ISSN:0975-9646

Sandeep Kumar et al, / (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 5 (2) , 2014, 2559-2565

New Local Search Strategy in Artificial Bee Colony Algorithm

Sandeep Kumar , Pawan Bhambu , Vivek Kumar Sharma

Faculty of Engineering and Technology
Jagannath University, Jaipur, India

***Abstract*— Artificial Bee Colony (ABC) algorithm is a Nature Inspired Algorithm (NIA) which based on intelligent food foraging behaviour of honey bee swarm. This paper introduces a local search strategy that enhances exploration competence of ABC and avoids the problem of stagnation. The proposed strategy introduces two new local search phases in original ABC. One just after onlooker bee phase and one after scout bee phase. The newly introduced phases are inspired by modified Golden Section Search (GSS) strategy. The proposed strategy named as new local search strategy in ABC (NLSSABC). The proposed NLSSABC algorithm applied over thirteen standard benchmark functions in order to prove its efficiency.**

Keywords*— artificial bee colony algorithm; nature inspired algorithm; local search; memetic computing; swarm intelligence.

I. INTRODUCTION

Artificial bee colony Optimization techniques is one of the fashionable swam intelligence optimization technique anticipated by D. Karaboga [1]. These modus operandi are used to discover a set of values for the independent variables that optimizes the value of one or more dependent variables. There are number of trifling multivariable optimization problems with capriciously high dimensionality which cannot be solved by precise search methods in stirred time. So search algorithms capable of searching near-optimal or good solutions within adequate computation time are very realistic in real life. In few years, the technical community has noticed the importance of a large number of nature-inspired metaheuristics and hybrids of these nature-inspired optimization methods. Metaheuristics may be measured a widespread algorithmic skeleton that can be applied to poles apart optimization problems with comparative a small number of modifications to get a feel for them to a specific problem. Metaheuristics are anticipated to make bigger the capabilities of heuristics by hybridizing one or more heuristic strategies using a higher-level methodologies (hence ‘meta’). Metaheuristics are strategies that provide guidance to the search process. Hyperheuristics are up till now an additional extension that focuses on heuristics that adapt their parameters in order to get better efficacy or result, or the effectiveness of the computation progression. Hyperheuristics endow with high-level methodologies that possibly will make use of machine learning and get a feel for their search behaviour by modifying the application of the sub-procedures or even which procedures are used [2]. Algorithms on or after the meadow of computational intelligence, biologically inspired intelligent computing, and metaheuristics are applied to troublesome problems, to which more classical approaches may not be significant. Michalewicz and Fogel says that these tribulations are difficult [3] as: they has large number of feasible solutions in the search space due to which they not able to exploit the best results; The problem is so intricate, that just to facilitate any reply at all, we have to make use of such beginner's models of the problem that any consequence is in essence a waste of time; the appraisal function that describes the quality of whichever proposed explanation is noisy or varies with time, by this means requiring not just a solitary solution but an entire series of solutions; the promising elucidations are so deliberately constrained that constructing even single feasible answer is incredibly easier said than done, let alone searching for an most advantageous solution; the human being solving the problem is inadequately composed or imagines some psychological fencing that prevents them from discovering exact solution.

Nature inspired algorithms are encouraged by some natural happening, can be categorized as per their source of encouragement. Major classes of NIA are: Evolutionary Algorithms, Immune Algorithms, Neural Algorithms, Physical Algorithms, Probabilistic Algorithms, Stochastic Algorithms and Swarm Algorithms [2].

Evolutionary Algorithms are motivated by advancement of natural selection strategy. Evolutionary Algorithms fit into the Evolutionary Computation field of learning concerned with computational methods encouraged by the itinerary of action and mechanisms of biological progression. Examples of evolutionary algorithm are Differential Evolution (EA), Evolutionary Programming (EP), Evolution Strategies (ES), Gene Expression Programming, Genetic Algorithm (GA), Genetic Programming (GP), Grammatical Evolution, Learning Classifier structure, Non-dominated Sorting Genetic Algorithm, and Strength Pareto Evolutionary Algorithm [2].

Immune Algorithms are aggravated by the adaptive immune system of vertebrates. A simplified narration of the immune organization is an appendage system anticipated to care for the host organism from the intimidation posed to it from pathogens and noxious substances. Pathogens include an assortment of microorganisms such as bacteria, viruses, parasites and pollen. The conventional viewpoint regarding the responsibility of the immune system is alienated into two most important tasks: the detection and elimination of pathogen. This activity is classically referred to as the delineation of self (molecules and cells that are in the right place to the host organisms) from potentially destructive non-self. Like Clonal Selection Algorithm (CSA), Negative Selection Algorithm, Artificial Immune Recognition System, Immune Network Algorithm and Dendritic Cell Algorithm [2].

