Enhanced Artificial Bee Colony for Real Parameter Optimization

Abdul G. Abro, Junita Mohamad-Saleh, and Noorazliza Sulaiman

***Abstract*—**Bio-inspired optimization algorithms are widely-used-subset of computational intelligence. The algorithms have adopted their working principles from different natural phenomena. Artificial Bee Colony (ABC) is formulated from honey bees foraging phenomenon. ABC optimization algorithm is a potent member of the algorithm-family. However, ABC suffers from slow convergence and poor exploitation. Researchers have evolved many ABC variants to overcome the demerits of ABC. Nevertheless, the variants are proposed either at the cost of additional control variables or required to evaluate the objective function numerous times for a food-source. The search of optimal values of the control variables to yield optimal performance of the ABC variants is exhaustive. Hence the variants are computationally intensive. This research proposes an enhanced ABC algorithm without adding any control-variable to the standard-ABC. Furthermore the proposed variant evaluates the objective function only once like the standard-ABC. The proposed algorithm has been compared with four efficient variants of ABC algorithm on a number of benchmark functions and a real-world application i.e. automatic-voltage-regulator performance enhancement. The two different statistical tests have been performed to assess significance of the proposed algorithm. The results suggest that not only the proposed algorithm has better capability to fend off local optima but it also possesses better convergence rate.

***Keywords*—**Artificial bee colony, ABC variant, foraging optimization algorithms, real parameter optimization

I. INTRODUCTION

As various real-world optimization problems are becoming severely difficult, global optimization using traditional techniques is becoming a profoundly challenging assignment [1]. Computational intelligence (CI), a driven branch of artificial intelligence (AI), is an area which solves most real-world problems efficiently and has yielded better results [2]. CI has been successfully applied in various disciplines [3]-[7]. Mainly, CI includes neural networks, bio-inspired optimization algorithms and fuzzy systems. Fuzzy systems are mostly confined to control applications. Nevertheless, bio-inspired optimization algorithms and neural networks cover very diverse engineering applications [8], [9]. Furthermore, the analysis carried out in reference [8] clearly claims that independent position of bio-inspired optimization algorithms in CI paradigms is increasing even more with time passage. The working principle of bio-inspired optimization algorithms is inspired by various natural phenomena. The algorithms are also known as population-based algorithms. There are different factors influencing the performance of the algorithms.

Bio-inspired optimization algorithms rely heavily on feedback for convergence and self-regulation [10]. Feedback is nothing but to induce self-influence or influence of any particular pattern available in the population. There are two types of feedbacks; one is called positive feedback and the other is known as negative feedback [11]. Positive feedback increases the occurrence tendency of any particular pattern. This accelerates convergence rate, but it could lead to instability [10], [12]. Negative feedback regulates impact of positive feedback as to maintain the equilibrium [10], [13].

The prime goal of honey bees during foraging is to optimize time spent on foraging and foraged energy. Artificial Bee Colony (ABC) algorithm uses foraging strategy of honey bees for evolving new possible-solutions from existing possible-solutions. It has been developed in 2007 [14]. Standard-ABC optimization algorithm is a popular and an efficient element of swarm-intelligence-based bio-inspired optimization algorithms [15], [16]. Standard-ABC has not only been able to yield better performance but it also has fewer parameters to tune [15], [17]. ABC optimization algorithm has been applied in a number of engineering applications [18]-[22]. Nonetheless, standard-ABC possesses few demerits like poor exploitation capability [23], slow convergence on unimodal problems [11], [24] and can easily be trapped local optima while handling multimodal problems [25]. The faster convergence and the ability to fend-off local optima are the motives in ABC algorithm research area. However, no research work has been proposed to achieve the motives simultaneously [25]. Moreover, the previously proposed solutions either require evaluation of objective function many times or need initialization of additional control variables. Hence, the solutions are computationally intensive.

The additional control variables serve to regulate the impact of positive feedback and hence, the induction of negative feedback. This research proposes a simple yet effective

This work was supported in part by the Ministry of Higher Education (MOHE) FRGS Grants No: 203/PELECT/6071247 and Universiti Sains Malaysia.

A. G. Abro is with the College of Applied Engineering, Al Muzahimiyah Campus, King Saud University, Riyadh, 11451 Kingdom of Saudi Arabia (e-mail: ghaniabro@gmail.com).

J. Mohamad-Saleh is with School of Electrical & Electronic Engineering, Universiti Sains Malaysia, Nibong Tebal, 14300 Penang, Malaysia (corresponding author phone: 604-599-6027; e-mail: jms@ usm.my).

N. Sulaiman is with the of Electrical & Electronic Engineering, Universiti Sains Malaysia, Nibong Tebal, 14300 Penang, Malaysia (e-mail: noorazliza_sulaiman@usm.my).

solution to avert the demerits of standard-ABC without adding any additional control variable. The proposed technique relies on top-three global-best possible-solutions rather than only the global-best possible-solution. Therefore, the algorithm capitalizes on the global best possible-solutions for faster convergence and concurrently the swarm does not accumulate at one location of search space. Hence, the search-space is properly explored in the search of global optimum. The proposed algorithm has been compared with other existing variants of ABC algorithm on nineteen commonly used benchmark functions. The results show better performance of the proposed variant of ABC algorithm in comparison to existing variants of the algorithm.

This paper is divided into eight sections. The immediate section demonstrates the working principle of standard-ABC algorithm. The third section is related to literature survey followed by the section presenting the proposed modifications. The fifth section presents parameter settings of the compared algorithms. Then results are compared and discussed in the following section. Finally application of the proposed algorithm to a real world application has been presented in the seventh section. The conclusion is then presented in eighth section.

II. Artificial Bee Colony Algorithm

There are three classes of bees associated to foraging task [11], [14]. One is called Employed-Bees (EBs), which are actually assigned food-sources around the hive. The food-sources represent the possible-solutions and hive is the search-space. The second type of bees is called Onlooker-Bees (OBs). Onlooker-bees wait in dancing area of the hive for seeking information from EBs of the food-sources having higher nectar-amount [16]. The nectar-amount of food-source corresponds to the quality or fitness of possible-solution. The third type of bees is known as Scout-Bees (SBs). If any possible-solution does not show any improvement over a preset number of iterations, the possible-solution is abandoned. The preset number of iteration is controlled by user-defined control variable named *limit* [25]. The EB associated with the abandoned food-source becomes SB. Then the SB flies around the hive for picking a new food-source [17].

The standard-ABC algorithm starts with random initialization of the food-sources around the hive. Each EB is assigned a food-source. Hence, the number of EBs and the food-sources is equal. Moreover in standard-ABC, the number of EBs and OBs is same. The function of EBs is to explore neighborhood of the associated food-sources using the following equation:

$$z_{ij} = y_{ij} + \varphi_{ij}\left(y_{ij} - y_{kj}\right) \tag{1}$$

where y_{ij} represents j_{th} index-magnitude of i_{th} food-source, y_{kj} represents j_{th} index-magnitude of k_{th} food-source, i and k are mutually exclusive food-sources, j Є [1,2,…. D], D is the dimension of search space and φ is a uniformly distributed random number within [-1, 1].

Each food-source has the number of indices equal to the dimension of a problem. The standard-ABC algorithm updates only one index of every food-source during EBs as well as OBs stage of the algorithm. Hence, for a food source objective function is evaluated only once in an iteration. The fitness of the modified food-source, also called candidate food-source, is calculated using the following equation:

$$fit_i = \begin{cases} \dfrac{1}{1+ fit_i}, & f_i \geq 0, \\ 1 + abs(f_i), & f_i < 0 \end{cases} \tag{2}$$

where f_i represents objective function value of i_{th} food-source and fit_i is the corresponding fitness value after transformation.

The fitness value of the candidate and the old food-source is compared. The food-source having higher fitness value is retained [14]. This is known as greedy selection. As mentioned earlier, OBs wait in the dancing area of the hive for seeking information of the explored food-sources. OBs probably select the food-sources having higher fitness values [16]. In standard-ABC, the probability of the food-source having higher fitness value is calculated by the following equation:

$$\mathrm{P}_i = \frac{fit_i}{\sum_{j=1}^{NS} fit_j} \tag{3}$$

where *NS* represents the number of food-sources and *i* is the selected food-source.

