An Improved Hybrid Artificial Bee Colony Algorithm for Solving Real Parameter Optimization Problems

Hai Shan, Toshiyuki Yasuda, and Kazuhiro Ohkura
Graduate School of Engineering,
Hiroshima University
Higashi-hiroshima, Hiroshima,
Japan

Abstract—**An artificial bee colony (ABC) algorithm is one of numerous swarm intelligence algorithms that employ the foraging behavior of honeybee colonies. To improve the convergence performance and search speed of finding the best solution using this approach, we propose an improved hybrid ABC algorithm in this paper. To evaluate the performance of the standard ABC, the proposed ABC, differential evolution (DE) and particle swarm optimization (PSO) algorithms, we implemented the experiments of real parameter numerical optimization problems based on 28 benchmark functions defined by IEEE Congress on Evolutionary Computation (CEC) 2013 test suite with all the dimension size of 10, 30, and 50, respectively. As the result, we observe that the proposed ABC algorithm performs very competitively than standard ABC, DE and PSO algorithms.**

Keywords—Artificial bee colony algorithm; swarm intelligence; real parameter optimization problems; differential evolution; particle swarm optimization

I. Introduction

In this couple of decades, swarm intelligence (SI), which is a discipline of artificial intelligence, has attracted the interest of many research scientists. SI techniques are population based stochastic methods used in combinatorial optimization problems in which the collective behavior of relatively simple individuals arises from their local interactions with their environment to produce functional global patterns. SI represents a meta-heuristic approach to solve a variety of problems. Bonabeau et al. [1] defined SI as “any attempt to design algorithms or distributed problem-solving devices inspired by the collective behavior of social insect colonies and other animal societies.” SI is becoming an important research area for computer scientists, engineers, economists, bioinformaticians, operational researchers, and many other disciplines.

Various SI algorithms have been proposed since the end of the last century, e.g., particle swarm optimization (PSO) [2], differential evolution (DE) [3], ant colony optimization [4], artificial bee colony (ABC) [5]. These SI algorithms have been proposed to solve optimization problems.

Optimization is collection of mathematical principles and methods used for solving quantitative problems in many disciplines. The subject grew from a realization that quantitative problems in manifestly different disciplines have important mathematical elements in common. The development of optimization techniques has paralleled advances not only in computer science but also in operations research, numerical analysis, game theory, mathematical economics, control theory, and combinatorics. In optimization problem, the best values are determined from all feasible solutions of a given problem; the aim of optimization is to obtain relevant parameter values that enable an objective function to generate the minimum or maximum value [6]. For solving the optimization problems with SI algorithm, the IEEE Congress on Evolutionary Computation (CEC) 2013 test suite [7] is an invaluable resource. The CEC 2013 test suite includes 28 benchmark functions and does not make use of exact equations. Effective and efficient optimization algorithms are required to solve more difficult problems, including complex real world optimization problems. As one of more developed SI algorithms in recent years, the artificial bee colony (ABC) algorithm is introduced to solve these benchmark problems defined on the CEC 2013 test suite.

The ABC algorithm was introduced by Karaboga [5] as a technical report, then its performance was measured using benchmark optimization functions [8,9]. A recent study [10] showed that ABC algorithm performs significantly better or at least comparable to other SI algorithms as genetic algorithm [11], DE, and PSO algorithms. The ABC algorithm has been applied to several fields in various ways, for example, training neural networks [12]; solving sensor deployment problem [13]; applied for engineering design optimization [14].

The ABC algorithm is superior to other algorithms in terms of its simplicity, flexibility and robustness. In addition, the ABC algorithm requires fewer training parameters; so combining it with other algorithms is easier. The standard ABC algorithm was based on the results of some standard benchmark problems, however, as an initial proposal, it still has a considerable performance gap with respect to state-of-the-art algorithms. In particular, it was found to have relatively poor performance on composite and non-separable functions, and have a slow convergence rate toward high quality solutions. To improve the performance, the ABC algorithm has been extended in a number of ways recently. For example, Alatas [15] proposed a chaotic ABC algorithm, in which many chaotic maps for parameters adapted from the original ABC were introduced to improve its convergence performance; Zhu and Kwong [16] proposed a global best (Gbest) guided ABC algorithm by incorporating the information of global best solution into the solution search equation to improve the exploitation; Banharnsakun et al. [17] introduced best-so-far selection to standard ABC algorithm;

Karaboga and Gorkemli [18] introduced a combinatorial ABC for travelling salesman problems. In addition, Gao and Liu [19] proposed a modified ABC algorithm (MABC) that used a modified solution search equation with chaotic initialization, which excluded the probabilistic selection scheme and scout bees phases.

Along with the advantages of the improved versions of ABC, however, a few disadvantages still exist. For example, ABC algorithms have low convergence speeds, low exploitation abilities, and are also easily trapped in local optima. To overcome these disadvantages, we focus on the predominance of hybridizing other algorithms easily and propose an improved hybrid ABC algorithm, which inspired by self adaptive mechanism, incorporated DE and PSO algorithms.