Neural Algorithms are encouraged by the flexibility and learning individuality of the human nervous coordination. Some well known neural algorithms are Perceptron, Back-propagation, Hopfield Network, Learning Vector Quantization and Self-Organizing Map [2].

Physical Algorithms are motivated by corporal and communal systems Physical algorithms are those algorithms motivated by a physical process. Most of the physical algorithm in general belong to the fields of metaheustics and Computational cleverness; even though do not fit neatly into the obtainable categories of the biological motivated techniques. In this vein, they could immediately as by far be referred to as nature inspired algorithms. Like Simulated Annealing, Extremal Optimization, Harmony Search, Cultural Algorithm, and Memetic Algorithm [2].

Probabilistic Algorithms are strategies that concentrate on methods that put together models and guesstimate distributions in search domains. Probabilistic Algorithms are those algorithms that sculpt a dilemma or explore a problem space using a probabilistic model of entrant solutions. Examples of probabilistic algorithms are Population-Based Incremental Learning, Univariate Marginal Distribution Algorithm, Compact Genetic Algorithm, Bayesian Optimization Algorithm and Cross-Entropy Method [2].

Stochastic Algorithms are algorithms that focus on the prologue of unpredictability into heuristic methods. Examples of stochastic algorithms are Random Search, Adaptive Random Search, Stochastic Hill Climbing, Iterated Local Search, Guided Local Search, Variable Neighbourhood Search, Greedy Randomized Adaptive Search, Scatter Search, Tabu Search, and Reactive Tabu Search [2].

Swarm Algorithms focus on strategies that make use of the properties of cooperative intelligence. Swarm intelligence is the study of computational systems motivated by the ‘collective intelligence’. Collective Intelligence emerges all the way through the cooperation of large numbers of uniform agents in the surroundings. Examples consist of schools of fish, group of birds, and colonies of ants. Such intelligence is decentralized, self-organizing and scattered throughout surroundings. In nature such systems are usually used to solve problems such as successful foraging for food, prey escaping, or colony repositioning. The information is classically stored all the way through the participating uniform agents. Some examples of swarm intelligence are Particle Swarm Optimization, Ant System, Ant Colony System, Bees Algorithm, Spider Monkey Optimization Algorithm and Bacterial Foraging Optimization Algorithm.

Real-world optimization problems and generalizations thereof know how to haggard from the majority fields of science, management, engineering, and information technology. Prominently, function optimization problems have had a long institution in the fields of Artificial Intelligence in encouraging basic research into new dilemma solving strategies, and for finding and verifying systemic behaviour against benchmark problem instances [1]. The enhanced local search strategy proposed in this paper is based on one of the youngest member of NIA family the Artificial Bee Colony (ABC) algorithm [1], which mimics the extra ordinary food foraging behaviour of most intelligent insect that is honey bees swarm. ABC is a swarm-based intelligent stochastic optimization algorithm that has shows significant performance in solving continuous [4]-[6], combinatorial [7]-[10] and many more complex optimization problems. Stochastic optimization algorithms are those so as to use unpredictability to induce non-deterministic characteristics, contrasted to entirely deterministic strategies. Most strategies from the fields of biologically motivated Computation, Computational Intelligence and Metaheuristics may be evaluated to relate the field of Stochastic Optimization [2].

The organization of rest of paper is as follow: In segment 2, it introduce Artificial Bee Colony Algorithm in detailed manner, one of the newest swarm based method introduces by D. Karaboga [1]. Next section discusses some recent and important improvement and modifications in ABC. Section 4 establishes the proposed New Local Search Strategy in ABC algorithm. Section 5 contains experimental setup and results followed by conclusion and reference sections.