If an abandoned food-source exists then the SB is assigned a randomly-initialized food-source to replace the abandoned food-source. In standard-ABC algorithm the number of SB is one. In standard-ABC, EBs and OBs exploit search space whereas SB adds exploration capability in the algorithm [16]. The SB is assigned a food-source by the equation given below:

$$y_{ij} = y_j^{\min} + rand(0,1)\left(y_j^{\max} - y_j^{\min}\right) \tag{4}$$

$_j^{min}$ is lower limit of search space and $_j^{max}$ is upper bound of search space.

Block diagram representation of standard-ABC algorithm is given in Fig. 1.

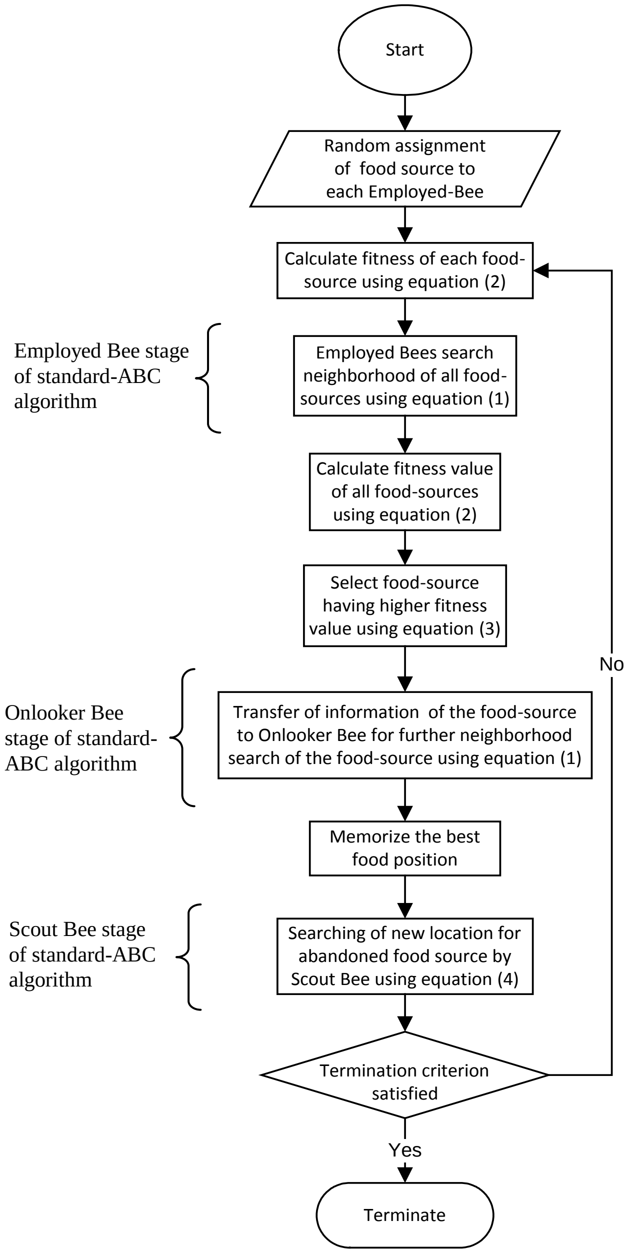

Fig. 1 ABC algorithm flowchart

III. Previous Work and Limitations

For enhancing the performance of standard-ABC, a number of researches have been carried out. References [26], [27] have modified standard-ABC for discrete parameter optimization. Researchers [24], [25], [28], [29] have incorporated chaotic maps in the standard-ABC and [1] has hybridized the standard-ABC and rosenbrock method for finding optimal solutions efficiently. Whenever chaotic-maps or rosenbrock methods have been incorporated into any optimization algorithm, the hybridized algorithm has performed better than the standard optimization algorithm [1], [24], [25], [28]-[30]. Therefore, it can be prophesized that incorporation of the techniques into the algorithm, proposed in this research work, will yield better results too. However, the incorporation for performance enhancement of the optimization algorithms, adds computational cost [1], [24].

Akay has proposed two modifications in the standard-ABC [11] at the cost of additional user-defined control variables. The proposed algorithm is computationally very intensive. Since it has to evaluate objective function numerous times for a food-source, in each iteration, on contrary to standard-ABC algorithm where objective function is evaluated only once in every iteration for a food-source. In modified-ABC algorithm, the number of objective function evaluations is proportional to the problem-dimension. Moreover results depict the standard-ABC has performed better than modified-ABC on various functions.

The proposals presented in [24], [25], [31]-[33] have inducted influence of the best possible-solution for enhancing the fitness of other possible-solutions. Another component has been added in the search equation of the standard-ABC [30]. The added component contains the influence of best possible-solution and a user-defined control-variable. The modified ABC has been named gbest-guided ABC (GABC) and has the following search equation.

$$z_{ij} = y_{ij} + \varphi_{ij}\left(y_{ij} - y_{kj}\right) + \psi\left(y_{ij} - x_{best}\right) \quad (5)$$

where y_i and y_k represent mutually exclusive food-sources, x_{best} is the globally-best food-source, j Є [1,2,…. D], D is the dimension of search space, φ is a uniformly distributed random number within [-1, 1], ψ_{ij} is the positive random integer within [0-C] and C is a user defined constant.

The performance of GABC has been compared with the standard-ABC. The results have shown better performance of GABC. GABC has been taken for the comparative analysis carried out in this research work.

The enhancement of each possible-solution using three different search-equations, in every generation, makes the algorithm proposed in [32] computationally very intensive. The proposed algorithm has been named Prediction-Selection-ABC (PS-ABC). Out of the three equations, two equations are of standard-ABC and GABC algorithms and the third equation, proposed in the research, is given below:

$$z_{ij} = y_{ij} w_{ij} + 2\left(\varphi_{ij} - 0.5\right)\left(y_{ij} - y_{kj}\right)\phi_1 + \psi\left(x_{best} - y_{kj}\right)\phi_2 \quad (6)$$

where y_i and y_k are mutually exclusive food-sources, x_{best} is the global-best food-source, w_{ij} = Φ_1 = 1/(1+exp(-fitness(y_i)/fitness(x_{best})), Φ_2 = 1 (for employed bees), Φ_2 = 1/ (1+exp (-fitness (i) / fitness (best)) (for onlooker bees) and, Φ_{ij} and Ψ_{ij} are random number with [0, 1].

Equation (6) is highly prone to local optima traps, as stated in [32]. Equation (6) generates candidate possible-solution around a percentage of existing possible-solution, as value of “w” is always less than 1. Hence, the equation may only work for the functions having optimum solution at zero. On contrary, real world applications do not have optimal solutions at zero value.

PS-ABC algorithm has been run on a real-world application, presented in Section-7, to assess its dependence on GABC search-equation, i.e. (5). PS-ABC algorithm has been run on four different C-values. Fig. 2 presents the convergence rate plots of PS-ABC algorithm at four different C-values of GABC search equation. The convergence rate plots are averaged over 30 runs. The figure clearly depicts huge dependence of PS-ABC algorithm on GABC search equation. The change in the C-value of GABC search equation has brought change in convergence rate of PS-ABC algorithm. Therefore, it can be concluded that the PS-ABC algorithm heavily rely on GABC search equation. GABC has been taken for the comparative analysis carried out in this research work.

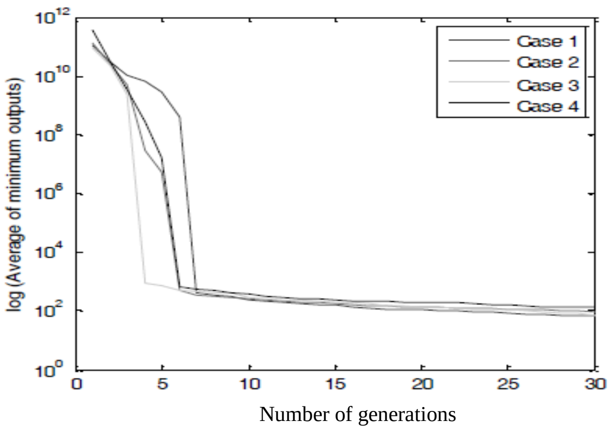

Fig. 2 Convergence rate plot of PS-ABC

The research work presented in [24] has proposed two different equations inspired by the mutation-equation of differential evolution. The algorithm has been named Improved-ABC (IABC). Following are the two proposed equations:

$$z_{ij} = x_{best,t} + \varphi_{ij}\left(y_{l,t} - y_{m,t}\right) \quad (7)$$

$$z_{ij} = y_{m,t} + \varphi_{ij}\left(y_{i,t} - y_{n,t}\right) \quad (8)$$

where y_i, y_m and y_n are mutually exclusive food-sources and x_{best} is the globally best food-source, t Є [1,2,….T], T is a selected-integer from [1,2,….D], *D* is the dimension of search space and φ is a uniformly distributed random number within [-1, 1].