The parameters for our proposed ABC are setup through the comparative experiments of parameter selection. We solve the CEC 2013 test suite benchmark problems based on the above setting. Finally, comparative experiments are implemented for our proposed ABC, the standard ABC, DE, and PSO algorithms. The remaining parts of the paper are organized as follows. The standard ABC algorithm is introduced in Section II. In Section III, our proposed ABC algorithm is described. The experimental setup and results are discussed in Section IV. Finally, Section V is the conclusion.

II. THE STANDARD ABC ALGORITHM

The ABC algorithm is a swarm based meta-heuristic algorithm introduced by Karaboga that has successfully been applied to numerical optimization problems [8,9,12]. In the ABC algorithm, the artificial bee colony comprises three kinds of bees: employed bees, onlooker bees, and scout bees. Employed bees search for food source sites by modifying the site in their memory, evaluating the nectar amount of each new source, and memorizing the more productive site through a selection process. These bees share information related to the quality of the food sources they exploit in the “dance area”. The number of employed bees is equal to the number of food sources for the hive. Onlooker bees search for food sources based on the information coming from employed bees within the hive. As such, more beneficial sources have higher probability to be selected by onlookers. Further, onlooker bees choose food sources depending on the given information through probabilistic selection and modify these sources. When the food source is abandoned, a new food source is randomly selected by a scout bee to replace the abandoned source. The number of food sources in ABC algorithm is equivalent to the number of solutions in a population for an optimization problem. The number of nectar sites of a food source represents the fitness cost of the associated solution.

The main steps of the algorithm are given below:

1. Initialize the population of solutions x_{ij} with

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

where, $i \in$ *1, 2, ..., SN* and $j \in$ *1, 2, ...,D* are randomly selected indexes, *SN* is the number of food source, and *D* is the dimension size.

2. Evaluate the population.

3. Initialize *cycle* to 1.

4. Produce new solutions v_i for the employed bees by using (1), then evaluate them as follows

$$v_{ij} = x_{ij} + \phi_{ij}(x_{ij} - x_{kj}) \tag{2}$$

where ϕ_{ij} is a uniformly distributed random number in the range [-1,1]; *i, k* $\in$ *1, 2, ..., SN* are randomly selected indexes with *k* different from *i* and *j* $\in$ *1, 2, ...,D* is a randomly selected index.

5. Apply the greedy selection process for the employed bees.

6. If the solution does not improve, add 1 to the *trail*, otherwise, set the *trail* to 0.

7. Calculate probability values P_i for the solutions using (3) as

$$p_i = \frac{fit_i}{\sum_{n=1}^{SN} fit_n} \tag{3}$$

where fit_i is the fitness value of solution *i*.

8. Produce new solutions for the onlooker bees from solutions x_i , which is selected depending on p_i , then evaluate them.

9. Apply the greedy selection process for the onlooker bees.

10. If the solution does not improve, add 1 to the *trail*, otherwise, set the *trail* to 0.

11. Determine the abandoned solution for the scout, if it exists, and replace it with a new random solution using (1).

12. Memories the best solution achieved so far.

13. Add 1 to *cycle*.

14. Repeat until cycle reaches a predefined maximum cycle number (*MCN*).

III. THE PROPOSED ABC ALGORITHM

There are two common aspects in population based heuristic algorithms, i.e., exploration and exploitation [20]. Exploration is the ability to expand the search space, and exploitation is the ability to find optima around a good solution. Exploration and exploitation play key roles in SI algorithms. They coexist in the evolutionary process of algorithms such as PSO, DE, and ABC, but they contradict each other.

To achieve a good optimization performance with higher convergence speeds and not trapped in local optima, a self adaptive mechanism to change the search range related with a cycle number was introduced, and then combined with DE to improve the performance of the employed bees. In the standard ABC algorithm, a random perturbation is added to the current solution to produce a new solution. This random perturbation is weighted by ϕ_{ij} selected from [-1,1] and is a uniformly distributed real random number in the standard ABC. Too large or too small value of ϕ_{ij} affects the convergence speed. Therefore, a self-adaptive mechanism is used to balance the convergence speed and the exploration ability of the algorithm for employed bees. The self adaptive mechanism of ABC consists of very simple structure and is easy to implement. The ϕ_{ij} is changed with the cycle number according to a random value called rand in the range [0,1] for the food searching process of an employed bee; ϕ_{ij} is determined as (4).

$$\phi_{ij} = \begin{cases} -\mathrm{e}^{-3*cycle/(25*MCN)}, & 0 \le rand \le 0.5 \\ \mathrm{e}^{-3*cycle/(25*MCN)}, & 0.5 < rand \le 1 \end{cases} \tag{4}$$

Swagatam and Suganthan [21] proved that DE algorithm was a simple yet powerful and efficient population based algorithm for many global optimization problems. To further improve the performance of the DE algorithm, researchers have suggested different schemes of DE. Like other evolutionary algorithms, DE also relies on an initial random population generation and then improves its population via mutation, crossover and selection processes. The searching food source process in the standard ABC algorithm is similar to the mutation process of DE. Besides, in DE, the best solution in the current population is very advantageous for higher convergence performance. As one scheme of the mutations of DE, “DE/best/1” can effectively maintain population diversity. The “DE/best/1” mutation strategy was combined with the food search process of the standard ABC algorithm to produce a new search equation (5) and improved the convergence ability.

$$v_{ij} = x_{best,j} + \phi_{ij}(x_{ij} - x_{kj}) \tag{5}$$

where *i, k* ∈ *1, 2, ..., SN* are randomly selected indexes with *k* different from *i* ; *j* ∈ *1, 2, ...,D* is a randomly selected index and ϕ_{ij} is the parameter given in (4).

