ISSN:0975-9646

Gajendra Shrimal et al, / (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 5 (4) , 2014, 5651-5658

A Hybrid Best So Far Artificial Bee Colony Algorithm for Function Optimization

Gajendra Shrimal[#1], Rakesh Rathi[#2]

[#]1M.Tech. Pursuing RTU, [#]2Asst. Professor (GEC, Ajmer)

Abstract—**Artificial Bee Colony Algorithm (ABC) is the most recent advance technique to solve many mathematical problems and engineering problems. The inspiration behind this is Nature, where problems are solved on the basis of behaviour of swarms, ants, bees etc. The foraging behaviour of honey bees plays an important role while approaching ABC algorithms. This paper introduces a new hybrid approach to enhance the performance of original ABC algorithm. In this a composition algorithm is introduced where Best-so-far ABC and Golden section search algorithms are hybrid together to improve the success rate of ABC algorithm. Best-so-far was modified version of Original ABC where exploitation and exploration both process were modified, and in golden section search a function can be optimized in a given range with the a parameter called as golden ratio. The proposed algorithm is named as BSFMeABC i.e. Best-so-far based on memetic search. This algorithm is tested over some benchmark functions and some well known engineering problems. On the basis of feature like success rate, mean function evaluation, acceptable error etc. a comparison chart is made which show the performance evaluation of these algorithms.**

Keywords— Artificial Bee Colony Algorithm, Golden Section Search, Evolutionary Computation, Particle Swarm Optimization, Swarm Intelligence, , Memetic Search.

I. INTRODUCTION

Nature-inspired computing (NIC) is the well known methodology provided the basis for the solution of many engineering and complex problems. The inspiration behind this is Nature, where problems are solved on the basis of behaviour of swarms, ants, bees etc. A comparison can be seen while looking for a better optimization solution where a Natural inspired algorithm gives better results [1].

In recent years, swarm intelligence becomes a crucial importance for the solution of many problems which cannot be easily solved with many classical mathematical techniques. The main concern while searching for new nature based solution is population. The collective behaviour of swarm's ants etc or any individuals inspire us to develop optimization-based algorithm. To find near optimal solution to the complex mathematical problems or any engineering problems, population-based optimization algorithms are developed which works on fitness (nectar) evaluation and therefore the population of potential solutions is expected to move towards the better fitness areas of the search space. Population-based optimization algorithms are majorly categorized into two categories first is Swarm intelligence [5] based algorithms and second is Evolution [6] based algorithms.

Artificial Bee Colony Algorithm (ABC) is the most recent advance technique to solve many mathematical problems and engineering problems. The inspiration behind this is Nature, where problems are solved on the basis of behaviour of swarms, ants, bees etc. The foraging behaviour of bees plays an important role while approaching ABC algorithms. Table I show some important feature of ABC algorithms.

TABLE I. FEATURE OF ABC

Feature	Description
Simplicity	Simple and easy to design strategies
Richness	Obtain solution are rich in nature
Flexibility	Flexible enough to modify and develop new algorithms
Robustness	Generated solution are robust in nature
Lesser control parameters	Few control parameters reduces complexity
Nature inspired	Based on foraging behavior of honey bees
Easy Implementation	As having fewer parameters, it becomes easy to implement it.
Easy Hybridization	It can be easily mixed with other algorithm

Recently D. Karaboga proposed a new approach, which was easy to implement. The inspiration behind this was honey bees. Honey bees looks for outstanding food sources near the common food source available near to the solution area. This was termed as Artificial Bee Colony Algorithm [7]. This was very similar to many of the other population based optimization algorithms; this algorithm also has a population of promising solutions. Food source actually represents one of the possible solutions for a honey bee. Here a quality parameter is calculated for one of the food source solutions, which is also termed as Fitness of the specific solution.

Exploration and exploitation are the two issues concerns with the performance of ABC algorithm, which has to be uniform and study feasible solution. Sometimes it is observed that the ABC stops proceeding headed for the global optimum despite the fact that the local optimum not achieved [8]. Some research revealed that the position update equation for ABC technique is fine at exploration however it is not good at exploitation [9]. It is exceedingly important to widen a local search policy in the fundamental ABC in order to exploit the search space so that balance between intensification and diversification can be maintained. In this phase a new technique was discussed to modify the exploitation and exploration process of the original ABC algorithm, termed as BSFABC [10]. This work is proposed to make hybrid approach by combining best properties of two approaches BSFABC [10] and properties of golden section search to enhance the performance of optimization.

This paper is planned in six sections. First will give a brief overview, second section is planned to ABC original algorithm, section three is for recent modified ABC algorithms, section four is used for the proposed work, section five will brief the results comparison, and last section six is used for references to conclude.

II. ARTIFICIAL BEE COLONY ALGORITHM

The ABC algorithm that is motivated by extraordinary food foraging conduct of honey bee insects is very simple to understand and implement. Each food source for honey bee symbolizes solution of a particular problem in ABC algorithm. Fitness of a particular food source computes its quality that represent amount of nectar in a food source. In ABC algorithm, honey bees are categorized into three sets that is to say employed bees, onlooker bees and scout bees. The employed bees and the onlooker bees must be same in quantity. The employed bee search new food sources and gather information about the eminence of the food sources. Some bees stay in the beehive and observe the activities of employed bees. Based on the activities of employed bees they select food sources are identified as onlooker bees. When a food source rejected due to low quality, then they are replaced by new food sources randomly. The ABC strategy follows iterative process it repeats these three phase [8] again and again. Each of the phases is illustrated as follows: First phase is to propel the employed bees on the food sources, modernize position of food sources based on quality of particular food source; second phase onlooker bees select a food source with higher probability based on its fitness. Third phase engender randomly new food sources in place of rejected food sources.