IABC is required to solve (7) or (8) systematically, at the cost of a user-defined control-variable called Probability (*P*). It is suggested that if the randomly generated number is greater than the P-value (i.e. rand > P-value) then solve (7) otherwise solve (8). The results demonstrate significantly better performance of IABC among all compared algorithms.

Best-so-far ABC (BSO) algorithm proposed in [33] has enhanced Onlooker-Bees (OBs) and Scout-Bee (SB) stages than the standard-ABC. In BSO, the number of objective function evaluations is equal to the number of problem-dimension for a food-source in an iteration i.e. BSO evaluates one hundred times the objective function for a food-source, in every iteration, if the problem has one hundred dimensions. On contrary, the standard-ABC evaluates the objective function only once for a food-source, in an iteration, irrespective of problem dimension. Therefore, BSO is also computationally very intensive. Furthermore, it is stated that the proposed equation has the tendency of local optima traps. Whereas the modification proposed for SB may only work for the functions having optimal solution at zero. Since BSO generates possible-solution for SB by reducing the abandoned possible-solution to almost 80% or lesser of the original value. Nevertheless the reduction depends upon user-defined control variables. Certainly more reduction will result in better convergence.

Globally best ABC, proposed in [25], has search-equations inspired by the mutation-equations of the differential evolution algorithm. The proposed equations are given below:

$$z_{ij} = x_{best,j} + \varphi_{ij}\left(y_{mj} - y_{nj}\right) \quad (9)$$

$$z_{ij} = x_{best,j} + \varphi_{ij}\left(y_{mj} - y_{nj}\right) + \varphi_{ij}\left(y_{kj} - y_{rj}\right) \quad (10)$$

where y_m, y_n and y_k are mutually exclusive food-sources and x_{best} is the globally best food-source, j Є [1,2,…. D], *D* is the dimension of search space and φ is a uniformly distributed random number within [-1, 1].

The algorithm based on (9) has been named ABC/best/1 (BABC1) and the algorithm based on (10) has been called ABC/best/2 (BABC2). Both algorithms have been considered to compare with the algorithm proposed in this research work.

IV. Proposed Enhanced ABC

The literature suggests that incorporation of the best possible-solution in the search equation of the standard-ABC has increased convergence rate. However, it also inducts the tendency for premature convergence. The premature convergence tendency has been tackled by introducing different control variables. As stated earlier, these control variables have performance-deciding-impact on the performance of algorithms. Moreover, there is no systematic way to decide upon the optimal values of the variables. This research work proposes Enhanced-ABC algorithm (EABC). The proposed algorithm does not require any extra control variable. The search equation of the proposed algorithm is given below:

$$z_{ij} = y_{ij} + \varphi_{ij}\left(y_{ij} - y_{best,/second-best/third-best,j}\right) \quad (11)$$

where $y_{best/second\text{-}best/third\text{-}best,\ j}$ is jth index of one of top-three food sources.

In EABC algorithm, the possible-solutions have been divided into three groups irrespective of fitness value of each possible-solution. Possible-solutions are also called food-sources. Each one of the top-three global best possible-solutions has been exploited to enhance possible-solutions present in each group, regardless of the presence of global-best possible-solution in the group. Whereas top-three possible-solutions have been updated by randomly chosen possible-solution, if top-three solutions are present in their respective groups.

Reliance on the single global-best possible-solution may accumulate the swarm in one location of the search space. However, the application of the top-three possible-solutions drives to avoid the accumulation of the swarm at any single location of the search space. Thus, the algorithm capitalizes on the best possible-solutions for accelerating convergence rate. Simultaneously the algorithm has the capability to avert local optima. Utilization of the best possible-solution can be seen as induction of positive feedback because it increases the occurrence tendency of the pattern. Whereas relying on the multiple best possible-solutions rather than the single best possible-solution can be seen as the induction of negative feedback. Since the technique regulates the impact of positive feedback.

Different combinations have also been tried to optimize the structure of the proposed algorithm. For example, four or more groups capitalizing on the top-four or more global-best possible-solutions have also been tried, but the results had been worse. Since as go down to fourth-best, fifth-best and so on, the fitness of possible-solutions deteriorates, hence curtails convergence rate and make the algorithm prone to local optima traps. Fig. 3 shows flow chart of the proposed ABC algorithm. As far as computational complexity of the proposed algorithm is concerned, the proposed algorithm performs the same number of fitness function evaluation as performed by the standard algorithm in each iteration. However, the proposed algorithm sorts out the best food-source in each iteration which is not done in the standard ABC algorithm. Moreover, computational complexity of the proposed algorithm is either lesser or similar to the compared algorithms e.g. it is similar to BABC1, GABC and lesser than IABC and BABC2.

V. Experimental Set-up and Parameter Settings

In this research work standard-ABC (ABC), Gbest-guided-ABC (GABC), Improved-ABC (IABC), ABC/best/1 and ABC/best/2 have been taken to compare the performance of the proposed variant of ABC named Enhanced-ABC (EBAC). GABC has been run at *C* equals to 0.5, 1.0, 1.5 and 2.0, to get optimal performance of GABC. Similarly, IABC has been run on *P* equals to 0.15, 0.25 and 0.35 and *T* has been fixed to one. Values of the algorithm specific control variables have been tried and set as instructed in their respective research works. The performance of all the considered algorithms has been analyzed using eighteen commonly used complex benchmark functions listed in Table 1 in Appendix 1 [1], [11], [14], [31], [33], [34]. Optimization algorithm needs to have balanced search strategy in terms of exploration and exploitation for optimizing complex multimodal functions like Griewank, Ackley and Schwefel functions [11], [16]. Rosenbrock and Colville have a narrow and very deep curving valley. Therefore, the algorithm is required to explore search-space properly and keep on changing directions continuously [1].

The last four functions are four-dimensional and rests are thirty-dimensional functions. The colony size of all algorithms has been fixed to 100 for thirty and 30 for ten dimensional functions. *limit* (i.e. the control variable) has been set to 100 and 30 for thirty and four dimensional functions respectively. The number of generations has been limited to 1500 and 200 for thirty and four dimensional functions respectively. Each algorithm has been run 30 times on each function to observe the variability of the algorithms. Random initialization has been used for all compared algorithms to make the comparison even. The performance analysis of the algorithms has been carried out on the basis of maximum, minimum, average over 30 runs and standard deviation among 30 outputs. The standard deviation predicts the robustness, the average value prophecies the convergence of the algorithms. The maximum and the minimum value give the worst and the best convergence of the algorithms. For obtaining statistical significance of the results, Wilcoxon-test and t-test have been performed. Moreover, data for error obtained by running all compared algorithms have been plotted using boxplot methods on a few functions as suggested in [49].

VI. Results and Discussion

Table 2 in Appendix 1 gives the performance-analysis results of GABC at the four different values of C. The results vividly depict the influence of C-value on the performance of GABC. The results do not show any systematic variation of GABC output in relation to the C-value. Therefore, GABC has to be run on different C-values for producing the optimal performance of GABC. The process is very exhaustive. GABC yielding the best convergence rate have been considered for further comparison and have been highlighted for clarity.

Table 3 in Appendix 2 presents the performance-analysis results of IABC at the three different values of *P* control variable. The results clearly depict the influence of *P* on the performance of IABC. The results do not show any systematic variation of IABC output in relation to *P* value. Hence different *P* values have to be tried for producing the optimal performance of IABC. The process is very exhaustive. IABC yielding the best convergence rate have been considered for further comparison and have been highlighted for clarity.