It was determined that the search ability of the ABC algorithm is good at exploration, but poor in terms of exploitation. Specifically, the relationship of employed bees and onlooker bees are focused on exploration and exploitation, respectively. Employed bees explore new food sources and send information to onlooker bees, and onlooker bees exploit the food sources explored by employed bees. In the standard ABC algorithm, much time is required to find the food source due to poor exploitation abilities and lower convergence speeds. To improve the exploitation ability and convergence performance of the algorithm, we incorporated PSO into the standard ABC algorithm. PSO is based on the simulation of simplified social animal behaviors and it has the advantage of good convergence performance. We modified the onlooker bee search solution by taking advantage of the search mechanism of PSO; our modified search equation for onlooker bees is shown as (6).

$$v_{ij} = x_{ij} + \varphi_{ij}(x_{ij} - x_{kj}) + \psi_{ij}(x_{best,j} - x_{ij}) \tag{6}$$

where *i, k* ∈ *1, 2, ..., SN* are randomly selected indexes with *k* different from *i* ; *j* ∈ *1, 2, ..., SN* is a randomly selected index; $x_{best,j}$ is the *j*-th element of the best dominant solution, and $\varphi_{ij} \in [-1,\ 1]$ and $\psi_{ij} \in [0,\ 1.2]$ are uniformly distributed random numbers.

The proposed ABC algorithm modified the standard ABC on step 4 and step 8. The modification (4) and (5) are introduced to substitute (2) on step 4; the modification (6) is introduced to substitute (2) on step 8.

IV. Experiments

A. Experimental Setup

The CEC 2013 test suite extends its predecessor CEC 2005 test suite. In the CEC 2013 test suite, the previously proposed composition functions are improved, and additional test functions are included. There are 28 numerical test functions that are minimization problems categorized into the following three groups: unimodal functions (F1-F5), multimodal functions (F6-F20), and composition functions (F21-F28). The detailed description of the CEC 2013 test suite is available in [7].

All test functions are minimization problems defined as follows:

To minimize *f(x)*, $x = [x_1, x_2, ..., x_D]^T$, where *D* is the number of dimensions.

Given $o = [o_1, o_2, ..., o_D]^T$, the shifted global optimum distributed randomly in the range $[-80, 80]^D$, all test functions are shifted to *o* and are scalable. For convenience, the same search ranges $[-100, 100]^D$ is defined for all test functions.

In this paper, both the standard ABC algorithm and the proposed ABC algorithm for all 28 test functions defined in the CEC 2013 test suite with parameters selected by comparing experiments at three dimension sizes, i.e., 10, 30, and 50 are evaluated. The 28 test functions were executed 51 times with respect to each test function at each problem dimension size. The algorithms were terminated when the *MCN* was reached for function evaluations or when the error value was smaller than 10^{-8} . In our experiments, we set maximum evaluation sizes to 100,000, 300,000, and 500,000 for problem dimension sizes of 10, 30, and 50, respectively. We also performed the Wilcoxon rank sum test with significance level of 0.05, and conducted comparative experiments for the standard ABC, proposed ABC, DE [22] and PSO [23] algorithms. We used the C language for our experiments on a Linux system with an Intel Core i3 CPU 540 @3.07GHz * 4 with 64-bit processing.

B. Experimental Results

Table 1 shows the parameter adjustment experiment results. NP is the number of population size. In this table, “-/-/-” indicates the competitive performance of the algorithm on three dimension sizes of 10, 30, and 50. “-” indicates that its performance is significantly worse than others on that dimension size.

TABLE 1. Parameter Adjustment Experiments for the Standard ABC and the Proposed ABC algorithms

Limit/NP	50	100	150	200	300
50	-/-/-	-/-/-	-/-/-	-/-/-	-/-/-
100	-/-/-	10/30/-	10/-/50	-/-/-	-/-/-
150	10/30/-	10/30/50	10/-/50	10/-/50	-/-/-
250	10/30/-	10/30/-	-/30/50	-/30/50	-/-/-
400	10/-/-	-/30/50	-/-/50	-/-/-	-/-/-

According to Table 1, we can observe that ABC is not sensitive to the parameter choice of lower or higher population size and limit. Parameters were selected of limit for 150 and population size for 100 respectively by comparing experiments.