These steps can be summarized into following sections.

1) Initialization of population

The ABC generated a finite distributed collection of solutions. Each solution can be termed as x_i, where each x_i is a D Dimension vector.

$$x_{ij} = x_{\min j} + rand[0,1](x_{\max j} - x_{\min j}) \qquad (1)$$

D. Karaboga [8] suggested that the quantity of food sources should be the sum of to the employed bees and onlooker bees. At the time of initialization it is considered that food sources (SN) are evenly dealt swarm, where a D-dimensional vector represent each food source x_i (i = 1, 2 ...SN). Each food source is initialized using Eq. (1) [8]:

Where

- rand[0,1] is a function that engender an equally dispersed arbitrary numeral in range [0,1].

2) Employed Bee Phase

Food source are exploited with the help of these. The position of current solution modernized according to knowledge of individual food source and available nectar in a particular flower. Now this knowledge is used to compare with two food sources and then on the basis of that comparison old food sources is replaced with new food source if the with higher fitness value. Actually fitness value is the measurement of nectar amount or food source's strength. The position of j^{th} dimension of i^{th} candidate modernizes using Eq. (2) [8]:

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

Where

- x_{ij}-x_{kj} decide size of step,
- $k \in \{1, 2, ..., SN\}$, $j \in \{1, 2, ...,D\}$ are two indices that are haphazardly preferred in such a way that $k \neq i$ in order to make sure that step size has some pinpointing enhancement.

3) Onlooker Bee Phase

One of unemployed bee is onlooker; counting of onlooker bees is identical to the quantity of employed bees. During this segment all employed bee share quality of new food sources with onlooker bees in form of fitness. Each food source judged based on it probability of selection. The highly fitted solution gets elected by the onlooker. There are various techniques for calculation of probability; however it must be a function of fitness. Probability of selection for each food source is determined with its fitness as per Eq. (3) [8]:

$$P_{ij} = \frac{fit_i}{\sum_{i=1}^{SN} fit_i} \qquad (3)$$

4) Scout Bee Phase

In case when the position of a particular food source is not modernized for a threshold (in term of number of cycles), that food source is derelict and a new phase starts named scout bees phase. The bees that are allied with the deserted food source transformed into scout bee and the food source is substituted by the capriciously elected food source within the search space. New food sources generated using Eq. (4) [8].

$$x_{ij} = x_{\min j} + rand[0,1](x_{\max j} - x_{\min j}) \qquad (4)$$

All these steps can be summarised in following algorithm 1

Algorithm 1: Artificial Bee Colony Algorithm

Initialize all parameters for population strength; xi (i = 1, 2 ...SN).
Repeat while curr_cycle not reaches to max_cycle
Step 1: For getting new solutions
Perform the Employed bee phase
Step 2: apply greedy process on employed bees
Step 3: Calculate the probability for each new solution xi
Step 4: for getting new solution (food source)
Perform onlooker bee phase for x_i solution on the basis of generated probability.
Step 5: for probing new food sources
Perform Scout bee phase for probing new food sources in place of discarded food sources.
Step 6: Memorize the finest food source known up to now.
End of while
Output: The finest solution recognized up to now.

III. RECENT MODIFIED ABC ALGORITHMS

Artificial Bee Colony Algorithm (ABC) is the most recent advance technique to solve many mathematical problems and engineering problems. The inspiration behind this is Nature, where problems are solved on the basis of behavior of swarms, ants, bees etc. The foraging behaviour of bees plays an important role while approaching ABC algorithms.

There are three main control parameters of the ABC algorithm: Number of food sources, “limit” and Φ ij (a uniformly distributed random number in the range [-1,1]).

In recent year it has fascinated many of the researchers to work on it. A number of researchers are working on ABC algorithm to solve optimization problems.

Since the origination of ABC, a lot of research has been carried out to increase the performance of ABC. These researches are based on various facts. These include introducing new strategies and finely modify the control parameters, introducing new control parameters, and many more. One of the most fascinated solutions is hybridization with other existing algorithms.

A shifted neighboured search was proposed by Baykasoglu et al.[11], which used the greedy randomized adaptive methodology. It was then applied on the generalized assignment problem.

A hybrid bee algorithm for solving container loading problems [12] was proposed by T. Derelia and G.S. Das in Applied Soft Computing, 2010, hybridized with the heuristic filling procedure for the solution of container loading problems.

In 2010, Huang and Lin [13] proposed a new bee colony optimization algorithm with idle-time-based filtering scheme and its application for open shop-scheduling problems. They categorised the foraging behaviours of bees in two terms Forward Pass and Backward Pass. Forward Pass expresses the process of a forager bee leaving the bee hive and flying towards a food source while Backward Pass denotes the process of a forager bee returning to the bee hive and sharing the food source information with other forager bees (role change).

In 2011, Nambiraj Suguna et al. [14] proposed an independent rough set approach hybrid with artificial bee colony algorithm for dimensionality reduction. In the proposed work, effects of the perturbation rate, the scaling factor (step size), and the limit are investigated on real-parameter optimization. In 2012, Bin Wu et al. [15] proposed improvement of Global swarm optimization (GSO) hybrid with ABC and PSO. They use neighbourhood solution generation scheme of ABC and accept new solution only when it is better than previous one to improve GSO performance.

In 2012 Proposed a Hybrid ABC (HABC) algorithm by introducing the crossover operator of Genetic Algorithm (GA)to ABC in information exchange (social learning) phase between bees for data clustering [16].

In 2012 proposed hybridization of ABC with Harmonic search algorithm called collaborative ABC algorithm (C-ABC) for adapting the connection weights, network architecture, the features of time series input data and the learning algorithms according to the problem environment [17].