Figs. 4 to 21 show the convergence rate comparison plots of standard-ABC and its variants, on various benchmarks functions. The performance of every considered algorithm varies from the best to the worst depending upon benchmark function nature.

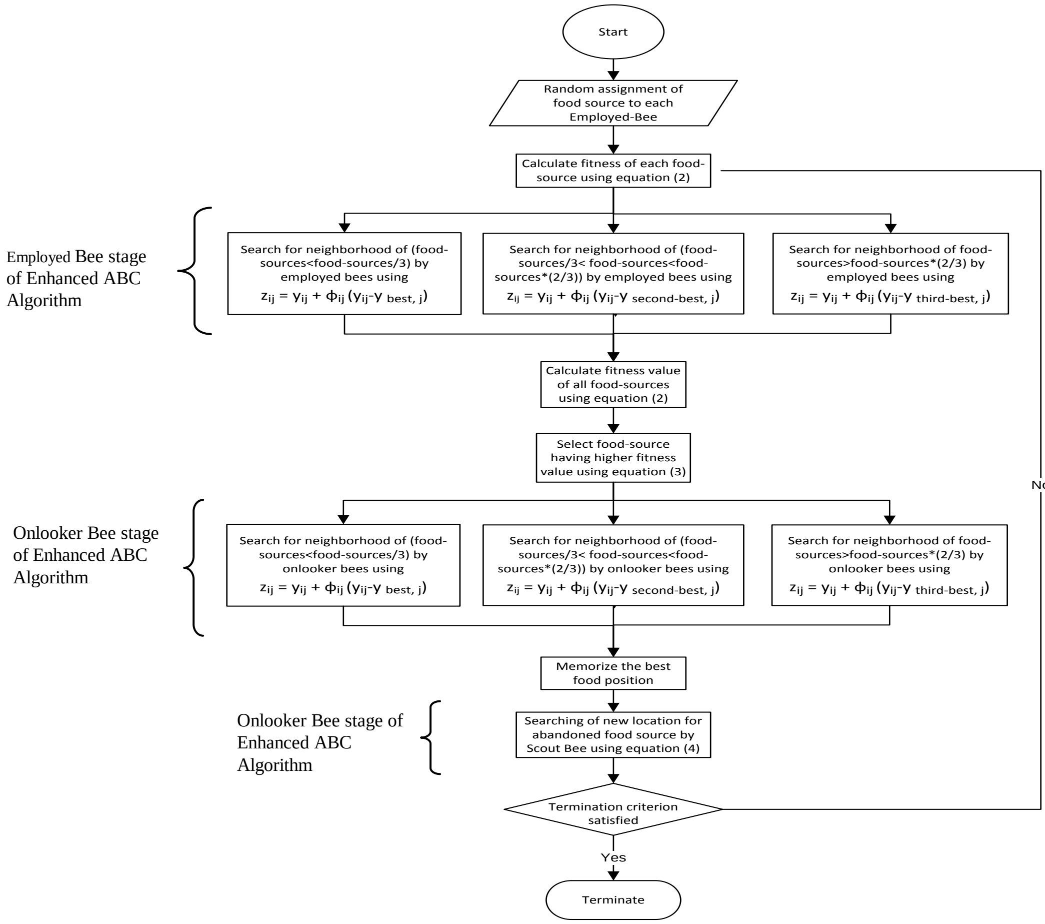

Fig. 3 Flowchart of the proposed ABC optimization algorithm

Table 1 Benchmark functions used to assess performance of optimization functions

No	Function Name	Equation	Initialization Range
f_1	Griewank (MN)	$f(x)=\frac{1}{4,000}(\sum_{i=1}^{D}(x_i^2))-(\prod_{i=1}^{D}\cos(\frac{x_i}{\sqrt{i}}))+1$	[-600 600]
f_2	Rosenbrock (UN)	$f(x)=\sum_{i=1}^{D}100(x_i^2-x_{i+1})^2+(1-x_i)^2$	[-15 15]
f_3	Ackley (MN)	$f(x)=20+e-20e^{(-0.2\sqrt{\frac{1}{D}\sum_{i=1}^{D}x_i^2})}-e^{\frac{1}{D}\sum_{i=1}^{D}\cos(2\pi x_i)}$	[-32 32]
f_4	Schwefel (MN)	$f(x)=D\times 418.9829+\sum_{i=1}^{D}-x_i\sin(\sqrt{\lvert x_i\rvert})$	[-500 500]
f_5	Himmelblau (MS)	$f(x)=\frac{1}{D}\sum_{i=1}^{D}(x_i^4-16x_i^2+5x_i)$	[-600 600]
f_6	Step (US)	$f(x)=\sum_{i=1}^{D}\left(\lfloor x_i+0.5\rfloor\right)^2$	[-100 100]
f_7	Bohachevsky (MN)	$f(x)=\sum_{i=1}^{D-1}x_i^2+2x_{i+1}^2-0.3\cos(3\pi x_i)\cos(3\pi x_{i+1})+0.3$	[-100 100]
f_8	Schwefel2 (UN)	$f(x)=\sum_{i=1}^{D}\lvert x_i^2\rvert+\prod_{i=1}^{D}\lvert x_i^2\rvert$	[-100 100]
f_9	Zakharov1 (UN)	$f(x)=\sum_{i=1}^{D}x_i^2+\left(\sum_{i=1}^{D}0.5x_i\right)^2+\left(\sum_{i=1}^{D}0.5x_i\right)^4$	[-20 20]
f_{10}	Schwefel Ridges (UN)	$f(x)=\sum_{i=1}^{D}\left(\sum_{j=1}^{i}x_j\right)^2$	[-100 100]

f_{11}	Schwefel Ridges with Noise (UN)	$f(x)=\sum_{i=1}^{D}\left(\sum_{j=1}^{i}x_j\right)^2*(1+0.4\lvert N(0,1)\rvert)$	[-100 100]
f_{12}	Shifted Rotated Elliptic (UN)	$f(x)=\sum_{i=1}^{D}\left(10^6\right)^{\frac{i-1}{D-1}}\left(x_i^2-10\right)-450$	[-100 100]
f_{13}	Random Shifted Sphere (US)	$f(x)=\sum_{i=1}^{D}\left(x_i-50*\lvert N(0,1)\rvert_i\right)^2$	[-100 100]
f_{14}	Random Shifted Griewank (MN)	$f(x)=\frac{1}{4{,}000}(\sum_{i=1}^{D}(x_i\text{-}50*\lvert N(0,1)\rvert_i)^2)\text{-}(\prod_{i=1}^{D}\cos(\frac{x_i\text{-}50*\lvert N(0,1)\rvert_i}{\sqrt{i}})\text{+}1$	[-600 600]
f_{15}	Schaffer (MN)	$f(x)=0.5+\frac{\left(\sin\sqrt{\sum_{i=1}^{4}x_i^2}\right)^2-0.5}{\left(1+0.001(\sum_{i=1}^{4}x_i^2)\right)^2}$	[-100 100]
f_{16}	Colville (UN)	$f(x)=100\left(x_1^2-x_2\right)^2+\left(x_1-1\right)^2+(x_3-1)^2+90(x_3^2-x_4)^2$ $+10.1\left(\left(x_2-1\right)^2+\left(x_4-1\right)^2\right)+19.8\left(x_2-1\right)\left(x_4-1\right)$	[-10 10]
f_{17}	Zakharov2 (UN)	$f(x)=\sum_{i=1}^{4}x_i^2+\left(\sum_{i=1}^{4}0.5ix_i\right)^2+\left(\sum_{i=1}^{4}0.5ix_i\right)^4$	[-10 10]
f_{18}	Quartic (US)	$f(x)=\sum_{i=1}^{4}ix_i^4+\lvert N(0,1)\rvert$	[-1.28 1.28]

However the results analysis shows that the proposed best among the considered algorithms. Wherever the proposed algorithm has performed second-best, the difference between the best and the proposed algorithm is marginal.

