satisfaction method by using DPGA which is a novel technique that tries to satisfy maximum constrains first and then it attempts to optimize objective function.The results are close to optimum value but fails to obtain exact optimum solution[17].In 2016, Chun-Feng Wang, and Yong-Hong Zhang used a chaotic system and an opposition-based method for initial population also, they adopted a chaotic search in the best solution of the current iteration to improve the exploitation of onlooker bees. Results demonstrate that proposed algorithm is better than other standard algorithms [18].In 2017, Pintu Pal presented Hybrid and Particle Swarm Optimization (HPSO). The basic ideas behind this concept are inspired from winning match of tournament that which applied between pair wise populations in different round of the tournament using PSO .the results show that the proposed method is better than other methods in terms of different comparative parameters[19].

3. ORIGINAL ARTIFICIAL BEE COLONY ALGORITHM (ABC)

In 2005, Karaboga [5] invented a new heuristic population algorithm which simulate foraging behaviour waggle dance of honey bee swarm. In this algorithm, the food source refers to the solution of the optimization problem and the nectar amount of food source denotes fitness value of the associated solution . The artificial bee colony consists of three groups of bees: employed bees, onlookers and scout. The hive classified in to 50% employed bees and 50% onlooker bees . The employed bees exploring the food sources and sharing the information of the food sources with the onlooker bees .The onlooker bees will exploit one of the food sources . The probability of the food source(solution) being selected by the onlooker bee is proportional to the quality of the food[16]. The number of employed bees is equal to the number of solutions SN in the population.At initialization step, ABC generates initial population of SN solutions randomly using the following equation :

$$x_{ij} = x_{ij} + \emptyset_{ij}(x_{ij} - x_{kj}) \qquad (2)$$

Where $k \in \{1,2,\dots.,SN\}$, and $J \in \{1,2,\dots.,D\}$. A randomly chosen index k has to be different from I and $\emptyset_{ij}$ is a random number in the range[-1,1] .

These food sources are randomly assigned to SN number of employed bees and their finesses are evaluated. The search equation for employed bees and onlooker bees can be described as follows:

$$x_{new(j)} = x_{i,j} + \emptyset_{i,j}(x_{i,j} - x_{k,j}) \qquad (3)$$

Once x_{new} is obtained, a greedy selection then performs between the old and candidate solutions. In the onlooker bee step, an onlooker bee selects a food source x_i depending on the probability value p_i calculated as follows:

$$p_i = \frac{f_i}{\sum_{i=1}^{SN} f_i} \qquad (4)$$

where f_i is the fitness value of the ith food source x_i . the higher the f_i is, the more probability that the ith food source is selected.

If a position cannot be improved through predetermined number of cycles, then the food source is assumed to be abandoned. The value of predetermined number of cycles is an important control parameter of ABC algorithm, which is called limit for abandonment. Assume that the abandoned source is Xi and j∈{1,2,...,D},then the scout discovers a new food source to be replaced with Xi.This operation can be defined as :

$$x_{i,j} = x_{min,j} + \text{rand}(0,1)(x_{max,j} - x_{min,j}) \qquad (5)$$

Original artificial bee colony algorithm

Initialize the population of solution
Evaluate the initial population cycle = 1

Repeat

Employed bee phase

Apply greedy selection process

Calculate the probability values

Onlooker bee phase

Scout bee phase

Memorize the best solution achieved so far

cycle = cycle+1

until cycle = maximum cycle number

fig.1: show Original artificial bee colony algorithm

4. Proposed Algorithm

4.1 Initial Population by using (1.1) Algorithm:

Set of food source positions which are selected by the bees using an orthogonal array to determine a small number of combinations that are scattered uniformly over the space of all possible combinations, to provide information about the location of the solution. So that, in this paper we generate initial population by using the orthogonal initialization method [13][20][7].