After comparative experiments of our proposed ABC, standard ABC, DE and PSO algorithms for the function error of mean values for 100,000, 300,000, and 500,000 evaluations on dimension sizes of 10D, 30D, and 50D, respectively, the number of better, similar and worse convergence performances of the mean values of the proposed ABC algorithm versus standard ABC, DE and PSO algorithms are analyzed statistically on 28 test functions in Table 2.

TABLE 2 Comparison Performances of Function Errors of Mean Value for the Proposed ABC to Standard ABC, DE, and PSO Algorithms

Proposed ABC (10D) VS	ABC	DE	PSO
Better	7	19	24
Similar	15	6	4
Worse	6	3	0
Proposed ABC (30D) VS	ABC	DE	PSO
Better	8	17	22
Similar	13	5	4
Worse	7	6	2
Proposed ABC (50D) VS	ABC	DE	PSO
Better	10	14	15
Similar	10	6	9
Worse	8	8	4

With the dimension size increasing, the number of functions with better and similar convergence performances of the proposed ABC decreased compared to standard ABC, DE and PSO algorithms according to Table 2. It is concluded that the better and similar convergence performances of the proposed ABC algorithm is quite competitive to standard ABC, DE, and PSO algorithms. Especially, the convergence performance is very significant as compared to PSO algorithm.

Fig. 1–10 illustrate the box-plots for function errors of mean value of the proposed ABC, standard ABC, DE, and PSO algorithms on 10D, 30D, and 50D, respectively. The number “1, 2, 3, 4”, “5, 6, 7, 8” , and “9, 10, 11, 12” of these figures indicate that the function errors of mean value of the proposed ABC, standard ABC, DE, and PSO algorithms on 10D, 30D, and 50D, respectively. Fig. 1 and Fig. 2 illustrate the box-plots for function errors of the mean value of unimodal functions. According to Fig. 1, the convergence performance of the proposed ABC algorithm is same as standard ABC, DE, and PSO algorithms for function F1 on 10D, 30D, and 50D, it was also observed that function F1 reached the best value of zero for function error of mean value. From Fig. 2, the convergence performance of the proposed ABC algorithm is competitive to the standard ABC, but not better than DE, and PSO algorithms for function F4. DE algorithm gets the best convergence performance on 30D, and 50D.

Fig. 3–7 show the box-plots for function errors of mean value of multimodal functions. According to Fig. 3, the convergence performance of the proposed ABC is quite similar to standard ABC, DE, and PSO algorithms on function F8 on 10D, 30D, and 50D. From Fig. 4, it is observed that the convergence performance of the proposed ABC algorithm for function F13 is competitive to other algorithms, but the standard ABC algorithm is not competitive to DE and PSO algorithms on 50D; moreover, DE algorithm is better than standard ABC and PSO algorithms on 30D and 50D. The convergence performance sequence for these algorithms is the

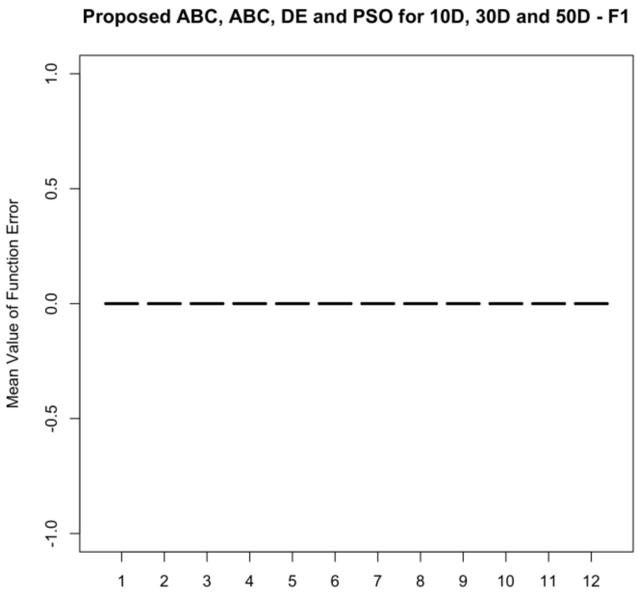

Fig. 1 Box-plot of function F1 with function errors of mean value of the proposed ABC, standard ABC, DE, and PSO algorithms on 10D, 30D, and 50D

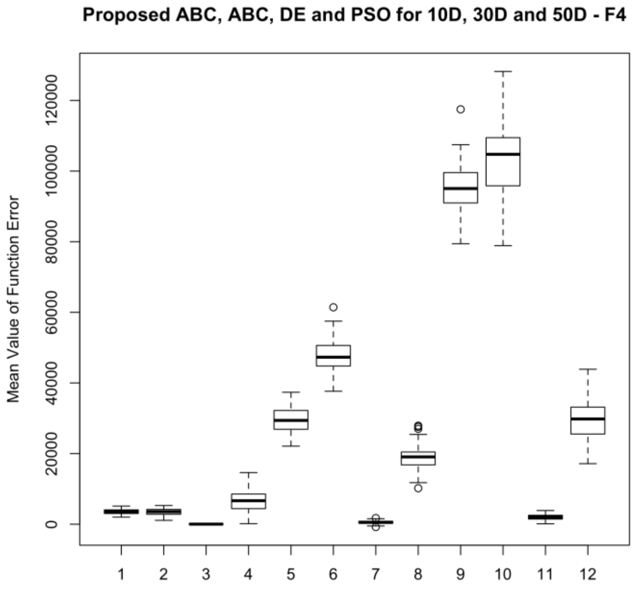

Fig. 2 Box-plot of function F4 with function errors of mean value of the proposed ABC, standard ABC, DE, and PSO algorithms on 10D, 30D, and 50D