The standard-ABC algorithm has performed worst on all the functions than all the algorithms except on Rosenbrock, Schafer, Zakharov and Schwefel's Ridges functions. On all these functions, the standard-ABC has produced third-best response. However, standard-ABC has produced the worst response on all multimodal functions, except Schafer function, than the other compared algorithm. GABC has produced the best convergence rate on Noisy Schwefel's Ridges function among all the compared algorithms. However, the response of GABC on the multimodal functions is the poorer in comparison to its response on the unimodal functions. This is because of the excessive self-reinforcement in GABC search equation. Nevertheless, on all the considered functions, GABC has performed better than the standard-ABC.

IABC algorithm has produced the worst response on Colvile, Schafer, Zakharov, Quartic and Rosenbrock functions, in comparison to all the other algorithms. Nonetheless, IABC has produced the best response on Schwefel function than all the algorithms. IABC has yielded better response on the multimodal functions than on the unimodal functions. Therefore, it can be concluded that IABC has the tendency to produce better response on the multimodal functions in comparison to the unimodal functions.

BABC2 has produced poorer response on all the functions in comparison to BABC1, except Schwefel function. The convergence rate plots of BABC1 and the proposed algorithm have overlapping regions on Ackley and Quartic functions. However on Rosenbrock, both Schwefel's Ridges, both Zakharov, and on Schaffer functions BABC1 has shown very little convergence. BABC1 and BABC2 have produced better algorithm has either performed the best or second-response on the multimodal functions in comparison to the unimodal functions. Therefore, it can be deduced from this comparative analysis that both BABC algorithms have the tendency to produce better response only on the multimodal functions than on the unimodal functions. The BABC algorithms produce response around best possible-solution only. Therefore, the algorithms are highly prone to the local optima trappings, specifically, when the global optimum lies in the deep and curving valley.

The proposed algorithm, EABC, has produced the best convergence rate on all functions except Schwefel and Noisy Schwefel's Ridges functions. On Schwefel function, EABC has produced better response only in comparison to GABC and ABC algorithms. EABC has yielded the best response second to GABC, on Noisy Schwefel's Ridges function. EABC is joint winner on Ackley (Fig. 6) and Quartic (Fig. 21) functions along with BABC1. Therefore it can be concluded that the proposed algorithm has equal tendency to produce the best response on the multimodal and the unimodal functions.

Table 4 in Appendix 3 gives the comparative performance analysis results of all the considered algorithms on all the functions. The table replicates the aforementioned discussion. The algorithms producing the best and second-best results have been highlighted for clarity. Furthermore, the table presents the statistical tests results obtained using performance data of all compared algorithms. The tests results show that the performance of the proposed algorithm is significantly better than the compared algorithms. This is also evident from boxplot results depicted in Figs. 22 to 25.