In 2011, Anan Banharnsakun et al. [18] proposed a new approach termed as best-so-far selection in artificial bee colony algorithm. This was proposed to enhance both exploitation explorations. According to the author three changes were made to enhance the performance of the ABC algorithm. In order to improve the efficiency of the onlooker bees some parameter were modified. In case of ABC algorithm, each onlooker bee selects the food source on the basis of probability that varies according to the fitness function explored by single bee, but here this depends not on a single bee. The exploration is done by all employed bees, means a best solution is explored by all employed bees. It also modifies the exploitation feature by applied an adjustable radius search. This adjustable radius search was used for scout bees where scout bee searches for the next solution to be explored. They modified the scout bee generation equation 4 to add some new parameters ω with a Range from 1.0(maximum) to 0.2(minimum). In [33] an improved memetic search is also employed to artificial bee colony algorithm

IV. Modified BSFABC Hybrid With Golden Section Search

In 2011, Anan Banharnsakun et al. [18] proposed a new approach termed as best-so-far selection in artificial bee colony algorithm. To enhance the exploitation and exploration processes, he had three changes in that.

First was for improving onlooker bee performance. In this they modify the parameters that were involved in updating of the position of a onlooker bee. In case Original ABC, the position updating on fitness value of the solution, and fitness were calculated on the basis of single employed bee. Here it changes this, by adding a new parameter “best solution”. The best solution is calculated by processing the information received from all employed bees. Then it calculates fitness based on that particular best solution, termed as $Fitness_b$ then this is compare with other solutions’ fitness. It $Fitness_b$ finds itself higher then other’s fitness; No position is updated. Else position is updated with new solution. This step can be summarizes in the following equation.

$$v_{id} = x_{ij} + \emptyset f_b (x_{ij} - x_{bj}) \quad (5)$$

where: v_{id} = The new candidate food source for onlooker bee position i dimension d, d = 1,2,3,. . .D; x_{ij} = The selected food source position i in a selected dimension j; Φ is a random number between −1 and 1; f_b = The fitness value of the best food source so far; x_{bj} = The best-so-far food source in selected dimension j.

Second change was made in the adjustable search radius. This is done because of the need to get out from the local optimum solution problem.

$$v_{ij} = x_{ij} + \emptyset_{ij} \left[wmax - \left(\frac{iteration}{MCN} \right) (wmax - wmin) \right] x_{ij} \quad (6)$$

where v_{id} is a new feasible solution of a scout bee that is modified from the current position of an abandoned food source (x_{ij}) and Φ_{ij} is a random number between [−1, 1]. The value of wmax and wmin represent the maximum and minimum percentage of the position adjustment for the scout bee. The value of wmax and wmin are fixed to 1 and 0.2, respectively. These parameters were static. Means here they were proposed a specific searching range of 100 to 20 in percentage.

The third change is finding the minimum objective value, here compare and to select between the old solution and the new solution in each iteration. Basically, the comparison of the new solution and the old solution is done by the fitness value. If the fitness of the new solution is better than the fitness of the old solution, it select the new one and ignore the old solution. The fitness value can be obtained from the following equation.

$$Fitness(f(x)) = \begin{cases} \frac{1}{1+f(x)} & if\ f(x) > 0 \\ 1 + |f(x)| & if\ f(x) < 0 \end{cases} \quad (7)$$

Fister et al. proposed a memetic ABC for Large- Scale Global Optimization [19]. In the proposed approach, ABC is hybridized with two local search heuristics: the Nelder-Mead algorithm (NMA) and the random walk with direction exploitation (RWDE) [20]. The former is attended more towards exploration, while the latter more towards exploitation of the search space. The stochastic adaptive rule as specified by Neri [21] is applied for balancing the exploration and exploitation. In 2013, Jagdish Chand Bansal et al. proposed a new approach based on golden section search [22], this search technique is applied on abc original algorithm.

Neri et al. proposed an unconventional memetic computing strategy for solving continuous optimization problems characterized by memory limitations [23]. The proposed algorithm, unlike employing an explorative evolutionary framework and a set of local search algorithms, employs multiple exploitative searches within the main framework and performs a multiple step global search. The proposed local memetic approach is based on a compact evolutionary framework. Iacca et al, proposed a counter-tendency approach for algorithmic design for memetic computing algorithms [24]. Further Kang et al. described a Rosenbrock ABC (RABC) that combines Rosenbrock's rotational direction method with ABC for accurate numerical optimization [25]. In RABC, exploitation phase is introduced in the ABC using Rosenbrock's rotational direction method.

The golden section search algorithm is used to find the optimization value of given function. Original GSS approach finds the optima of a unimodal continuous function without using any gradient information of the function. GSS processes the interval [a= -1.2, b = 1.2] and generates two intermediate points:

F1 = b – (b-a) * ψ and F2 = a – (b-a) * ψ

Where ψ= 0.618 is the golden ratio. The pseudo-code of GSS algorithm is shown in Algorithm 2:

Algorithm 2: Golden Section Search Process

Input Optimization Function Min(f(x)) and Initialize a= -1.2 and b = 1.2, where f(x) is subjected to a≤ x ≤ b

Repeat until terminate condition occurs.

Calculate F1 = b – (b-a) * ψ and F2 = a – (b-a) * ψ

Also calculate f(F1) and f(F2)

If f(F1)<f(F2) then

b=F2 solution lies in [a,b]

Else

a=F1 solution lies in [a,b]

End if

End loop

In our proposed work hybridization the BSFABC with this golden section search algorithm is proposed. The idea behind this is to merge the best features exists in both the algorithm, and generate a new algorithm based on these two, which have capabilities of these two algorithms.

Two major changes in this hybrid process are done. As BSFABC modifies both exploration and exploitation properties and golden section gives us the optimized value of given function in constrained environment, we need to be careful while hybridizing both of them.