Step 1.1 :

Divide the feasible solution space [l,u] into S subspaces $[l_1, u_1]$, $[l_2, u_2]$, L ,$[l_s, u_s]$ based on the following equations:

$$\begin{cases} l_i = l + (i-1)\left(\frac{u(s)-l(s)}{S}\right) l_{s,} \\ u_i = u + (S-i)\left(\frac{u(s)-l(s)}{S}\right) l_{s,} \end{cases} \quad i = 1,2,L\ ,S. \qquad (6)$$

Here , $u(s) - l(s) = \underset{LsisD}{max}\{u_i - l_i\}$.

Step 1.2 :

Quantize subspace $[l_i, u_i]$ into Q1 levels based on

$$\alpha_{ij} = \begin{cases} l_{i,} & j = 1, \\ l_i + (j-1)\left(\frac{u_i - l_i}{Q_1 - 1}\right) & 2 \le j \le Q_1 - 1, \\ u_i, & j = Q_1, \end{cases} \qquad (7)$$

Where Q_1 is odd.Afterthat, construct orthogonal array $L_{M_1}(Q_1^N) = [a_{ij}]_{M_1 * N}$ to select M_1 individuals based on

$$\begin{cases} (\alpha_{1,a_{11}}, \alpha_{2,a_{12}}, L, \alpha_{N,a_{1N}}) \\ (\alpha_{1,a_{21}}, \alpha_{2,a_{22}}, L, \alpha_{N,a_{2N}}) \\ L \\ (\alpha_{1,a_{M_11}}, \alpha_{2,a_{M_12}}, L, \alpha_{N,a_{M_1N}}) \end{cases} \qquad (8)$$

Here, $L_{M_1}(Q_1^N)$) can be generated as follows. Select the smallest J_1 fulfilling $(Q_1^{J_1} - 1)/(Q_1 - 1) \ge N$ .If $(Q_1^{J_1} - 1)/(Q_1 - 1) = N$, then N′=N else N′=$(Q_1^{J_1} - 1)/(Q_1 - 1)$ .Then, construct the basic columns based on $j = \frac{Q_1^{k-1} - 1}{Q_1 - 1} + 1$, $a_{ij} = \left\lfloor \frac{i-1}{Q_1^{J_1 - k}} \right\rfloor mod\ Q_1$, for i=1,…,L , M_1, K=1,L,J_1. Construct the non- basic columns as $j = \frac{Q_1^{k-1} - 1}{Q_1 - 1} + 1$, $a_{j+(s-1)(Q_1-1)+t} = (a_s * t + a_j) mod Q_1$, for s =1,L , j-1 , t=1,L, Q_1 . Thus, the orthogonal array $L_{M_1}(Q_1^{N'})$ is constructed. Delete the last N′-N columns of $L_{M_1}(Q_1^{N'})$ to get $L_{M_1}(Q_1^N)$ where $M_1 = Q_1^{J_1}$.

Step 1.3 :

Among the M_1S individuals, select SN individuals having the smallest cost as the initial population.

4.2 Search Mechanism by using Differential evolution (DE)

The bee algorithm achieved high results in the solution of constrained optimization problems[3], and showed better performance than other algorithms like :PSO ,GA and Ant Colony Algorithms .The swarm intelligent Algorithms try to achieve balancing between exploration and exploitation abilities to check good optimization performance . exploration means the ability to search the indefinite regions in the solution space to find the global optimum, while the exploitation refers to the ability to reach better solutions based on the information of the previous good solutions.So that , the search equation in ABC Algorithm is not good at exploitation. In order to improve it, a new search mechanism is proposed by Differential evolution (DE) .