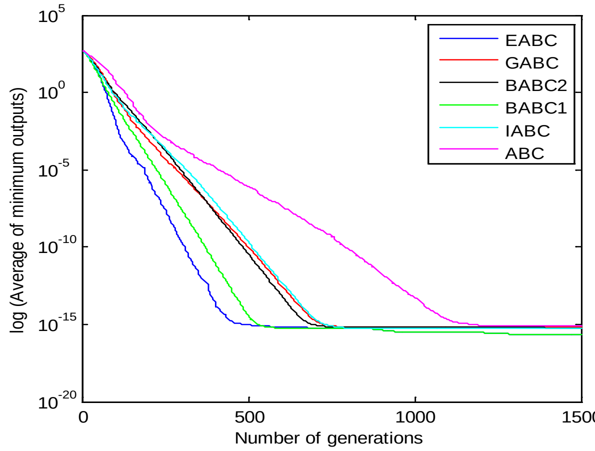

Fig. 4 Convergence rates of the algorithms, tested on Griewank function

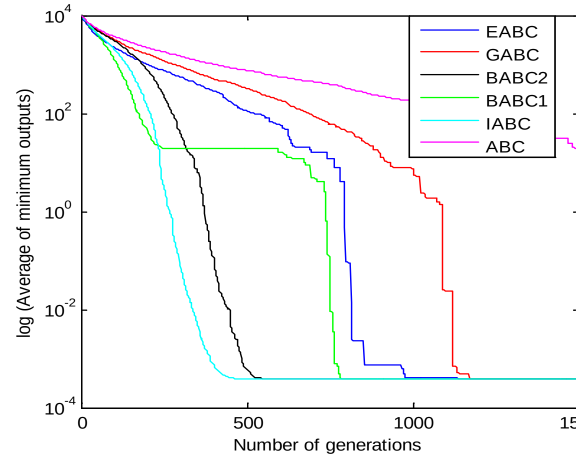

Fig. 7 Convergence rates of the algorithms, tested on Schwefel function

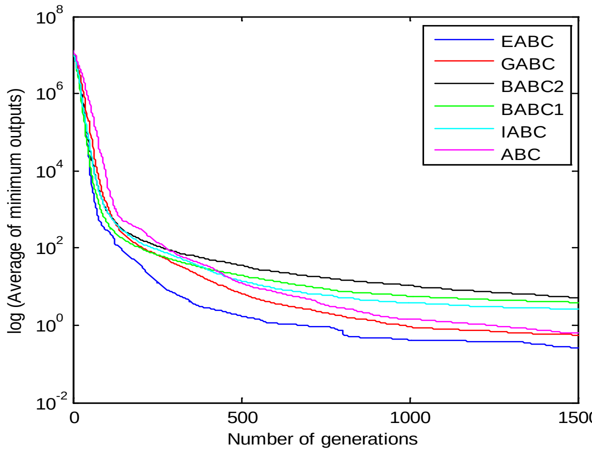

Fig. 5 Convergence rates of the algorithms, tested on Rosenbrock function

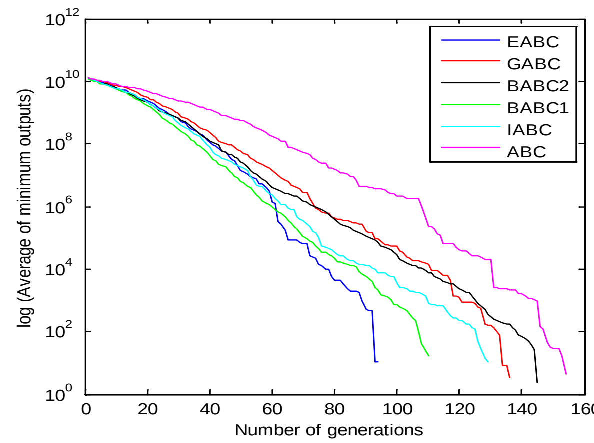

Fig. 8 Convergence rates of the algorithms, tested on Himmelblau function

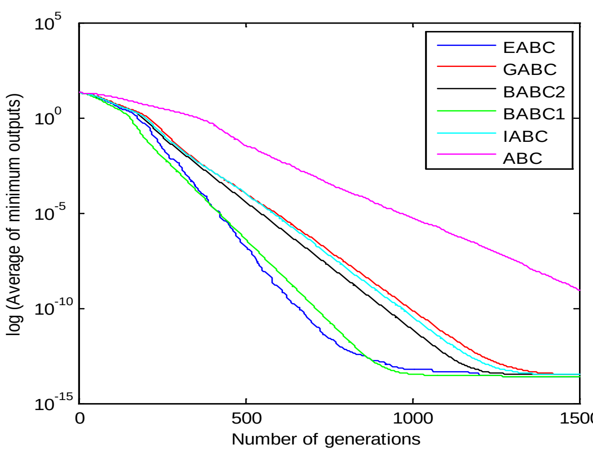

Fig. 6 Convergence rates of the algorithms, tested on Ackley function

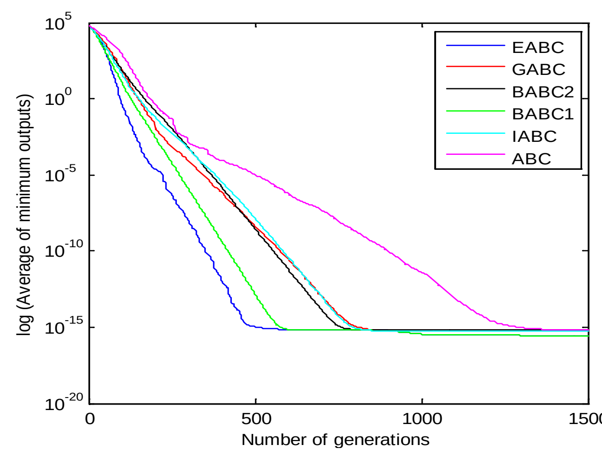

Fig. 9 Convergence rates of the algorithms, tested on Step function

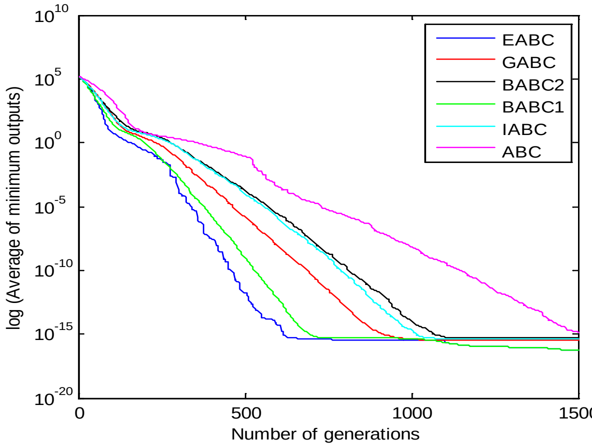

Fig. 10 Convergence rates of the algorithms, tested on Bohachevsky function

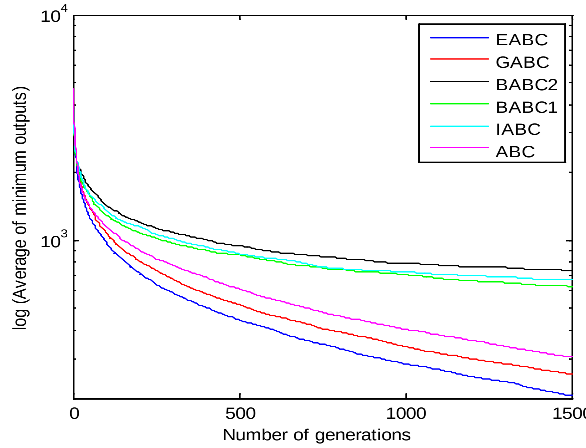

Fig. 13 Convergence rates of the algorithms, tested on Schwefel's Ridges function

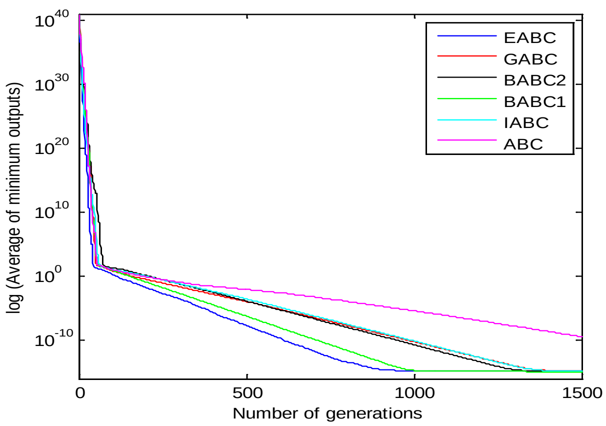

Fig. 11 Convergence rates of the algorithms, tested on Schwefel2 function

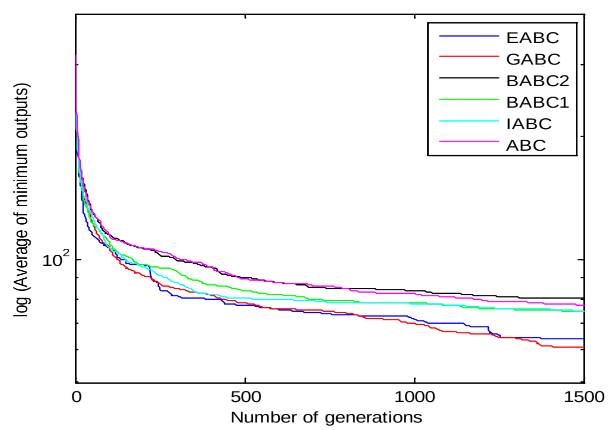

Fig. 14 Convergence rates of the algorithms, tested on Noisy Schwefel's Ridges function

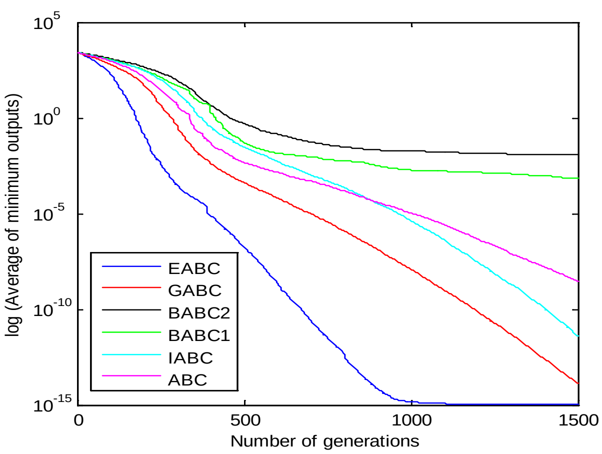

Fig. 12 Convergence rates of the algorithms, tested on Zakharov1 function

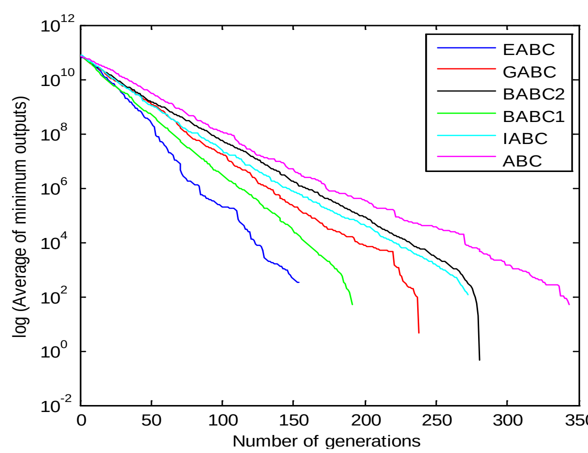

Fig. 15 Convergence rates of the algorithms, tested on Shifted Rotated Elliptic function

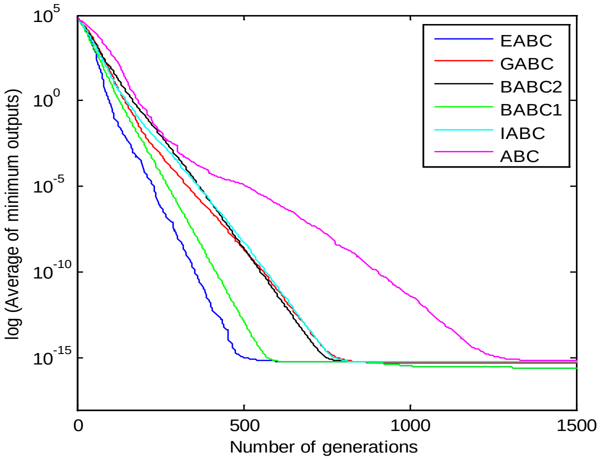

Fig. 16 Convergence rates of the algorithms, tested on Random Shifted Sphere function