First change done here is, in the scout bee phase. In BSFABC a control parameters wmax and wmin were used to enhance the search radius. The value of wmax and wmin represent the maximum and minimum percentage of the position adjustment for the scout bee. The value of wmax and wmin are fixed to 1 and 0.2, respectively. These parameters were static. Means here they were proposed a specific searching range of 100 to 20 in percentage. Here an adaptive method is used to find the values of wmax and wmin, which will bring us the radius limits for the search space. As in employed bee phase, next food source chosen by onlooker bee is decided by best solution strategy; similarly this bestfitness can also find us the better near new solution for the scout bees. This can lead us to explore the new solutions near best solutions values. Here modify the values of wmax and wmin according to the fitness of best solution with the given modification.

If the best fitness is greater than upcoming fitness of the x_i this range can be reduced so that we can get optimum solution only near to the best solution we have already found. In other case this bound is increased. The assumption behind this is that we can find new solutions (more nectar amount values) near the already find best solution as compare to search in bigger range or bigger search radius. This can be seen in the following equation.

if (bestfitness > fitness_abc[i]){
reducing the wmax-wmin range by having wmax and wmin such as 0.4and 0.1 respectively.
}
else {
increases the wmax-wmin range by having wmax, wmin such as 1.0 and 0.4 respectively.
} (8)

Second change we have done here is to apply a golden section search methodology on this modified Best-so-far ABC.

This can be implemented with following algorithm.

Algorithm 3: Golden Section Search Algorithm

Initialize a= -1.2 and b = 1.2

Repeat while (|a-b| < ϵ)

Compute $f_1 = (b - (b-a)*\Psi)$

$f_2 = (a + (b-a)*\Psi)$

Generate two new solutions X_{new1} and X_{new2} using f_1 and f_2 respectively according to MeABC

Calculate $f(X_{new1})$ and $f(X_{new2})$ for objective function

if $(f(X_{new1}) < f(X_{new2}))$ then $b = f_2$

if $(f(X_{new1}) < f(X_{best}))$

then $X_{best} = X_{new1}$

else $a = f_1$

if $(f(X_{new2}) < f(X_{best}))$

then $X_{best} = X_{new2}$

The newly introduced local search phase (As shown in algorithm 3) improve exploration process as adaptive search in artificial bee colony algorithm is good in exploration of local search space.

So a hybrid approach of BSFABC (exploration and exploitation) and golden section search (exploration) can deliver better results. The algorithm representation of this approach can be seen as in algorithm 4.

Algorithm 4: Memetic search with BSFABC

Initialize all parameters;
Repeat until iteration reaches to a max level
 Step 1: for employed bees
  Perform the Employed bee phase for compute new food sources as seen in BSFABC
 Step 2: for updating the onlooker bees position
  Perform the Onlooker bees phase for updating position the food sources based on their amount of nectar using equation (5).
 Step 3: Perform modified Scout bee phase for searching new food sources in place of abandoned food sources according to condition (8).
 Step 4: Apply golden section search using algorithm 3.
End of loop
Output: The best solution recognized so far.

V. EXPERIMENTAL RESULTS

A. *Test problems*

Experiment result here are consists of two phases. First well known benchmarks are used for the optimization process and in second phase we have introduced some complex engineering problem to this approach.

It gives improved outcome or not at diverse probability and also applied for two factual world problems namely compression spring problem and welded beam design problem. Benchmark problems considered in this paper are of different individuality like uni-model or multi-model and separable or non-separable and of diverse dimensions. So as to analyse the performance of MGABC it is applied to global optimization problems (f_1 to f_{14}) outlined in Table I. Test problems f_1 –f_{13} are taken from [26][27].

In this we have also included used some well known engineering problems like Lennard-Jones problem, compression string, Welded beam design optimization problem