II. ARTIFICIAL BEE COLONY ALGORITHM

The ABC algorithm impersonates the food foraging behaviour of the honey bees with three groups of bees: employed bees, onlookers and scouts. A honey bee working to forage a food source (i.e. solution) previously visited by itself and searching only around its vicinity is called an employed bee. Employed bees perform waggle dance upon returning to the hive to propagate the information of its food source to the rest of the colony. A bee waiting around the dance floor to choose any of the employed bees to follow is called an onlooker. A bee indiscriminately searching a search space for finding a food source is called a scout. For every food source, there is only one employed bee and a number of adherent bees. The scout bee, after finding some food source better than some threshold value, also performs a waggle dance to share this information. In the ABC algorithm accomplishment, half of the colony consists of employed bees and the other half constitutes the onlookers. The number of food sources (i.e., solutions being exploited) is equal to the number of employed bees in the colony. The employed bee whose food source (i.e. solution) is fatigued (i.e. the solution has not been improved after quite a few attempts) becomes a scout. The meticulous algorithm is given below:

Algorithm 1: Artificial Bee Colony Algorithm

Step 1. Engender a preliminary population of N uniformly distributed individuals. Each individual x_{ij} is a food source (i.e. solution) and has D attributes. D is the dimensionality of the problem. x_{ij} represent i^{th} solution in j^{th} dimension. Where $j \in \{1, 2, \ldots, D\}$

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

Step 2. Guesstimate the fitness of each individual solution using the subsequent method,
if (solution value >= 0)

then

$$fit_i = \frac{1}{(2*solution\ value+1)}$$

else

$$fit_i = (1 + fabs(\frac{1}{solution\ value}))$$

Step 3. Each employed bee, sited at a food source that is poles apart from others, search in the propinquity of its current situation to find a better food source. For each employed bee, engender a new solution, v_i around its current location, x_i using the subsequent formula.

$$v_{ij} = x_{ij} + \phi_{ij}(x_{ij} - x_{kj})$$

Here, k ∈ {1, 2,…, N} and j ∈ {1, 2, …, D} are randomly preferred index. N is number of employed bees. ϕ_{ij} is a uniform random number from [-1, 1].

Step 4. Compute the fitness of both x_i and v_i. Apply insatiable selection strategy to pick better one of them.

Step 5. Determine and stabilize the likelihood values, P_i for each solution x_i using the subsequent formula.

$$P_{ij} = \frac{fit_i}{\sum_{i=1}^{SN} fit_i}$$

Step 6. Allocate each onlooker bee to a solution, x_i at haphazard with probability comparative to P_i.

Step 7. Engender new food positions (i.e. solutions), v_i for each onlooker bee.

Step 8. Calculate the fitness of every onlooker bee, x_i and the innovative solution, v_i. Apply insatiable selection procedure to keep the fitter one and throw out other.

Step 9. if a fastidious solution x_i has not been enhanced over a predefined number of cycles, then select it for denunciation. Replace the result by insertion a scout bee at a food source generated in an even way at random within the search space by means of subsequent formula.

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

for j = 1, 2,……,D

Step 10. Maintain pathway of the best food sources (solution) found as far as this.

Step 11. Check termination criteria. If the best solution found is acceptable or reached the maximum iterations, stop and return the best solution found so far. Otherwise go back to step 2 and repeat again.

III. Modifications in Artificial Bee Colony Algorithm

Often real world provides some complex optimization problems that cannot be easily dealt with available mathematical optimization methods. If the user is not very aware about the exact solution of the problem in hand then intelligence emerged from social behaviour of social colony members may be used to solve these kinds of problems. Honey bees are in the category of social insects. The foraging behaviour of honey bees produces an intelligent social behaviour, called as swarm intelligence. This swarm intelligence is simulated and an intelligent search algorithm namely, Artificial Bee Colony (ABC) algorithm is established by Karaboga in 2005 [1]. Since its commencement, a lot of research has been conceded to make ABC more and more efficient and to apply ABC for different types of problems.

In order to liberate the drawbacks of original ABC, researchers and scientists have adapted ABC in numerous ways. The potential where ABC can be enhanced are improvement of ABC control parameters named SN, ϕ_{ij} and limit (maximum cycle number). Hybridization of ABC with new population based probabilistic or deterministic algorithms. Some new control parameters are also introduced in different phases of ABC algorithm. D. Karaboga [1] has recommended that the value of ϕ_{ij} should be in the range of [-1, 1]. The value of limit (maximum cycle number) should be SN × D, where, SN is the number of possible solutions and D is the dimension of the problem. Wei-feng Gao et al. [11] proposed an improved solution search method in ABC, which depends on the fact that bee searches around the best solution of the preceding iteration to increase the exploitation. A. Banharnsakun et al. [12] introduced a new variant of ABC namely the best-so-far selection in artificial bee colony algorithm. To enhance the exploitation and exploration processes, they propose to make three major changes by introducing the best-so-far method, an adjustable search radius, and an objective-value-based comparison method in DE. J.C. Bansal et al. [13] anticipated balanced ABC; they added a new control parameter, Cognitive Learning Factor and also tailored range of ϕ in Artificial Bee Colony algorithm. Qingxian and Haijun [14] anticipated a change in the initialization scheme by making the initial group symmetrical, and the Boltzmann selection mechanism was employed instead of roulette wheel selection for humanizing the convergence ability of the ABC algorithm.