proposed ABC, standard ABC, DE, and PSO for function F14 on 10D, 30D, and 50D according to Fig. 5. Especially, the standard ABC and proposed ABC algorithms reached the best performance. From Fig. 6, it can be seen that the convergence performance of the proposed ABC and standard ABC algorithms are very similar and better than DE and PSO algorithms for function F17. The convergence performance of PSO algorithm is better than DE algorithm on 10D and the convergence performance of DE algorithm is better than PSO algorithm on 30D and 50D.

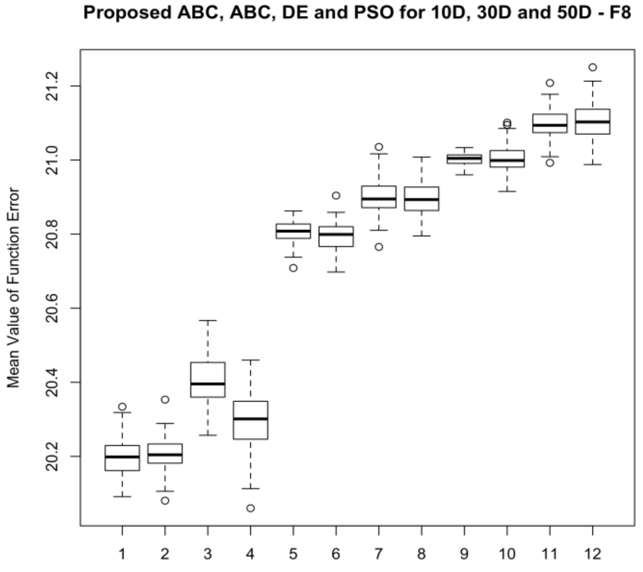

Fig. 3 Box-plot of function F8 with function errors of mean value of the proposed ABC, standard ABC, DE, and PSO algorithms on 10D, 30D, and 50D

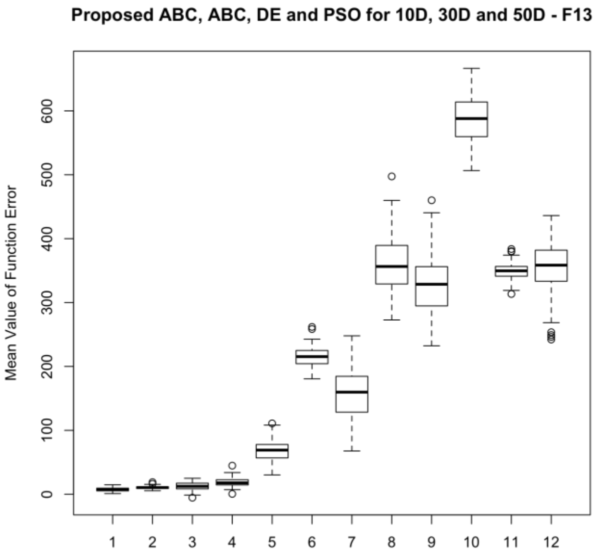

Fig. 4 Box-plot of function F13 with function errors of mean value of the proposed ABC, standard ABC, DE, and PSO algorithms on 10D, 30D, and 50D.

According to Fig. 7, the convergence performance of the proposed ABC is better than standard ABC, but worse than DE and PSO algorithms on 30D and 50D. For remaining multimodal functions, it is observed that the convergence performances of the proposed ABC and standard ABC algorithms are competitive to DE and PSO algorithms, DE algorithm is much better than PSO algorithm. The convergence performance of DE algorithm was the best on function F7 on 30D and 50D. The same result is seen for function F18 on D50.

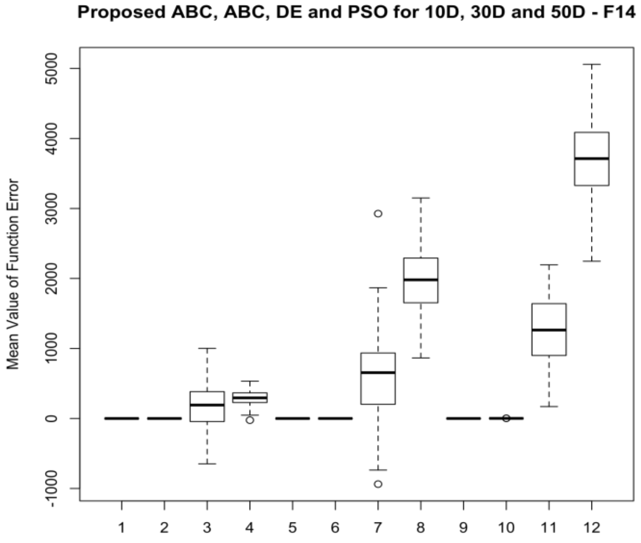

Fig. 5 Box-plot of function F14 with function errors of mean value of the proposed ABC, standard ABC, DE, and PSO algorithms on 10D, 30D, and 50D