TABLE I. TEST PROBLEMS

Test Problem	Objective Function	Search Range	Optimum Value	D	Acceptable Error
Griewank	$f_1(x)=\frac{1}{4000}\left(\sum_{i=1}^{D}(x_i^2)\right)-\left(\prod_{i=1}^{D}\cos\left(\frac{x_i}{\sqrt{i}}\right)\right)+1$	[-600, 600]	*f(0) = 0*	30	1.0*E*-05
Rastrigin	$f_2(x)=\sum_{i=1}^{D}\left[x_i^2-10\cos(2\pi x_i)+10\right]$	[-5.12, 5.12]	*f(0) = 0*	30	1.0*E*-05
Alpine	$f_3(x)=\sum_{i=1}^{n}\lvert x_i\sin x_i+0.1x_i\rvert$	[-10, 10]	*f(0) =0*	30	1.0*E*-05
Zakharov	$f_4(x)=\sum_{i=1}^{D}x_i^2+\left(\sum_{i=1}^{D}\frac{ix_i}{2}\right)^2+\left(\sum_{i=1}^{D}\frac{ix_i}{2}\right)^4$	[-5.12, 5.12]	*f(0) =0*	30	1.0*E*-02
Salomon Problem	$f_5(x)=1-\cos(2\pi\sqrt{\sum_{i=1}^{D}x_i^2})+0.1(\sqrt{\sum_{i=1}^{D}x_i^2})$	[-100, 100]	*f(0) = 0*	30	1.0*E*-01
Inverted Cosine wave function	$f_6(x)=-\sum_{i=1}^{D-1}(\exp(\frac{-(x_i^2+x_{i+1}^2+0.5x_ix_{i+1}}{8},$ $Where.I=\cos(4\sqrt{x_i^2+x_{i+1}^2+0.5x_ix_{i+1}})$	[-5. 5]	*f(0) = -D+1*	10	1.0*E*-05
Neumaier 3 Problem (NF3)	$f_7(x)=\sum_{i=1}^{D}(x_i-1)^2-\sum_{i=2}^{D}x_ix_{i-1}$	[-100, 100]	*f(0) = -210*	10	1.0*E*-01
Colville function	$f_8(x)=100(x_2-x_1^2)^2+(1-x_1)^2+90(x_4-x_3^2)^2+(1-x_3)^2$ $+10.1[(x_2-1)^2+(x_4-1)^2]+19.8(x_2-1)(x_4-1)$	[-10, 10]	*f(1) = 0*	4	1.0*E*-05
Kowalik function	$f_9(x)=\sum_{i=1}^{11}(a_i-\frac{x_1(b_i^2+b_ix_2)}{b_i^2+b_ix_3+x_4})^2$	[-5, 5]	*f(0.1928, 0.1908, 0.1231, 0.1357) = 3.07E-04*	4	1.0*E*-05
Shifted Rosenbrock	$f_{10}(x)=\sum_{i=1}^{D-1}(100(z_i^2-z_{i+1})^2+(z_i-1)^2+f_{bias},$ $z=x-o+1, x=[x_1,x_2,...x_D], o=[o_1,o_2,.......o_D]$	[-100, 100]	*f(o)=f_{bias}=390*	10	1.0*E*-01
Shifted Griewank	$f_{11}(x)=\sum_{i=1}^{D}\frac{z_i^2}{4000}-\prod_{i=1}^{D}\cos(\frac{z_i}{\sqrt{i}})+1+f_{bias},$ $z=(x-o), x=[x_1,x_2,...x_D], o=[o_1,o_2,.......o_D]$	[-600, 600]	*f(o)=f_{bias}=-180*	10	1.0*E*-05
Hosaki Problem	$f_{12}(x)=(1-8x_1+7x_1^2-\frac{7}{3}x_1^3+\frac{1}{4}x_1^4)x_2^2\exp(-x_2)$	$x_1\in[0,5]$, $x_2\in[0,6]$	*-2.3458*	2	1.0*E*-06
Meyer and Roth Problem	$f_{13}(x)=\sum_{i=1}^{5}(\frac{x_1x_3t_i}{1+x_1t_i+x_2v_i}-y_i)^2$	[-10, 10]	*f(3.13, 15.16,0.78) = 0.4E-04*	3	1.0*E*-03

The proposed hybrid of best-so-far and memetic search algorithm is produced. The results produced are compared with original ABC algorithm, BSFABC algorithm, and Memtic ABC algorithm. This hybrid approach is tested over above discussed problem with the following environment settings.

- No. Of colony size (employed bees+ onlooker bees) = Population size SN = 50
- Number of Employed bee = Number of Onlooker bee =SN/2 = 25
- The maximum number of cycles for foraging MCN = 100000
- Number of repetition of experiment =Runtime =100
- Limit = D x SN, for the results which cannot improve further, after the limit they are abandoned for employed be, Where D stands for dimension.

The results includes mean function values (MFV), standard deviation (SD), mean error (ME), average function evaluation (AFE) and success rate (SR), also they are compared with other above discussed algorithms.

B. *Compared Results*

Results produced by this hybrid approach with above discussed settings are shown in Table II. Mathematical results of MGABC with experimental setting as per previous subsection are outlined in Table II. The results includes mean function values (MFV), standard deviation (SD), mean error (ME), average function evaluation (AFE) and success rate (SR), also they are compared with other above discussed algorithms. Table III are summary report of the above Table II, where a "+" sign indicates a significant improvement and "-" represent failure with a small variant. In table IV the acceleration rates are compared.

Lennard-Jones problem is approximation methodology that demonstrates the energy of interaction between two nonbonding atoms or molecules based. The idea behind this is the distance they in their separation. This is discussed by Clerc m.et al. [29].

The function to minimize is a kind of potential energy of a set of N atoms. The position X_i of the atom i has three coordinates, and therefore the dimension of the search space is 3N. In practice, the coordinates of a point X are the concatenation of the ones of the Xi. In short, we can write X = (X_1, X_2, ..., X_N), and we have then.

$$E_1 = \sum_{i=1}^{N-1}\sum_{j=i+1}^{N}\left(\frac{1}{||X_i-X_j||^{2\alpha}} - \frac{1}{||X_i-X_j||^{\alpha}}\right) \quad (9)$$

In this study N = 5, a = 6, and the search space is [2, 2]

Compression spring another engineering optimization application is compression spring problem discussed by Onwubolu et al.[30] and Sandgren.[31] This problem minimizes the weight of a compression spring, subject to constraints of minimum deflection, shear stress, surge frequency, and limits on outside diameter and on design variables. There are three design variables: the wire diameter x1, the mean coil diameter x2, and the number of active coils x3. This is a simplified version of a more difficult problem.

In case of compression spring three design variables considered: The diameter of wire(x_1), mean coil diameter (x_2) and count of active coils (x_3). Simple mathematical representation of this problem is:

$$x_1 \in \{1,2,3,...............,70\} granularity_1$$

$$x_2 \in [0.6;3], x_3 \in [0.207;0.5] granularity_{0.001}$$

And four constraints

$$g_1 := \frac{8c_f F_{\max} x_2}{\pi x_3^3} - S \le 0, g_2 := l_f - l_{\max} \le 0$$

$$g_3 := \sigma_p - \sigma_{pm} \le 0, g_4 := \sigma_w - \frac{F_m ax - F_p}{K} \le 0$$

$$\text{Where}: c_f = 1 + 0.75\frac{x_3}{x_2 - x_3} + 0.615\frac{x_3}{x_2}, F_{\max} = 1000,$$

$$S = 189000, l_f = \frac{F_{\max}}{K} + 1.05(x_1+2)x_3, l_{\max} = 14, \sigma_p = \frac{F_p}{K},$$

$$\sigma_{pm} = 6, F_p = 300, K = 11.5\times10^6 \frac{x_3^4}{8x_1x_2^3}, \sigma_w = 1.25$$

And the function to be minimized is

$$f_{cs}(X) = \pi^2 \frac{x_2 x_3^2 (x_1+2)}{4}$$

The best ever identified solution is (7, 1.386599591, 0.292), which gives the fitness value f =2.6254 and 1.0E-04 is tolerable error for compression spring problem.