In order to take full advantage of the exploitation power of the onlooker stage, Tsai et al. anticipated the Newtonian law of universal gravitation in the onlooker segment of the basic ABC algorithm in which onlookers are elected based on a roulette wheel [15]. Baykasoglu et al. integrated the ABC algorithm with swing vicinity searches and greedy randomized adaptive search heuristic and applied it to the widespread assignment problem [16]. In addition, tailored versions of the Artificial Bee Colony algorithm are introduced and applied for efficiently solving real-parameter optimization problems by Bahriye Akay and Dervis Karaboga [17]. To adjust ABC behaviour for constrained search space Mezura et al. [18] anticipated four modifications related with the selection strategy, the scout bee machinist, and the equality and border line constraints. As an alternative of fitness relative selection, tournament selection is performed to utilize employed bee food sources by onlooker bees. Second, they employed self-motivated tolerance for equality constraints. In 2010, Zhu and Kwong [19] expected an enhanced version of ABC algorithm called gbest-guided ABC (GABC) algorithm by including the information of global best (gbest) explanation into the solution search equation to get better the exploitation. GABC is encouraged by PSO [20], which, in order to improve the exploitation capability, takes advantage of the information of the global best (gbest) solution to guide the search by candidate solutions. J.C. Bansal et al. [25] introduced memetic search in ABC algorithm with the intention of balancing exploitation and exploration. S. Kumar et al.[30]-[36] modified memetic search with new parameters and applied modified golden section search process in ABC algorithm, DE and Spider Monkey

Optimization algorithm to speed rate of convergence. In 2010, Derelia and Das [21] anticipated a hybrid bee(s) algorithm for solving container loading problems. In the wished-for algorithm, a bee(s) algorithm is hybridized with the heuristic filling modus operandi for the solution of container loading problems. In 2010, Huang and Lin [22] anticipated a new bee colony optimization algorithm with idle-time-based filtering scheme and its application for open shop-scheduling problems. In 2011, Nambiraj Suguna et al. [23] anticipated a self-regulating rough set approach hybrid with artificial bee colony algorithm for dimensionality diminution. In the wished-for work, effects of the perturbation rate, the scaling factor, and the limit are examined on real-parameter optimization. ABC algorithm hybridized with genetic algorithm to poise exploration and exploitation of search space [26], [27]. In 2012, Bin Wu et al. [24] anticipated perfection 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 preceding one to advance GSO performance.

IV. New Local Search Strategy In ABC

The proposed new local search strategy in ABC has two supplementary phases to original ABC [1]. The proposed algorithm introduces two new strategies; it modified the range of GSS process and add memetic search phase. The detailed modified ABC algorithm is given below:

Algorithm 2: New Local Search Strategy in ABC

Step 1. Generate an initial population of N uniformly distributed individuals. Each individual xij is a food source (i.e. solution) and has D attributes. D is the dimensionality of the problem. x_{ij} represent i^th solution in j^th dimension. Where j ∈ {1, 2, …, D}

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

Step 2. Estimate the fitness of each individual solution using the following method,
if (solution_value >= 0)
then

$$fit_i = \phi*(1/(2*solution\ value+1))+ (1\text{-}\phi)*(1+\text{fabs}(1/solution\ value))$$

else

$$fit_i = (1-\phi)*(1/(2*solution\ value+1))+ \phi*(1+\text{fabs}(1/solution\ value))$$

Step 3. Each employed bee, placed at a food source that is different from others, search in the proximity of its current position to find a better food source. For each employed bee, generate a new solution, v_{ij} around its current position, x_{ij} using the following formula.

$$v_{ij} = x_{ij} + \phi_{ij}(x_{ij} - x_{kj})$$

Here, k ∈ {1, 2,…, N} and j ∈ {1, 2, …, D} are randomly chosen indices. N is number of employed bees. ϕ_{ij} is a uniform random number from [-1, 1].

Step 4. Compute the fitness of both x_{ij} and v_{ij}. Apply greedy selection strategy to select better one of them.