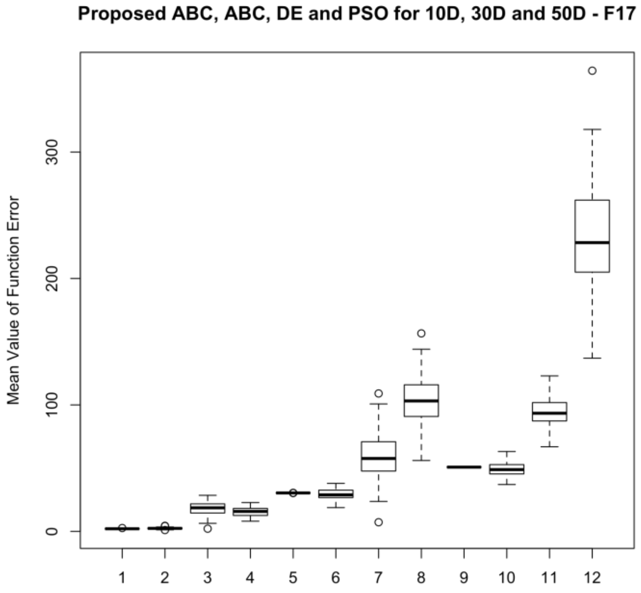

Fig. 6 Box-plot of function F17 with function errors of mean value of the proposed ABC, standard ABC, DE, and PSO algorithms on 10D, 30D, and 50D

In addition, the convergence performance of DE algorithm is better for functions F9 and F10 on 10D, as well as on functions F10, F12, and F20 on 50D.

Fig. 8–10 illustrate the box-plots for function errors of mean value of composition functions. According to Fig. 8, the convergence performances of the proposed ABC algorithm and the standard ABC algorithm are very similar, but much better than DE and PSO algorithms, the convergence performance of DE algorithm is competitive to PSO algorithm on 10D, 30D, and 50D for function F22.

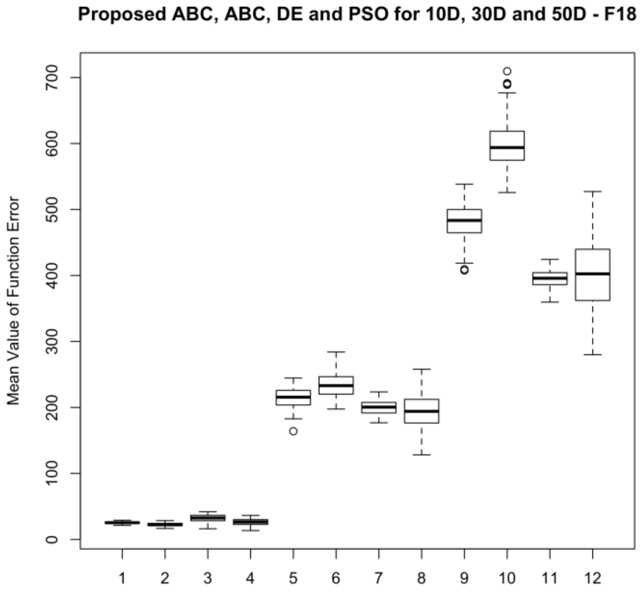

Fig. 7 Box-plot of function F18 with function errors of mean value of the proposed ABC, standard ABC, DE, and PSO algorithms on 10D, 30D, and 50D

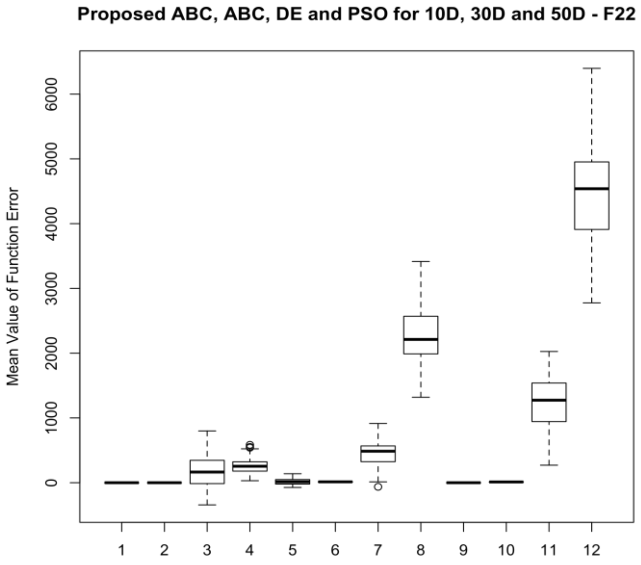

Fig. 8 Box-plot of function F22 with function errors of mean value of the proposed ABC, standard ABC, DE, and PSO algorithms on 10D, 30D, and 50D