Welded beam design optimization problem: It is a problem of designing a welded beam with minimum cost [32]. Here it is required to identify the minimum fabricating cost of the welded beam subject to constraints on bending stress σ, load of buckling P_c, end deflection δ, shear stress τ, and side constraint. In case of this problem four design variables are considered: x_1, x_2, x_3 and x_4. The simple mathematical formulation of the objective function is described as follows:

$$f(x) = 1.1047x_1^2x_2 + 0.04822x_3x_4(14.0 + x_2)$$

Subject to:

$$g_1(x) = \tau(x) - \tau_{\max} \le 0, g_2(x) = \sigma(x) - \sigma_{\max} \le 0,$$

$$g_3(x) = x_1 - x_4 \le 0, g_4(x) = \delta(x) - \delta_{\max} \le 0,$$

$$g_5(x) = P - P_c(x) \le 0$$

$0.125 \le x_1 \le 5$, $0.1 \le x_2, x_3 \le 10$ and $0.1 \le x_4 \le 5$

Where

$$\tau(x) = \sqrt{\tau'^2 - \tau'\tau''\frac{x_2}{R} + \tau''^2}, \tau' = \frac{P}{\sqrt{2x_1x_2}}, \tau'' = \frac{MR}{J},$$

$$M = P(L + \frac{x_2}{2}), R = \sqrt{\frac{x_2^2}{4} + (\frac{x_1+x_2}{2})^2}, \sigma(x) = \frac{6PL}{x_4x_3^2},$$

$$J = \frac{2}{\sqrt{2}x_1x_2[\frac{x_2^2}{4} + (\frac{x_1+x_2}{2})^2]}, \delta(x) = \frac{6PL^3}{Ex_4x_3^2},$$

$$P_c(x) = \frac{4.013Ex_3x_4^3}{6L^2}(1 - \frac{x_3}{2L}\sqrt{\frac{E}{4G}})$$

P=6000 lb, L=14 in, δ_{max} = 0.25 in, σ_{max} = 30000 psi, τ_{max} = 13600 psi, E = 30*10^6 psi, G= 12*10^6 psi

The best known solution is (0.205730, 3.470489, 9.036624, 0.205729), which gives the function value 1.724852. Acceptable error for this problem is 1.0E-01.

VI. CONCLUSION

In this paper a new hybrid approach based on Best-So-FAR ABC and Golden Section Search is developed. We have also modified the original BSFABC, in its scout bee phase; and tested this hybrid approach on various benchmark function and engineering problems.

The results generated are far better as compared to other algorithms discussed; we have also some cases where other algorithms performed better than this. As result summary shows in most of the cases this hybrid algorithm's performance was better as compare to others.

A result also compared using the convergence rate. Higher the AFE (Average Function Evaluations), higher the convergence rate and lower the AFE, lower the convergence rate. The Acceleration Rate can be defined as