Step 5. Calculate and normalize the probability values, P_{ij} for each solution x_i using the following formula.

$$P_{ij} = \phi \frac{fit_i}{Maximum\_fitness} + (1-\phi)\frac{fit_i}{\sum_{i=1}^{N} fit_i}$$

Step 6. Assign each onlooker bee to a solution, x_i at random with probability proportional to P_{ij}.

Step 7. Generate new food positions (i.e. solutions), v_{ij} for each onlooker bee.

Step 8. Compute the fitness of each onlooker bee, x_{ij} and the new solution, v_{ij}. Apply greedy selection process to keep the fitter one and abandon other.

Step 9. First memetic search phase inspired by GSS process.
Repeat while termination criteria meet
Compute $f_1 = b-(b-a)\times\psi,$ and $f_2 = a+(b-a)\times\psi,$ Calculate $f(f_1)$ and $f(f_2)$
If $f(f_1)$ < $f(f_2)$ then
b = f2 and the solution lies in the range [a, b]
else
a = f1 and the solution lies in the range [a, b]
End of if
End of while

Step 10. If a particular solution x_{ij} has not been improved over a predefined number of cycles, then select it for rejection. Replace the solution by placing a scout bee at a food source generated evenly at random within the search space using

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

for j = 1, 2,……,D

Step 11. Second memetic search phase inspired by GSS process.
Repeat while termination criteria meet
Compute $f_1 = b-(b-a)\times\psi,$ and $f_2 = a+(b-a)\times\psi,$
Calculate $f(f_1)$ and $f(f_2)$
If $f(f_1)$ < $f(f_2)$ then
b = f2 and the solution lies in the range [a, b]
else
a = f1 and the solution lies in the range [a, b]
End of if
End of while

Step 12. Keep track of the best food sources (solution) found so far.

Step 13. Check termination criteria. If the best solution found is acceptable or reached the maximum iterations, stop and return the best solution found so far. Otherwise go back to step 2 and repeat again.

V. Experimental Results

A. Test problems under consideration

In order to analyse the performance of NLSSABC diverse comprehensive optimization problems (f_1 to f_{13}) are chosen. These are continuous optimization problems and have different degrees of complexity, search range and multimodality. Test problems are taken from [28], [29] with the associated counterbalance values.

B. Experimental Setup

To prove the competence of NLSSABC, it is compared with original ABC and MeABC algorithms. To test NLSSABC over considered problems, subsequent experimental setting is adopted:

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
Zakharov	$f_2(x)=\sum_{i=1}^{D}x_i^2+(\sum_{i=1}^{D}\frac{ix_i}{2})^2+(\sum_{i=1}^{D}\frac{ix_1}{2})^4$	[-5.12, 5.12]	*f*(0) = 0	30	1.0*E*-02
Salomon Problem	$f_3(x)=1-\cos(2\pi\sqrt{\sum_{i=1}^{D}x_i^2})+0.1(\sqrt{\sum_{i=1}^{D}x_i^2})$	[-100, 100]	*f*(0) = 0	30	1.0*E*-01
Colville function	$f_4(x)=100(x_2-x_1^2)^2+(1-x_1)^2+90(x_4-x_3^2)^2+(1-x_3)^2$ $+10.1[(x_2-1)^2+(x_4-1)^2]+19.8(x_2-1)(x_4-1)$	[-10, 10]	*f*(1) = 0	4	1.0*E*-05
Kowalik function	$f_5(x)=\sum_{i=1}^{11}(a_i-\frac{x_1(b_i^2+b_ix_2)}{b_i^2+b_ix_3+x_4})^2$	[-5, 5]	*f*(0.1928, 0.1908, 0.1231, 0.1357) = 3.07*E*-04	4	1.0*E*-05
Shifted Rosenbrock	$f_6(x)=\sum_{i=1}^{D-1}(100(z_i^2-z_{i+1})^2+(z_i-1)^2+f_{bias}$, $z=x-o+1, x=[x_1,x_2,...x_D], o=[o_1,o_2,.......o_D]$	[-100, 100]	*f(o)=f_{bias}=390*	10	1.0*E*-01
Six-hump camel back	$f_7(x)=(4-2.1x_1^2+\frac{1}{3}x_1^4)x_1^2+x_1x_2+(-4+4x_2^2)x_2^2$	[-5, 5]	f(-0.0898, 0.7126) = -1.0316	2	1.0E-05
Easom's function	$f_8(x)=-\cos x_1\cos x_2 e^{(-(x_1-\pi)^2-(x_2-\pi)^2)}$	[-10, 10]	*f(π, π) = -1*	2	1.0*E*-13
Meyer and Roth Problem	$f_9(x)=\sum_{i=1}^{5}(\frac{x_1x_3t_i}{1+x_1t_i+x_2v_i}-y_i)^2$	[-10, 10]	*f(3.13, 15.16,0.78) = 0.4E*-04	3	1.0*E*-03
Braninss Function	$f_{10}(x)=a(x_2-bx_1^2+cx_1-d)^2+e(1-f)\cos x_1+e$	$x_1\in[-5,10]$, $x_2\in[0,15]$	*f(-π, 12.275) = 0.3979*	2	1.0*E*-05
Alpine	$f_{11}(x)=\sum_{i=1}^{n}\lvert x_i\sin x_i+0.1x_i\rvert$	[-10, 10]	*f(0) =0*	30	1.0*E*-05
McCormick	$f_{12}(x)=\sin(x_1+x_2)+(x_1-x_2)^2-\frac{3}{2}x_1+\frac{5}{2}x_2+1$	$-1.5\le x_1\le 4$, $-3\le x_2\le 3$	*f(-0.547, -1.547) = -1.9133*	30	1.0*E*-04
Shubert	$f_{13}(x)=-\sum_{t=1}^{5}i\cos((i+1)x_1+1)\sum_{i=1}^{5}i\cos((i+1)x_2+1)$	[-10, 10]	*f(7.0835, 4.8580)= -186.7309*	2	1.0*E*-05