Differential evolution (DE) is a simple efficient evolutionary based algorithm, whose follows the general stages of an evolutionary algorithm. It depends on generating a new position for an individual by calculating vector differences between other randomly selected individuals in the population. In DE algorithm, there are three evolutionary important operations : mutation, crossover and selection . There are several types of mutation operation, which adopted differently. Among them, "DE/best/2" can improve the exploitation abilities of algorithm, which can be described as follows:

$$DE/best/1'': Vi = X_{best} + F(Xr1 - Xr2), \qquad (9)$$

$$'DE/best/2'': Vi = X_{best} + F(Xr1 - Xr2) + F(Xr3 - Xr4), \quad (10)$$

Where i={1,2,...,SN}and r_1,r_2,r_3 and r_4 are different random integer which selected from{1,2,...,SN}.x_{best} is the global best solution; F∈ [0.5,1] is a positive real number, which is known as scaling factor or amplification factor .Depending on the property of ABC algorithm and two different DE algorithms, a new search strategies is proposed as follows :

''ABC/best/1'': $v_{i,j} = x_{best,j} + \Phi_{i,j}(x_{r1,j} - x_{r2,j})$, *(11)*

''ABC/best/2'': $v_{i,j} = x_{best,j} + \Phi_{i,j}(x_{r1,j} - x_{r2,j}) + \Phi_{i,j}(x_{r3,j} - x_{r4,j})$, *(12)*

Where the indices r_1,r_2,r_3 and r_4 are different from the base index i. j∈{1,2,L,D} is a randomly selected index; $\Phi_{i,j}$ is a random number in the range[−1,1].Eq.(11)or Eq.(12),can choose the new candidate solution only around the best solution of the old iteration[21].

4.2.1 Evaluate the Fitness Function

The greedy selection is replaced by Deb's rules [10] to process the constraints:

– Any feasible solution is preferred to any infeasible solution.

– Among two feasible solutions, the one having better objective function value is preferred.

– Among two infeasible solutions, the one having smaller constraint violation is preferred.

Our aim is to exploit the important information in infeasible solution near the optimum point more than feasible solution that far away from the optimum point .So, we use the Deb's rules by replacing the feasible solutions with the approximate feasible solutions. Here, a solution x_i may be an approximate feasible solution if the penalty value $vio(x_i)$ does not exceed an adaptive control parameter d_t, which should be given a big value to allow more infeasible solutions to survive to keep the diversity of the colony at the first stage and become smaller during the iteration to guarantee the gained final solution to be the feasible solution[16]. the Formula of control parameter d_t as follows :

$$d_t = \begin{cases} d_0\left(\frac{\propto}{d_0}\right)^{\frac{t}{T_1}}, & if\ t \le T_1, \\ \propto \left(1 - \frac{t}{T_2}\right), & if\ T_1 < t \le T_2, \\ 0 & otherwise, \end{cases} \qquad (13)$$

Where (t) is the number of the current cycle, d_0 is an acceptable initial penalty value which corresponds to the median value of all the penalty values, a is a pre-set small nonnegative real number, T_1 and T_2 are the cycle numbers where the control parameter d_t will become α and zero, respectively.

So, the selection mechanism between two solutions using the relaxed Deb's rules can be described as follows:

– If the one is the approximate feasible solution and the other is the infeasible solution with a penalty value exceeding dt, the approximate feasible solution is preferred.

– If they are both approximate feasible solutions, the one having better objective function value is preferred.

– If the penalty values of them exceed dt, the one having smaller constraint violation is preferred.

4.2.2 Calculate the probability

In order to achieve balancing between the objective function value and the constraints in the selection progress of the probability, we can use rank selection [16] which has been used for solving the unconstrained optimization problems such as tournament selection mechansim and disruptive selection mechansim [16] [22] . The detail of the rank selection strategy can be described as follows:

- Rank the SN solutions in ascending order based on their violation degrees of the constraints $vio(x_i)$ and the population denoted by P_1 after ranking.
- Sort the former s solutions of the population P_1 in ascending order according to their objective function value $f(x_i)$ and the population denoted by P_2 after this rank. Where, s is determined by :

$$s = \max\{i | vio(x_i) \le d_t, x_i \in P_1\}$$

Calculate the probability p_i of the solution x_i by the following formula [22]:

$$\left.\begin{aligned} p_i &= \frac{1}{SN} + a_t \frac{SN+1-2i}{SN(SN+1)}, \quad i=1,2,\ldots,SN, \\ a_t &= 0.2 + \frac{3t}{4MCN}, t = 1,2,\ldots.MCN, \end{aligned}\right\} \qquad (14)$$

where (t) represents the number of the current cycle and MCN is the maximum pre-set number of cycles.