AR = AFEALGO/AFEMGABC

Here

$AR1 = AFE_{ABC} / AFE_{BSFMeABC}$

$AR2 = AFE_{BSFABC} / AFE_{BSFMeABC}$

$AR3 = AFE_{MeABC} / AFE_{BSFMeABC}$

TABLE II. RESULT COMPARISON

Test Problem	Algo	Mean Fun Val	SD	Error	Total Mean Fun Eval	SR
Griewank	ABC	4.36E-03	7.51E-03	4.36E-03	76412	68
	BSFABC	1.25E-03	4.07E-03	1.25E-03	61601	89
	MeABC	7.96E-04	2.78E-03	7.96E-04	69791.56	92
	BSFMeABC	2.30E-04	1.64E-03	2.30E-04	49244.18	98
Rastrigin	ABC	3.41E+00	1.58E+00	3.41E+00	99490	2
	BSFABC	8.41E-01	5.80E-01	8.41E-01	99959	2
	MeABC	4.82E-01	5.71E-01	4.82E-01	98320.46	18
	BSFMeABC	4.01E-02	1.95E-01	4.01E-02	97958.3	47
Alpine	ABC	2.26E-02	1.51E-02	2.26E-02	100000	0
	BSFABC	1.18E-03	1.69E-03	1.18E-03	99991.5	1
	MeABC	6.89E-03	4.27E-03	6.89E-03	100048	0
	BSFMeABC	2.36E-04	2.35E-04	2.36E-04	99921.46	2
Zakharov	ABC	9.48E+01	1.44E+01	9.48E+01	100000	0
	BSFABC	1.22E+02	1.43E+01	1.22E+02	100000	0
	MeABC	9.96E-03	1.04E-03	9.96E-03	83766.72	84
	BSFMeABC	1.25E-02	4.81E-03	1.25E-02	93508.78	45
Salomon Problem	ABC	1.57E+00	2.51E-01	1.57E+00	99873.3	1
	BSF ABC	1.92E+00	2.19E-01	1.92E+00	100002.01	0
	MeABC	9.27E-01	3.21E-02	9.27E-01	24194.47	100
	BSFMeABC	9.20E-01	2.76E-02	9.20E-01	25889.21	100
Inverted Cosine	ABC	-2.44E+00	4.85E-01	6.56E+00	100010.43	0
	BSFABC	-8.86E+00	2.97E-01	1.36E-01	70011.56	68
	MeABC	-8.82E+00	4.02E-01	1.81E-01	72603.35	67
	BSFMeABC	-8.95E+00	1.95E-01	4.99E-02	44706.13	93
Neumaier 3 Problem	ABC	-5.76E+01	2.47E+01	1.52E+02	100037.76	0
	BSFABC	-2.03E+02	2.20E+01	6.97E+00	99447.98	3
	MeABC	-2.10E+02	1.09E-02	8.93E-02	23383.57	100
	BSFMeABC	-2.10E+02	9.72E-03	9.06E-02	24535.91	100
Colville function	ABC	2.21E-01	1.46E-01	2.21E-01	99305.3	1
	BSFABC	4.75E-02	4.86E-02	4.75E-02	91697.82	22
	MeABC	2.81E-02	4.32E-02	2.81E-02	66821.51	49
	BSFMeABC	7.76E-03	2.30E-03	7.76E-03	34125.51	100
Kowalik function	ABC	4.89E-04	7.16E-05	1.81E-04	90257.71	17
	BSFABC	4.89E-04	1.06E-04	1.82E-04	86195.29	33
	MeABC	4.07E-04	4.44E-05	9.95E-05	56582.12	82
	BSFMeABC	4.09E-04	6.21E-05	1.01E-04	47638.39	93
Shifted Rosenbrock	ABC	3.96E+02	7.60E+00	6.44E+00	95883.67	7
	BSFABC	3.93E+02	4.53E+00	3.11E+00	98790.52	7
	MeABC	3.95E+02	8.62E+00	5.22E+00	96327.29	7
	BSFMeABC	3.91E+02	1.89E+00	1.14E+00	90726.57	24
Shifted Griewank	ABC	-9.07E+01	1.59E+01	8.93E+01	100010.29	0
	BSFABC	-1.80E+02	5.76E-03	4.94E-03	63474.32	53
	MeABC	-1.80E+02	3.47E-03	1.73E-03	55363.63	79
	BSFMeABC	-1.80E+02	4.14E-03	2.25E-03	53639.3	75
Hosaki Problem	ABC	-2.31E+00	2.90E-02	3.23E-02	100024.26	0
	BSFABC	-2.35E+00	6.40E-06	5.90E-06	11469.26	89
	MeABC	-2.35E+00	7.17E-06	6.58E-06	15486.57	85
	BSFMeABC	-2.35E+00	7.38E-06	6.76E-06	17360.48	83
Meyer and Roth	ABC	1.91E-03	3.22E-06	1.95E-03	24454.11	96
	BSFABC	1.91E-03	3.12E-06	1.95E-03	14543.5	100
	MeABC	1.91E-03	2.76E-06	1.95E-03	9095.81	100
	BSFMeABC	1.91E-03	2.81E-06	1.95E-03	5915.46	100
lennard_jones	ABC	-3.89E+00	5.77E-01	5.22E+00	100032.81	0
	BSFABC	-9.10E+00	1.57E-03	1.64E-03	92189.12	51
	MeABC	-9.10E+00	1.36E-04	8.37E-04	14589.43	100
	BSFMeABC	-9.10E+00	1.11E-04	8.62E-04	15905.73	100
Welded beam design optimization	ABC	2.06E+00	1.27E-01	3.35E-01	100020	0
	BSFABC	1.82E+00	6.35E-03	9.59E-02	48674.76	91
	MeABC	1.91E+00	7.91E-02	1.85E-01	93686.68	12
	BSFMeABC	1.82E+00	5.75E-03	9.44E-02	36508.73	99
Compression spring	ABC	2.65E+00	1.08E-02	2.41E-02	99771.99	2
	BSFABC	2.66E+00	4.63E-03	3.01E-02	100037.5	0
	MeABC	2.64E+00	1.23E-02	1.42E-02	92797.98	11
	BSFMeABC	2.63E+00	1.21E-02	8.37E-03	60002.98	54

TABLE III. SUMMARY OF TABLE II

Test Problem	ABC vs BSFMeABC	BSFABC vs BSFMeABC	MeABC vs BSFMeABC
Griewank	+	+	+
Rastrigin	+	+	+
Alpine	+	+	+
Zakharov's	+	+	-
Salomon Problem	+	+	+
Inverted cosine wave function	+	+	+
Neumaier 3 Problem (NF3)	+	+	+
Colville function	+	+	+
Kowalik function	+	+	+
Shifted Rosenbrock	+	+	+
Shifted Griewank.	+	+	-
Hosaki Problem (HSK)	+	-	-
Meyer and Roth Problem (MR)	+	+	+
lennard_jones	+	+	+
Welded beam design optimization	+	+	+
Compression spring	+	+	+
Total No. of "+"	**16**	**15**	**13**

TABLE IV ACCELERATION RATE COMPARISON

Test Problem	ABC	BSFABC	MeABC
Griewank	1.551696058	1.25093	1.417255
Rastrigin	1.015636245	1.020424	1.003697
Alpine	1.000786017	1.000701	1.001266
Zakharov's	1.069418294	1.069418	0.895817
Salomon Problem	3.857719104	3.862691	0.934539
Inverted cosine wave function	2.237063016	1.566039	1.624013
Neumaier 3 Problem (NF3)	4.07719787	4.05316	0.953035
Colville function	2.910001931	2.687075	1.95811
Kowalik function	1.894642325	1.809366	1.187742
Shifted Rosenbrock	1.056842224	1.088882	1.061732
Shifted Griewank.	1.864496554	1.183355	1.032147
Hosaki Problem (HSK)	5.761606822	0.660653	0.892059
Meyer and Roth Problem (MR)	4.133932103	2.458558	1.537634
lennard_jones	6.289105247	5.795969	0.917244
Welded beam design optimization	2.739618716	1.333236	2.566145
Compression spring	1.662783915	1.667209	1.546556

REFERENCES

[1] Brownlee, Clever algorithms: nature-inspired programming recipes. Jason Brownlee, 2011.