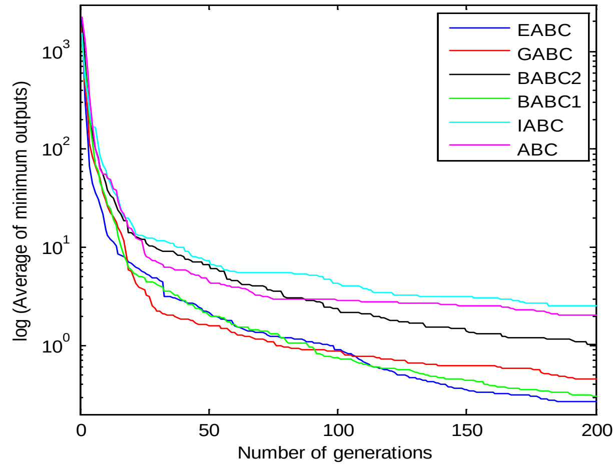

Fig. 19 Convergence rates of the algorithms, tested on Colvile function

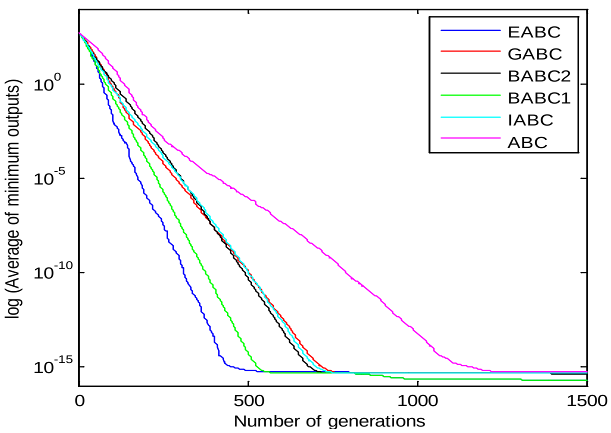

Fig. 17 Convergence rates of the algorithms, tested on Random Shiftted Griewank function

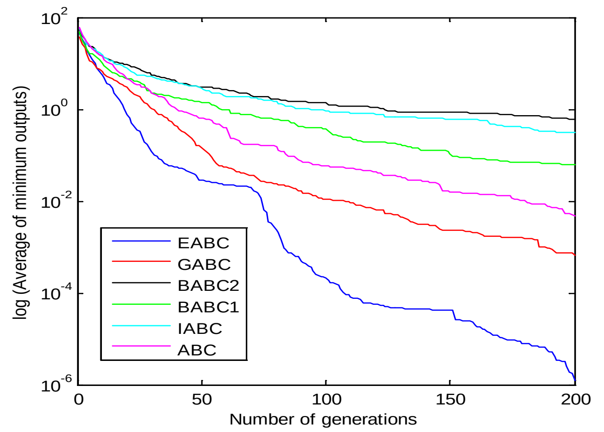

Fig. 20 Convergence rates of the algorithms, tested on Zakharov function

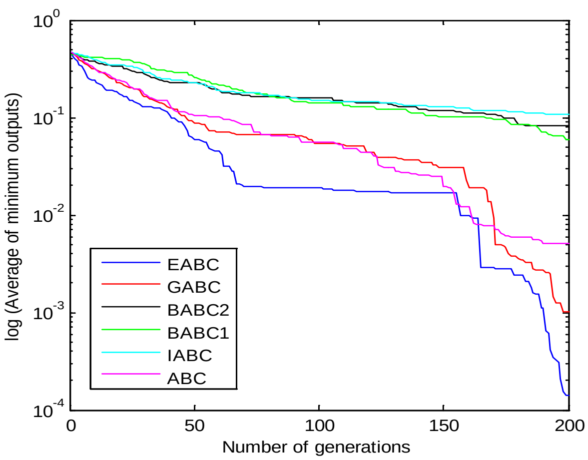

Fig. 18 Convergence rates of the algorithms, tested on Schafer function

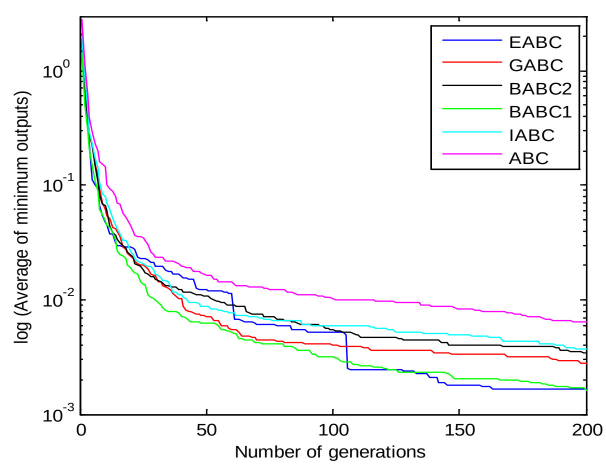

Fig. 21 Convergence rates of the algorithms, tested on Quartic

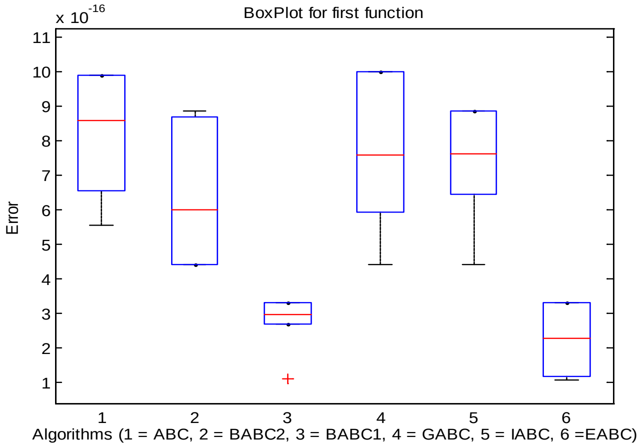

Fig. 22 Box-plotting of error of the compared algorithms over function 1

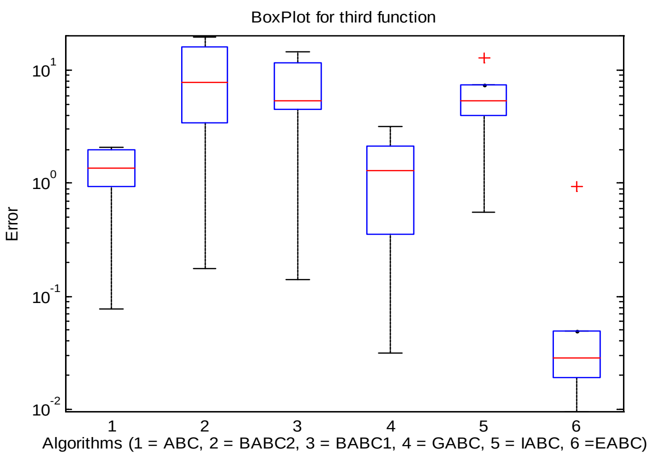

Fig. 23 Box-plotting of error of the compared algorithms over function 3

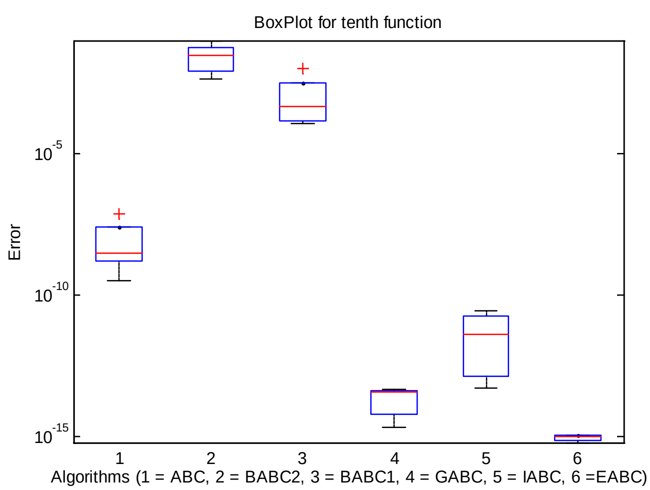

Fig. 24 Box-plotting of error of the compared algorithms over function 10

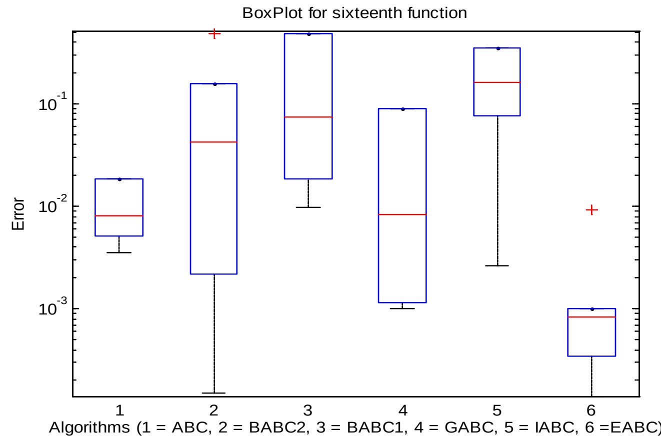

Fig. 25 Box-plotting of error of the compared algorithms over function 16

VII. Application of Proposed Algorithm

A. Introduction

It is very difficult to classify the real world problems in terms of separable and non-separable, and unimodal and multimodal problems. Moreover the setting of control variables requires expertise as in case of Bacterial Foraging Optimization Algorithm (BFOA) or exhaustive as in case of GABC or IABC. On contrary, the algorithm requiring lesser expertise and easy implementation suits application engineers the most. In this research work Automatic Voltage Regulator (AVR) performance enhancement is taken as a real-world problem for the comparative analysis.