1. If there is a x_i (optimal solution)need to be abandoned, replace it by Eq (15) :

$$x_{ij} = x_{ij} + \varphi_{ij}(x_{kj} - x_{ij}) + (1 - \varphi_{ij})(x_{bj} - x_{ij}) \qquad (15)$$

Where k is a random integer selected from {1,…,i-1,i+1,…SN},x_{bj} is the jth parameter of the best-so-far solution x_b and φ_{ij} is a randomly generated value in [–1,1].

Algorithm 1.2:show proposed algorithm

1: Initialization: preset population size SN and limit by Perform Algorithm 1.1 to create an initial population x_i,i∈{1,⋯,SN}.

2: Cycle=1

3: repeat

4: Employed bee explores x_i to generate v_i by Eq.(11) or by Eq.(12) .

5: Make a selection between x_i and v_i by the relaxed Deb's rules eq.(13).

6: Calculate the probability p_i of x_i based on rank selection by eq.(14).

7: Onlooker bee selects a solution x_i with a probability p_i.

8: Onlooker bee exploits x_i by Eq.(11) or by Eq.(12) to generate v_i.

9: Make a selection between xi and vi by the relaxed Deb's rules eq.(13).

10: If there is a xi need to be abandoned, replace it by Eq.(15).

11: Memorize the best-so-far solution.

12: Cycle=Cycle+1.

13: until Cycle=Maximum cycle number(MCN)

14: return the best solution.

Fig2.show proposed algorithm (Deb's- ABC)

5. EXPERIMENT ANALYSIS

This article has introduced a ABC-based approach for constrained optimization problems, called (Deb's- ABC).This approach introduces relatively many changes to a traditional ABC algorithm, aiming to provide better diversity maintenance and better exploration of constrained search spaces. The simulation experiments are performed on 2.40 GHz Core i5 with 4GB RAM using Matlab 2010a software .we used a set of 20 benchmarks problems described in [21][7] and 3 nonnegative linear least squares test problems described in[7].

5.1 Experiments and comparisons:

Do not include headers, footers or page numbers in your submission. These will be added when the publications are assembled.

5.1.1 Benchmark functions and parameter settings

This paper applied 20 benchmark functions and 3 nonnegative linear least squares test problems in order to minimize them . The benchmark functions presented in Table1 are tested of dimension D=30 and D=60. Comparison is performed to show the performance of proposed algorithm with that of ABC algorithm.