- Colony size N = 60,
- Number of food sources S_N = N/2,
- ϕ_{ij} = rand[−1, 1],
- limit = 1500,
- The stopping hallmark is either maximum number of function assessment (which is set to be 200000) is reached or the tolerable error (outlined in Table I) has been achieved,
- The number of imitations/run =100,
- Parameter settings for the ABC and MeABC algorithm are identical to NLSSABC.

C. *Result Comparison*

Statistical results of NLSSABC with experimental setting as per last subsection are outlined in Table II. Table II show the comparison of results based on mean function value (MFV), standard deviation (SD), mean error (ME), average function evaluations (AFE) and success rate (SR) are accounted. Table II shows that most of the time NLSSABC do better than other considered algorithms in terms of competence (with less number of function appraisals) and correctness. Table III shows upshots of table II between NLSSABC, MeABC and original ABC algorithm. The proposed algorithm at all times gets better AFE and most of the time it also improve SD and ME. It is due to newly initiated local search phase and modified GSS process.

TABLE II: COMPARISON OF RESULTS FOR TEST PROBLEMS

Test Problem	Algorithm\Measure	MFV	SD	ME	AFE	SR
f_1	ABC	4.61E-03	9.10E-03	4.61E-03	81988.8	61
	MeABC	2.04E-04	1.42E-03	2.04E-04	37417.2	98
	NLSSABC	**1.04E-03**	**2.98E-03**	**1.04E-03**	**77758.02**	**89**
f_2	ABC	9.92E+01	1.49E+01	9.92E+01	100020	0
	MeABC	1.88E-02	9.79E-03	1.88E-02	98848.74	18
	NLSSABC	**9.59E-03**	**4.43E-04**	**9.59E-03**	**58718.52**	**100**
f_3	ABC	1.67E+00	2.30E-01	1.67E+00	100020.1	0
	MeABC	9.22E-01	3.58E-02	9.22E-01	21210.76	100
	NLSSABC	**9.22E-01**	**2.84E-02**	**9.22E-01**	**16846.92**	**100**
f_4	ABC	1.87E-01	1.38E-01	1.87E-01	100041.9	0
	MeABC	7.14E-03	2.92E-03	7.14E-03	28743.11	98
	NLSSABC	**7.68E-03**	**2.33E-03**	**7.68E-03**	**21845.4**	**100**
f_5	ABC	4.73E-04	6.99E-05	1.66E-04	87420.15	26
	MeABC	4.10E-04	5.28E-05	1.02E-04	62047.63	88
	NLSSABC	**3.98E-04**	**1.32E-05**	**9.01E-05**	**30936.77**	**100**
f_6	ABC	3.96E+02	9.68E+00	6.43E+00	95518.76	10
	MeABC	3.91E+02	3.63E+00	1.34E+00	74898.53	48
	NLSSABC	**3.90E+02**	**1.03E+00**	**4.13E-01**	**104029.3**	**84**
f_7	ABC	3.00E+00	2.07E-06	5.03E-07	73563.42	35
	MeABC	3.00E+00	4.40E-15	4.33E-15	5664.01	100
	NLSSABC	**3.00E+00**	**4.19E-15**	**4.16E-15**	**4401.02**	**100**
f_8	ABC	-1.03E+00	4.39E-03	4.60E-03	100043.5	0
	MeABC	-1.03E+00	1.32E-05	1.69E-05	61663.03	43
	NLSSABC	**-1.03E+00**	**1.40E-05**	**1.91E-05**	**120735.1**	**40**
f_9	ABC	-1.87E+00	4.18E-02	3.91E-02	99637.14	1
	MeABC	-1.91E+00	9.39E-06	9.15E-05	41575.84	87
	NLSSABC	**-1.91E+00**	**6.34E-06**	**8.81E-05**	**5036.13**	**100**
f_{10}	ABC	3.98E-01	6.65E-06	5.53E-06	10056.23	92
	MeABC	3.98E-01	6.56E-06	5.73E-06	9080.57	92
	NLSSABC	**3.98E-01**	**7.53E-06**	**6.99E-06**	**42614.76**	**79**
f_{11}	ABC	3.06E-02	2.13E-02	3.06E-02	100020	0
	MeABC	1.22E-05	2.38E-05	1.22E-05	75174.72	90
	NLSSABC	**9.25E-06**	**1.06E-06**	**9.25E-06**	**60865.44**	**100**
f_{12}	ABC	-2.31E+00	4.25E-02	3.72E-02	100031.6	0
	MeABC	-2.35E+00	1.23E-05	8.48E-06	48699.6	81
	NLSSABC	**-2.35E+00**	**6.18E-06**	**5.88E-06**	**20200.95**	**92**
f_{13}	ABC	1.91E-03	5.19E-06	1.95E-03	30838.67	91
	MeABC	1.91E-03	2.94E-06	1.95E-03	3577.26	100
	NLSSABC	**1.91E-03**	**2.96E-06**	**1.95E-03**	**3575.05**	**100**