Fig. 9 shows better convergence performance for the proposed ABC algorithm than standard ABC algorithm on 10D, and 50D for function F24, however, the convergence performance of DE algorithm is the best on 30D and 50D. In addition, the performance of proposed ABC and standard ABC algorithms is very similar and much competitive to DE and PSO algorithms. According to Fig. 10, the convergence performance of the proposed ABC algorithm is very similar to standard ABC algorithm. DE algorithm performs better than PSO algorithm on 10D, 30D, and 50D for function F27. For the remaining composition functions, we also could conclude that the convergence performance of our proposed ABC algorithm is better or similar to standard ABC algorithm; DE algorithm is competitive to PSO algorithm; however, PSO algorithm performs well for functions F21 and F23 on 10D, 30D, and 50D, as well as for function F26 on 10D and 30D, for function F28 on 10D and 50D.

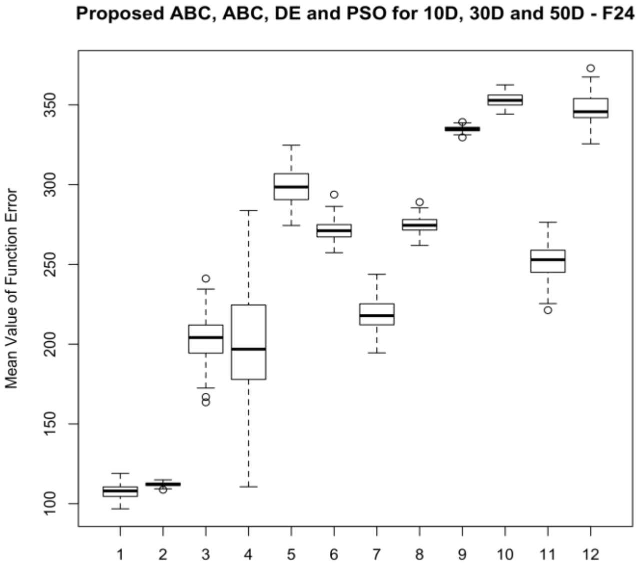

Fig. 9 Box-plot of function F24 with function errors of mean value of the proposed ABC, standard ABC, DE, and PSO algorithms on 10D, 30D, and 50D

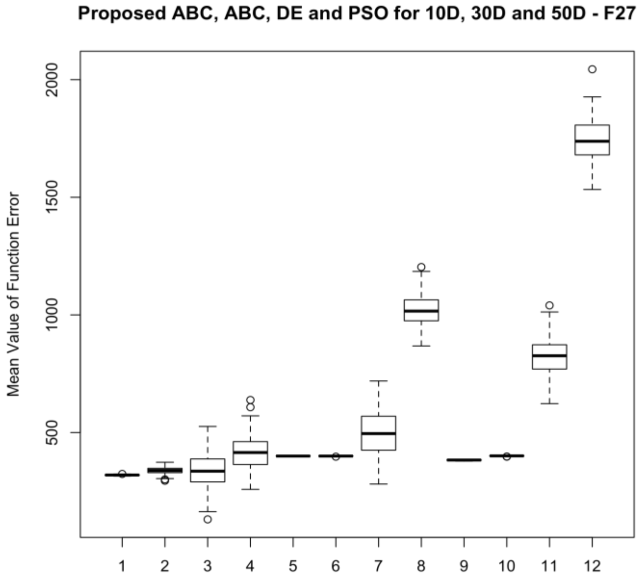

Fig. 10 Box-plot of function F27 with function errors of mean value of the proposed ABC, standard ABC, DE, and PSO algorithms on 10D, 30D, and 50D

According to the above analyses with the figures and tables, we can conclude that the proposed ABC algorithm and standard ABC algorithm are not effective on unimodal functions, especially, on functions F2, F3, and F4. For multimodal functions and composition functions, the convergence performance of the proposed ABC is the best as a whole, especially on 10D. The convergence performance of standard ABC algorithm is better than DE and PSO algorithms. The convergence performance of PSO algorithm is the worst. For all the dimension sizes of 10D, 30D, and 50D, functions F1, F5, and F11 reached the best convergence performance with best values of zero for function error.

V. CONCLUSIONS

In this paper, we have implemented comparative experiments of our proposed ABC and standard ABC algorithms using real parameter optimization benchmark problems defined by CEC 2013 test suite. We introduced a self adaptive mechanism, incorporated DE and PSO algorithms into standard ABC algorithm and proposed an improved hybrid ABC algorithm. We selected the best parameter settings through a number of initial comparison experiments and evaluated the performance of both algorithms on dimension sizes of 10, 30, and 50, respectively. The comparison experiments of our proposed ABC, standard ABC, DE and PSO algorithms were implemented. Statistical comparison analyses showed that the convergence performance of our proposed ABC algorithm was competitive or similar to standard ABC algorithm, quite better than DE and PSO algorithms on all the dimension sizes of 10, 30, and 50. The standard ABC algorithm performed better than DE and PSO algorithms. In addition, the performance of DE algorithm was better than PSO algorithm.