Table1. Benchmark functions f1 - f 20 used in experiments. D: Dimension

functions	D	Search range
$f_1(X) = \sum_{i=1}^{D} x_i^2$	30 and 60	$(-100, 100)^D$
$f_2(X) = \sum_{i=1}^{D} \lvert x_i \rvert + \prod_{i=1}^{D} \lvert x_i \rvert$	30 and 60	$(-10, 10)^D$
$f_3(X) = max_i\{\lvert x_i \rvert, 1 \le i \le D\}$	30 and 60	$(-100, 100)^D$
$f_4(X) = \sum_{i=1}^{D} (\lfloor x_i + 0.5 \rfloor)^2$	30 and 60	$(-100, 100)^D$
$f_5(X) = \sum_{i=1}^{D} i x_i + rand[0,1)$	30 and 60	$(-1.28, 1.28)^D$
$f_6(X) = \sum_{i=1}^{D} \left[x_i^2 - 10cos(2\pi x_i) + 10\right]$	30 and 60	$(-5.12, 5.12)^D$
$f_7(X) = \sum_{i=1}^{D} \left[y_i^2 - 10cos(2\pi y_i) + 10\right]$	30 and 60	$(-5.12, 5.12)^D$
$f_8(X) = \frac{1}{4000}\sum_{i=1}^{D} x_i^2 - \prod_{i=1}^{D} \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	30 and 60	$(-600, 600)^D$
$f_9(X) = D * 418.982887 - \sum_{i=1}^{D} \left(x_i \sin\left(\sqrt{\lvert x_i \rvert}\right)\right)$	30 and 60	$(-500, 500)^D$
$f_{10}(X) = -2\exp\left(-0.2\sqrt{\frac{1}{D}\sum_{i=1}^{D} x_i^2}\right) - \exp\left(\frac{1}{D}\sum_{i=1}^{D} cos2\pi x_i\right) + 20 + e$	30 and 60	$(-32, 32)^D$
$f_{11}(X) = \frac{\pi}{D}\{10sin^2(\pi y_1) + \sum_{i=1}^{D}(y_i - 1)^2[1 + sin^2(\pi y_{i+1}] + (y_D - 1)^2 + \sum_{i=1}^{D} u(x_i, 10,100,4)$	30 and 60	$(-50, 50)^D$
$f_{12}(X) = 0.1\{sin^2(3\pi x_1) + \sum_{i=1}^{D}(x_i - 1)^2[1 + sin^2(3\pi y x_{i+1}] + (y_D - 1)^2[1 + sin^2(3\pi x_D)]\} + \sum_{i=1}^{D} u(x_i, 5,100,4)$	30 and 60	$(-50, 50)^D$
$f_{13}(X) = \sum_{i=1}^{D} \lvert x_i . \sin(x_i) + 0.1 . x_i \rvert$	30 and 60	$(-10, 10)^D$
$f_{14}(X) = \sum_{i=1}^{D}(x_i - 1)^2[1 + sin^2(3\pi y x_{i+1}] + sin^2(3\pi y x_1) + \lvert x_D - 1 \rvert[1 + sin^2(3\pi y x_D)]$	30 and 60	$(-10, 10)^D$
$f_{15}(X) = \sum_{i=1}^{D}(\sum_{k=0}^{k\,max}[a^k \cos(2\pi b^k(x_i + 0.5))]) - D\sum_{k=0}^{k\,max}[a^k \cos(2\pi b^k(x_i + 0.5))] \quad where\ a = 0.5, b = 3, kmax = 20$	30 and 60	$(-0.5, 0.5)^D$

functions	D	Search range
$f_{16} = -20exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}\right) - \exp\left(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)\right) + 20 + e$	30 and 60	$(-32,32)^D$
$f_{17} = (x_1 - 1)^2 + \sum_{i=2}^{n} i\left(2x_i^2 - x_{i-1}\right)^2$	30 and 60	$(-10,10)^D$
$f_{18} = \frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n}\cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	30 and 60	$(-600,600)^D$
$f_{19} = sin^2(\pi y_1) + \sum_{i=1}^{n-1}[(y_1 - 1)^2(1 + 10sin^2(\pi y_1 + 1))] + (y_n - 1)^2(1 + 10sin^2(2\pi y_n))$; $y_i = 1 + \frac{x_i - 1}{4}$, $i = 1, L, n.$,	30 and 60	$(-10,10)^D$
$f_{20} = \sum_{i=1}^{n} x_i^2 + (\sum_{i=1}^{n} 0.5ix_i)^2 + (\sum_{i=1}^{n} 0.5ix_i)^4$	30 and 60	$(-5,10)^D$

5.1.2 Comparison of Deb's ABC with ABC :

This section shows the comparison between Debs's ABC and the original ABC algorithm. Both algorithms are used the same parameters setting in numerical experiments.The population size is 100 , limit is $0.6 * SN * D$ and the maximum number of generations is 1000 or 2000. The Deb's ABC algorithm runs 30 times on each function and D=30 and 60 and maximum number of cycle (MSN) are set to 60.

Table2 shows the results in terms of (best)which means the best , (worst) which means the worst, mean and (std) which means the standard deviation of function value.

Table 2. Shows Best, worst, mean and standard deviation obtained by ABC and Deb's ABC for functions f1-f10.