TABLE III: SUMMARY OF TABLE II OUTCOMES

Test Problem	NLSSABC vs ABC	NLSSABC vs MeABC
f_1	+	-
f_2	+	+
f_3	+	+
f_4	+	+
f_5	+	+
f_6	+	+
f_7	+	+
f_8	+	-
f_9	+	+
f_{10}	-	-
f_{11}	+	+
f_{12}	+	+
f_{13}	+	+
Total Number of + sign	**12**	**10**

VI. CONCLUSION

At this point in this paper, two new phases are initiated in original ABC. Newly added steps are inspired by memetic ABC and position update achieved on the basis of appropriateness of individual in order to balance intensification and diversification of local search breathing space. Additional, the advanced strategy is applied to get to the bottom of 13 well-known benchmark functions. With the help of experiments over test problems, it is shown that the addition of the proposed strategy in the original ABC improves the trustworthiness, competence and accurateness as weigh against to their original adaptation. Table II and III show that the anticipated NLSSABC is competent to solve largest part the considered problems with smaller amount of pains.

REFERENCES

[1] D. Karaboga, "An idea based on honey bee swarm for numerical optimization", Technical Report-TR06, Erciyes Univ., Computer Engg. Department, 2005.

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

[3] Z. Michalewicz and DB Fogel. How to Solve It: Modern Heuristics. Springer, 2004.

[4] D. Karaboga, B. Basturk, Artificial Bee Colony (ABC) Optimization Algorithm for Solving Constrained Optimization Problems, LNCS: Advances in Soft Computing: Foundations of Fuzzy Logic and Soft Computing, CA, Vol: 4529/2007, pp: 789-798, Springer- Verlag, 2007, IFSA 2007.

[5] D. Karaboga, B. Basturk, "An artificial bee colony (ABC) algorithm for numeric function optimization", in Proceedings of the IEEE Swarm Intelligence Symposium 2006, Indianapolis, Indiana, USA, 12–14 May 2006.

[6] XS Yang, "Engineering Optimizations via Nature-Inspired Virtual Bee Algo.", IWINAC2005, LNCS 3562, J. M. Yang and J.R. Alvarez (Eds.), Springer-Verlag, Berlin Heidelberg, pp. 317-323, 2005.

[7] P. Lucic, D. Teodorovic, "Computing with Bees: Attacking Complex Transportation Engineering Problems", in International Journal on Artificial Intelligence Tools, vol. 12, no. 3, pp. 375-394, 2003.

[8] CS Chong, M. Y. H. Low, A. I. Sivakumar, K. L. Gay, "A Bee Colony Optimization Algorithm to Job Shop Scheduling", Proc. of the 37th Winter Simulation, California, pp. 1954-1961, 2006.