[2] M Dorigo, G Di Caro (1999) Ant colony optimization: a new metaheuristic. In: Evolutionary computation, 1999. CEC 99. Proceedings of the 1999 congress on, 2. IEEE

[3] J Kennedy, R Eberhart (1995) Particle swarm optimization. In: Neural networks, 1995. Proceedings., IEEE international conference on, 4, pp 1942–1948. IEEE

[4] KV Price, RM Storn, JA Lampinen (2005) Differential evolution: a practical approach to global optimization. Springer, Berlin

[5] J Vesterstrom, R Thomsen (2004) A comparative study of differential evolution, particle swarm optimization, and evolutionary algorithms on numerical benchmark problems. In: Evolutionary computation, 2004. CEC2004. Congress on, 2, pp 1980–1987. IEEE

[6] KM Passino (2002) Biomimicry of bacterial foraging for distributed optimization and control. IEEE Control SystMag 22(3):52–67

[7] D Karaboga (2005) An idea based on honey bee swarm for numerical optimization. Techn. Rep. TR06, Erciyes University Press, Erciyes

[8] D Karaboga, B Akay (2009) A comparative study of artificial bee colony algorithm. Appl Math Comput 214(1):108–132

[9] G Zhu, S Kwong (2010) Gbest-guided artificial bee colony algorithm for numerical function optimization. Appl Math Comput 217(7):3166–3173

[10] A. Banharnsakun, Tiranee Achalakul, Booncharoen Sirinaovakul (2011) The best-so-far selection in Artificial Bee Colony algorithm, Applied Soft Computing 11 (2011) 2888–2901

[11] A. Baykasoglu, L. Ozbakir, and P. Tapkan. Artificial bee colony algorithm and its application to generalized assignment problem. Swarm Intelligence: Focus on Ant and Particle Swarm Optimization, pages 113–144, 2007.

[12] T. Derelia and G.S. Dasb. A hybrid'bee (s) algorithm'for solving container loading problems. Applied Soft Computing, 2010.

[13] Y.M. Huang and J.C. Lin. A new bee colony optimization algorithm with idle- time-based filtering scheme for open shop-scheduling problems. Expert Systems with Applications, 2010.

[14] N. Suguna and K.G. Thanushkodi. An independent rough set approach hybrid with artificial bee colony algorithm for dimensionality reduction. American Journal of Applied Sciences, 8, 2011.

[15] B. Wu, C. Qian, W. Ni, and S. Fan. The improvement of glowworm swarm optimization for continuous optimization problems. Expert Systems with Applications, 2012.

[16] X. Yan, Y. Zhu, W. Zou, and L. Wang. A new approach for data clustering using hybrid artificial bee colony algorithm. Neurocomputing, 2012.

[17] D. Shanthi and R. Amalraj. Collaborative artificial bee colony optimization clustering using spnn. Procedia Engineering, 30:989–996, 2012.

[18] A. Banharnsakun, T. Achalakul, and B. Sirinaovakul. The best-so-far selection in artificial bee colony algorithm. Applied Soft Computing, 2011.

[19] Fister I, Fister Jr I, Brest J, Zumer V (2012) Memetic artificial bee colony algorithm for large-scale global optimization. Arxiv preprint arXiv:1206.1074

[20] Rao SS, Rao SS (2009) Engineering optimization: theory and practice. Wiley, New York

[21] Iacca G, Neri F, Mininno E, Ong YS, Lim MH (2012) Ockham's razor in memetic computing: three stage optimal memetic exploration. Inf Sci: Int J 188:17–43

[22] Jagdish Chand Bansal, Harish Sharma, K. V. Arya • Atulya Nagar(2013) Soft Computing, DOI 10.1007/s00500-013-1032-8

[23] Neri F, Iacca G, Mininno E (2011) Disturbed exploitation compact differential evolution for limited memory optimization problems. Inf Sci 181(12):2469–2487

[24] Iacca G, Neri F, Mininno E, Ong YS, Lim MH (2012) Ockham's razor in memetic computing: three stage optimal memetic exploration. Inf Sci: Int J 188:17–43

[25] Kang F, Li J, Ma Z (2011) Rosenbrock artificial bee colony algorithm for accurate global optimization of numerical functions. Inf Sci 181(16):3508–3531

[26] MM Ali, C Khompatraporn, and ZB Zabinsky. "A numerical evaluation of several stochastic algorithms on selected continuous global optimization test problems." J. of Global Optimization, 31(4):635–672, 2005.

[27] P.N. Suganthan, N. Hansen, J.J. Liang, K. Deb, YP Chen, A. Auger, and S. Tiwari. "Problem definitions and evaluation criteria for the CEC 2005 special session on real-parameter optimization." In CEC 2005, 2005.

[28] Akay B, Karaboga D (2010) A modified artificial bee colony algorithm for real-parameter optimization. Inf Sci. doi: 10.1016/j.ins.2010.07.015

[29] Clerc M (2012) List based pso for real problems., 16 July 2012

[30] Onwubolu GC, Babu BV (2004) New optimization techniques in engineering. Springer, Berlin

[31] Sandgren E (1990) Nonlinear integer and discrete programming in mechanical design optimization. J Mech Des 112:223

[32] Ragsdell KM, Phillips DT (1976) Optimal design of a class of welded structures using geometric programming. ASME J Eng Ind 98(3):1021–1025.

[33] Kumar Sandeep, Vivek Kumar Sharma, and Rajani Kumari. "An Improved Memetic Search in Artificial Bee Colony Algorithm." *International Journal of Computer Science and Information Technology (0975–9646)* 5.2 (2014): 1237-1247.