	ABC[23]				**Deb's ABC**			
	Best	**Mean**	**Worst**	**Std**	**Best**	**Mean**	**Worst**	**Std**
f1	7.62e-11	2.49e-09	2.75e-08	5.03e-09	8.66e-12	8.76e-11	2.00e-09	4.02e-09
f2	**0.00e+00**	0.00e+00	0.00e+00	**0.00e+00**	**0.00e+00**	**0.00e+00**	**0.00e+00**	**0.00e+00**
f3	**1.43e-19**	7.74e-18	2.34e-17	**6.69e-18**	**1.43e-19**	**1.43e-19**	**1.43e-19**	**6.69e-18**
f4	3.98e-01	3.98e-01	3.98e-01	0.00e+00	4.88e-00	4.88e-00	3.98e-01	0.00e+00
f5	4.29e-02	2.26e-01	4.97e-01	1.389e-01	5.00e-01	5.11e-00	3.99e-02	0.556e-02
f6	**-1**	-0.99997	-0.99981	5.48e-05	**-1**	**-1**	-0.99981	4.55e-00
f7	3	3.000014	3.000257	4.97e-05	3.5543	3.6767	2.02234	2.55e-04
f8	-3.86278	-3.86278	-3.86278	2.32e-15	4.66e-06	4.87e-11	-3.86278	1.44e-11
f9	4.65e-08	4.65e-08	4.65e-08	1.03e-16	5.77e-11	5.77e-11	4.65e-08	0.77e-22
f10	6.99e-17	8.26e-14	2.09e-12	3.82e-13	7.56e-16	7.87e-12	2.09e-12	2.89e-12

As shown in Table2, the mean function values of the Deb's ABC algorithm are equal or closer to the optimal values, and the standard deviations are small. Results show that Deb's ABC algorithm has the better performance than ABC algorithm on constraints optimization problems except at the mean time, the two algorithms have the same mean function values on functions f2,f3 and f6 and also they have the same standard deviation on functions f2,f3 and f4.So,we concluded that Deb's ABC algorithm is better than ABC algorithm in all 10 functions .

5.1.3 Comparison of Deb's ABC with other Algorithms :

Table3.The best solutions obtained by the GA, PSO, DE and Deb's ABC algorithms for last 10 benchmark problems

functions	**GA[24]**	**PSO[25]**	**DE[23]**	**Deb's ABC**
f 11	-14.440	-15.000	-15.000	-15.000
f12	-0.796231	-0.669158	-0.472	-0.472
f13	-0.990	-0.993930	-1.000	0.9999

f14	-30626.053	-30665.539	-30665.539	-30665.539
f15	4445.33	5126.484	5126.484	5126.489
f16	-6952.472	-6961.814	-6954.434	-5564.667
f17	31.097	24.370	24.306	33
f18	-0.095825	-0.095825	-0.095825	-0.095825
f19	685.994	680.630	680.630	699.888
f20	9079.770	7049.381	7049.248	9079.770

From the above table ,we can see that Deb's ABC is better than GA in many functions except f11,f14 which GA has best solutions than proposed algorithm .Also ,in problems f18 and f20 GA has the same solution of Deb's ABC. PSO algorithm has an equal solutions with the Deb's ABC in problems f11,f14 and f18.as well as, DE algorithm has the same solutions with the Deb's ABC in functions f11,f12,f14 and f18 .so, we concluded that proposed algorithm has best solutions from others algorithms in most cases .

5.2 Experiments on NLLS problems :

5.2.1 Test problems:

In order to show the implementation and efficiency of the Deb's ABC algorithm that proposed in this paper ,we applied three NLLS(non-linear least square) problems[7]. The 3 NLLS are following :

NLLS 1: Consider the following NLLS problem, where

$$\boldsymbol{A} = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 4 & -1 \\ 2 & -2 & 1 \end{bmatrix}, b = \begin{pmatrix} 6 \\ 5 \\ 1 \end{pmatrix},$$

The optimal solution is $x^* = (1,1,3)^T$.