[9] GZ Markovic, D. Teodorovic, V. S. Acimovic-Raspopovic, "Routing and Wavelength Assignment in All-Optical Networks Based on the Bee Colony Optimization", AI Communications, v.20 no.4, p.273-285, December 2007.

[10] N Quijano, KM Passino, "Honey Bee Social Foraging Algorithms for Resource Allocation Theory and Application", American Control Conference, New York City, USA, 2007.

[11] W Gao and S Liu. A modified artificial bee colony algorithm. Computers & Operations Research, 2011.

[12] A Banharnsakun, T Achalakul, and B Sirinaovakul. The best-so-far selection in artificial bee colony algorithm. Applied Soft Computing, 2010.

[13] JC Bansal, H Sharma, A Nagar and KV Arya. "Balanced artificial bee colony algorithm." International Journal of Artificial Intelligence and Soft Computing 3.3 (2013): 222-243.

[14] D Haijun, F Qingxian. Bee colony algorithm for the function optimization. Science Paper Online, August 2008.

[15] P.W. Tsai, J.S. Pan, B.Y. Liao, and S.C. Chu. Enhanced artificial bee colony optimization. International Journal of Innovative Computing, Information and Control, 5(12), 2009.

[16] 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.

[17] B. Akay and D. Karaboga. A modified artificial bee colony algorithm for real-parameter optimization. Information Sciences, 2010.

[18] E. Mezura-Montes and O. Cetina-Dom´ınguez. Empirical analysis of a modified artificial bee colony for constrained numerical optimization. Applied Mathematics and Computation, 2012.

[19] G. Zhu and S. Kwong. Gbest-guided artificial bee colony algorithm for numerical function optimization. Applied Mathematics and Computation, 2010.

[20] J Kennedy and RC Eberhart. Particle swarm optimization. In Proceedings IEEE int'l conf. on neural networks Vol. IV, pages 1942–1948, 1995.

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

[22] 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.

[23] 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.

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

[25] J.C. Bansal, H. Sharma, K.V. Arya and A. Nagar, "Memetic search in artificial bee colony algorithm." Soft Computing (2013): 1-18.

[26] S. Kumar, V.K. Sharma, and R. Kumari. "A Novel Hybrid Crossover based Artificial Bee Colony Algorithm for Optimization Problem." International Journal of Computer Applications 82, 2013.

[27] S. Pandey and S. Kumar "Enhanced Artificial Bee Colony Algorithm and It's Application to Travelling Salesman Problem." HCTL Open International Journal of Technology Innovations and Research, Volume 2, 2013:137-146.

[28] 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.

[29] 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.

[30] S Kumar, VK Sharma and R Kumari, "Randomized Memetic Artificial Bee Colony Algorithm". International Journal of Emerging Trends & Technology in Computer 3(1): 52-62, March 2014.

[31] S Kumar, VK Sharma and R Kumari, "An Improved Memetic Search in Artificial Bee Colony Algorithm", International Journal of Computer Science and Information Technology (0975 – 9646), 5(2): 1237-1247, 2014

[32] S Kumar, VK Sharma and R Kumari, "Modified Position Update in Spider Monkey Optimization Algorithm", International Journal of Emerging Technologies in Computational and Applied Sciences (IJETCAS), 7(2): 198-204, March 2014

[33] S Kumar, VK Sharma and R Kumari, "Enhanced Local Search in Artificial Bee Colony Algorithm", International Journal of Emerging Technologies in Computational and Applied Sciences (IJETCAS), 7(2): 177-184, March 2014

[34] S Kumar, VK Sharma and R Kumari, "Memetic Search in Differential Evolution Algorithm", International Journal of Computer Applications (0975 – 8887) 90(6):40-47, March 2014. Published by Foundation of Computer Science, New York, USA. DOI: 10.5120/15582-4406

[35] S Kumar, VK Sharma and R Kumari, "Improved Onlooker Bee Phase in Artificial Bee Colony Algorithm", International Journal of Computer Applications (0975 – 8887) 90(6):31-39, March 2014. Published by Foundation of Computer Science, New York, USA. DOI: 10.5120/15579-4304

[36] S Kumar, VK Sharma and R Kumari, "Comparative study of Hybrids of Artificial Bee Colony Algorithm", International Journal of Information, Communication and Computing Technology (IJICCT) Volume 1, Issue 2, pages: 20-28. Feb 2014.