REFERENCES

[1] E. Bonabeau, M. Dorigo, and G. Theraulaz, Swarm Intelligence: From Natural to Artificial Systems, Oxford University Press, NY, pp. 1-21, 1999.

[2] J. Kennedy and R. Eberhart, “Particle swarm optimization,” IEEE International Conference on Neural Networks , pp. 1942-1948, 1995.

[3] R. Storn and K. Price, “Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces,” Journal of Global Optimization , Vol. 11, No. 4, pp. 341-359, 1997.

[4] M. Dorigo, M. Birattari and Stutzle, “Ant colony optimization: artificial ants as a computational intelligence technique,” IEEE Computational Intelligence Magazine, Vol. 1, No. 4, 2006.

[5] D. Karaboga, “An idea based on honey bee swarm for numerical optimization,” Tech. Rep. TR06, Erciyes Univ. Press, Erciyes, 2005.

[6] P. Civicioglu and E. Besdok, “A conceptual comparison of the cuckoo-search, particle swarm optimization, differential evolution and artificial bee colony algorithms,” Artificial Intell. Review, Vol. 39, No. 4, pp. 315-346, 2013.

[7] J. J. Liang, B. Y. Qu, P. N. Suganthan and A. G. Hernandez-Dıaz, “Problem definitions and evaluation criteria for the CEC 2013 special session on real-parameter optimization,” Computational Intell. Lab., Zhengzhou University, Zhengzhou, China, Tech. Rep. 201212 and Nanyang Technological University, Singapore, Tech. Rep., 2013.

[8] D. Karaboga and B. Basturk, “A powerful and efficient algorithm for numerical function optimization: Artificial bee colony(ABC) algorithm,” Journal of Global Optimization , Vol. 39, No. 3, pp. 459-471, 2007.

[9] D. Karaboga and B. Basturk, “On the performance of artificial bee colony (ABC) algorithm,” Applied Soft computing , Vol. 8, pp. 687-697, 2008.

[10] D. Karaboga and B. Akay, “A comparative study of artificial Bee colony algorithm,” Applied Mathematics and Computation , Vol. 214, No. 1, pp. 108-132, 2009.

[11] D. Whitley, “A genetic Algorithm tutorial,” Statistics and Computing , Vol. 4, pp. 65-85, 1994.

[12] D. Karaboga and B. Akay, “An artificial bee colony (abc) algorithm on training artificial neural networks,” 15th IEEE Signal Processing and Communications Applications , pp.1-4, 2007.

[13] S. K. Udgata, S. L. Sabat and S. Mini, “Sensor deployment in irregular terrain using artificial bee colony algorithm,” IEEE Congress on Nature & Biologically Inspired Computing , pp. 1309-1314, 2009.

[14] B. Akay and D. Karaboga, “Artificial bee colony algorithm for large-scale problems and engineering design optimization,” Journal of Intelligent Manufacturing, Vol. 23, No. 4, pp. 1001-1014, 2010.

[15] B. Alatas, “Chaotic bee colony algorithm for global numerical optimization,” Expert Systems with Applications, Vol. 37, pp. 5682-5687, 2010.

[16] G. Zhu and S. Kwong, “Gbest-guided artificial bee colony algorithm for numerical function optimization,” Applied Mathematics and Computation , Vol. 217, pp. 3166-3173, 2010.

[17] A. Banharnsakun, T. Achalakul and B. Sirinaovakul, “The best-so-far selection in artificial bee colony algorithm,” Applied Soft Computing, Vol. 11, No. 2, pp. 2888-2901, 2011.

[18] D. Karaboga and B. Gorkemli, “A combinatorial artificial bee colony algorithm for traveling salesman problem,” International Symposium on Innovation in Intelligent Systems and Applications (INISTA) , pp. 50-53, 2011.

[19] W. Gao and S. Liu, “A modified artificial bee colony algorithm,” Computers & Operations Research , Vol. 39. pp. 687-697, 2012.

[20] E. Rashedi, H. Nezamabadi-pour and S. Saryazdi, “GSA: a gravitational search algorithm,” Information Sciences , Vol. 179, No. 13, pp. 2232-2248, 2009.

[21] D. Swagatam and P. N. Suganthan, “Differential evolution: a survey of the state-of-the-art,” IEEE Trans. Evolotionary Comptation , Vol. 15, No. 1, 2011.

[22] A. K. Qin and X. Li, “Differential evolution on the CEC-2013 single-objective continuous optimization testbed,” Proc. of IEEE Congress on Evolutionary Computation, Cancun, Mexico, pp. 1099-1106, 2013.

[23] C. Stephen, M. James and B. R. Antonio, “Standard Particle Swarm Optimization on the CEC2013 Real-Parameter Optimization Benchmark Functions,” Tech. Rep., School of Information Technology, York University, 2013.