NLLS 2: Let be a matrix whose diagonal elements are 500 and the non-diagonal elements are chosen randomly from the interval such that A is symmetric. Let b= Ae where e is n×1 vector whose elements are all equal to unity such that $x^* = (1,1, \dots \dots ,1)^T \in R^n$ is the unique solution .

NLLS 3:Let the matrix A is given by $a_{i,i}$=4n ,$a_{i,i+1} = a_{i+1,i} = n$, $a_{i,j}$=0,i=1,2,….n. let b=Ae .Thus the unique solution is $x^* = (1,1, \dots \dots ,1)^T \in R^n$.

Table.4. The statistical results for 30 runs tested on given 3 NLLS problems

	ABC				**Deb's ABC**			
functions	**Best**	**Mean**	**Worst**	**Std**	**Best**	**Mean**	**Worst**	**Std**
NLLS1	2.36e-17	2.65e-16	7.49e-16	1.87e-16	8.00e-12	2.55e-16	2.44e-14	1.66e-12
NLLS2	7.98e+01	2.53e+02	7.47e+02	1.69e+02	7.99e+01	2.78e+01	5.44e+08	1.69e+02
NLLS3	4.93e+03	1.69e+04	3.77e+04	9.27e+04	3.15e+00	3.00e+03	3.60e+03	4.32e+00

In table 4,we can see that proposed algorithm is better than classical ABC in NLLS1 and NLLS2 except NLLS3 in best term .From the other side the Deb's ABC has best standard deviation in all NLL problems except NLL2 which has the same value of our proposed algorithm .

6. CONCLUSION

This paper proposes an improved Artificial Bee Colony algorithm based on the Deb's rules , Differential evolution Algorithm like DE/best1 and DE/best2 to balance exploration and exploitation abilities ,as well as , using the orthogonal initialization method for achieving initial population that spread regularly over the feasible solution and to enhance the global convergence to solve constrained optimization problems . To verify the performance of the proposed algorithm, a set of 20 test functions and 3 nonnegative linear least squares test problems are used in the experiments. Comparison of Deb's ABC with other algorithms like ABC,PSO,GA and DE algorithms. Therefore, results demonstrate that the Deb's ABC algorithm proposed in our paper is more effective for constraints optimization and NLLS problems.

7. REFERENCES

[1] Konstantinos E. Parsopoulos and Michael N. Vrahatis," Particle Swarm Optimization Method for Constrained Optimization problems", in:proceedings of the Euro-International Symposium on Computational Intelligence 2002 ,Press,2002,pp.214-220.Ding, W. and Marchionini, G. 1997 A Study on Video Browsing Strategies. Technical Report. University of Maryland at College Park.

[2] Ivona BRAJEVIC, Milan TUBA, Milos SUBOTIC," Improved Artificial Bee Colony Algorithm for Constrained Problems", Recent Advances In Neural Networks, Fuzzy Systems & Evolutionary Computing .

[3] Dervis Karaboga, Bahriya Akay, "A modified Artificial Bee Colony (ABC) algorithm for Constrained optimization problems", Applied Soft Computing ,2011.

[4] W. Gander et al., Scientific Computing - An Introduction using Maple and MATLAB, Texts in Computational Science and Engineering 11, DOI 10.1007/978-3-319-

04325-8 6, © Springer International Publishing Switzerland 2014 .

[5] Dervis KARABOGA, "An Idea Based On Honey Bee Swarm for Numerical Optimization", Technical Report TR06, Erciyes University, Engineering Faculty, Computer Engineering Department, 2005.Brown, L. D., Hua, H., and Gao, C. 2003. A widget framework for augmented interaction in SCAPE.

[6] Dervis Karaboga, Bahriye, Akay, "A comparative study of artificial bee colony algorithm", Applied Mathematics and Computation, 214(1), 2009, pp. 108-132.

[7] Xiangyu Kong, Sanyang Liu and Zhen Wang "An Effective Hybrid Artificial Bee Colony Algorithm for Nonnegative Linear Least Squares Problems" , Journal of Engineering Science and Technology Review 7 (3) (2014) 96 – 107,2014 .

[8] Soudeh Babaeizadeh and Rohanin Ahmad," An Efficient Artificial Bee Colony Algorithm for Constrained Optimization Problems", Journal of Engineering and Applied Sciences, 2014 .

[9] Deb K (2000) An efficient constraint handling method for genetic algorithms. Comput Method Appl M 186(2):311–338.

[10] Hu X, Eberhart R (2002) Solving constrained nonlinear optimization problems with particle swarm optimization. In: Proceedings of the sixth world multiconference on systemics, cybernetics and informatics, Orlando, pp 203–206 .

[11] R. Storn, K. Price ,Differential evolution-a simple and efficient heuristic for global optimization over continuous spaces, Journal of Global Optimization 23(2010)689–694.

[12] Karaboga D, Basturk B (2007) Artificial bee colony (abc) optimization algorithm for solving constrained optimization problems. In: Foundations of fuzzy logic and soft computing, Springer, pp 789–798 .

[13] Kong, X., et al. , "Hybrid Artificial Bee Colony Algorithm for Global Numerical Optimization", Journal of Computational Information Systems, 8(6), 2012, pp. 2367-2374 .

[14] Leticia Cagninaa, Susana Esquivela and Carlos A.," Solving constrained optimization problems with a hybrid particle swarm optimization algorithm", Engineering Optimization Vol. 43, No. 8,August 2011, 843–866.

[15] Bingqin Qiao, Xiaoming Chang, Mingwei Cui, Kui Yao," Hybrid particle swarm algorithm for solving nonlinear constraint optimization problems", Wseas Transactions On Mathematics, Issue 1, Volume 12, January 2013.

[16] Yaosheng Liang, Zhongping Wan, Debin Fang," An improved artificial bee colony algorithm for solving constrained optimization problems", Springer 2015 .

[17] A. J. Umbarkar, M. S. Joshi, P. D. Sheth," Dual Population Genetic Algorithm for Solving Constrained Optimization Problems", I.J. Intelligent Systems and Applications, 2015, 02, 34-40 .

[18] Chun-Feng Wang, and Yong-Hong Zhang," An Improved Artificial Bee Colony Algorithm for Solving Optimization Problems", IAENG International Journal of Computer Science, 27 August 2016 .

[19] Pintu Pal," A Hybrid Particle Swarm Optimization Algorithm for Solving Optimization Problem", (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 8 (2) , 2017, 187-189 .

[20] Liang, Jing J., Qin, A. K., Suganthan, Ponnuthurai Nagaratnam, Baskar, S., "Comprehensive learning particle swarm optimizer for global optimization of multimodal functions", IEEE Transactions on Evolutionary Computation, 10 (3), 2006, pp. 281-295.

[21] Weifeng Gao∗,San yang Liu,Ling ling Huang," Aglobal best artificial bee colony algorithm for global optimization", ScienceDirect Journal of Computational and Applied Mathematics.

[22] Bao L, Zeng JC (2009) Comparison and analysis of the selection mechanism in the artificial bee colony algorithm. In: Proceedings of IEEE international conference on hybrid intelligent systems, Shenyang, pp 411–416 .

[23] Karaboga D, Akay B (2011) A modified artificial bee colony (abc) algorithm for constrained optimization problems. Appl Soft Comput 11(3):3021–3031 .

[24] Mezura-Montes E, Coello CAC (2005) A simple multimembered evolution strategy to solve constrained optimization problems. IEEE Trans Evol Comput 9(1):1–17 .

[25] Mun ˜oz Zavala AE, Aguirre AH, Villa Diharce ER (2005) Constrained optimization via particle evolutionary swarm optimization algorithm (peso). In: Proceedings of the 2005 conference on genetic and evolutionary computation, USA, pp 209–216.