Currently, bio-inspired optimization algorithms are very popular among application engineers from various disciplines [35]-[38]. The algorithms have resulted in better solutions of real-world problems, with reasonable computational effort, where analytical methods have failed [36]. However, the application of bio-inspired optimization algorithms is still limited in the field of online tuning of controller parameters [39]. On the other hand, Neural Networks (NNs) have been successfully applied in the field though NN with gradient-based learning algorithms are highly prone to local optima trappings [40], [41]. Since NN offers lesser processing time than the optimization algorithms. The higher processing time of the optimization algorithms can be curtailed by reducing the number of generations and the population size. Nonetheless, the steps may lead to improper exploration of search-space.

Micro-genetic algorithm also uses small population size. Nevertheless, the performance of genetic algorithm deteriorates if the parameters to be optimized are correlated [39]. Moreover, swarm-intelligence-based algorithms perform better than evolution-based optimization algorithms [15]-[17]. Bees Algorithm (BA), BFOA and ABC, elements of swarm-intelligence-based optimization algorithms, have the ability to induct completely new and random solutions into the swarm [15]. Therefore, these algorithms have the capability to properly explore search-space even with the smaller population size. BFA and BA have many control variables and there is no any systemic way to decided upon control variables values. Moreover, ABC performs better than BFA and BA [15].

B. Automatic Voltage Regulator

Power system is the most complex, vulnerable, highly interconnected and widely distributed system in today's world [41]. The restructuring of electric industry, ever increasing need of the energy, and the recent blackouts have increased the necessity to install the optimally tuned power systems controllers [42], [43]. AVR is used to keep the terminal voltage constant [44]. Deviation of the terminal voltage from rated value deteriorates the life and efficiency of all associated equipments [44], [45]. Moreover AVR enhances power system stability [42], [46]. Thus, the controller plays a major role in power system operation during the steady-state and the transient periods.

Many PID controller tuning methods can be found in the literature, since PID is the most commonly used controller in the industries [47]. Ziegler-Nichols (ZN) is the most commonly used classical method for PID controller tuning [47], [48]. However, it is often difficult to get optimal or near optimal PID gains using ZN method [49]. On the other hand, bio-inspired optimization algorithms are the well proven algorithms for producing the optimal solutions of numerous engineering problems [48]. PID controller is used to regulate gain of AVR [46]. Hence the performance enhancement of AVR is basically the optimization of PID controller gains.

Fig. 26 shows the model of power system used in this analysis. The same model has been used by [47]-[49]. Fig. 26 illustrates that PID-controller generates the control-signal on the basis of error-signal magnitude (ΔV_e). The error-signal is the deviation of generator output voltage from the reference voltage. The proportional controller minimizes magnitude of the error signal. The integral component of the controller reduces the steady-state error whereas the derivative element of the controller damps out oscillations.

Table 5 gives the transfer functions and gain-values of the power system components. The values are well within limits suggested in [47]-[49].

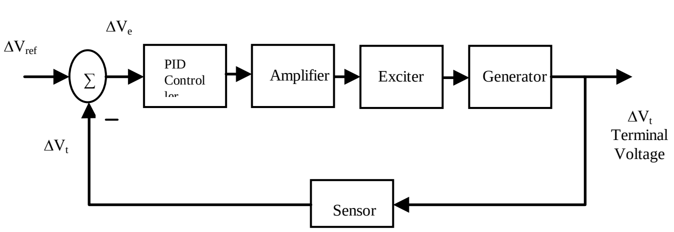

Fig. 26 Block diagram of power system model

Table 5 Transfer functions of power-system-model components and their values

Components	Transfer Function	Parameter Values
Amplifier	TF $_{Amplifier}$ = $K_a/1+\tau_a s$	$K_a = 12$ $\tau_a = 0.1$
Exciter	TF $_{Exciter}$ = $K_e/1+\tau_e s$	$K_e = 1$ $\tau_e = 0.4$
Generator	TF $_{Generator}$ = $K_g/1+\tau_g s$	$K_g = 1$ $\tau_g = 1$
Sensor	TF $_{Sensor}$ = $K_s/1+\tau_s s$	$K_s = 1$ $\tau_s = 0.01$
Controller	TF $_{Controller}$ = $(k_p s+k_i+k_d s^2)/s$	

C. Performance assessment and variable setting

The fitness function used in this research work to optimize PID-AVR is given in (12). The fitness function has been taken from [49].

$$f(PID)=\frac{e^{-\beta}*\dfrac{t_s}{\max(t)}}{1-e^{-\beta}*\left|1-\dfrac{t_r}{\max(t)}\right|}+e^{-\beta}*O_{sh}+E_{ss} \qquad (12)$$

where O_{sh} is an overshoot, E_{SS} represents a steady-state error, t_S is the settling time, t_r shows the rise-time, β=1.5 and *max(t)*=1sec.

As BABC2 has produced inferior response than BABC1 on all benchmark functions, the algorithm has been omitted from further comparison. IABC has been run on 0.15, 0.25 and 0.35 values of *P*. GABC has been run on 0.5, 1.0 and 1.5 values of *C*. The colony size of all the algorithms has been fixed to 08. *limit* (i.e. the control variable) has been set to 08. The number of generations has been limited to 30. The initialization range has been fixed from 0.1 to 1.5. Each algorithm has been run 30 times to observe the variability of the algorithms. Performance analysis of the algorithms has been carried out on basis of maximum, minimum, average over thirty runs and standard deviation among thirty outputs.

D. Results and Discussion

GABC has performed best on *C* value equals to 0.5 and IABC on *P* value equals to 0.25. Table 6 gives output results of all the considered optimization algorithms, tested on (12) fitness function for optimizing PID-AVR. The results show that EABC, the proposed algorithm, has produced least maximum, minimum, average and standard-deviation values among all considered algorithms.

Fig. 27 clearly shows superiority of EABC. EABC has yielded the best convergence rate in comparison to all the compared algorithms. Keeping in view the robust performance and considerably higher convergence rate of the proposed algorithm, this research work can be taken as first step towards online tuning of PID-AVR parameters using the optimization algorithms.

Table 6 Output results of the compared algorithms for PID-AVR optimization

Algorithm	Maximum	Minimum	Average	Standard Deviation
ABC	24.5678	16.7150	20.1924	2.3981
GABC-0.5	8.1913	0.2899	1.2499	1.6372
IABC-0.25	15.7360	0.4058	3.4096	3.3025
BABC1	29.3804	0.2951	5.5139	7.7941
EABC	**4.9467**	**0.2786**	**0.6935**	**0.8775**

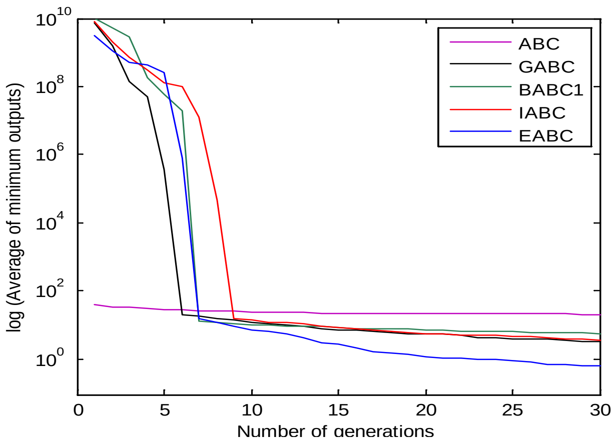

Fig. 27 Convergence rates of the algorithms on PID-AVR optimization objective function

VIII. Conclusion

In this research work, the performance of ABC, gbest-guided ABC (GABC), Improved-ABC (IABC), best/ABC/1 (BABC1), best/ABC/2 (BABC2) and proposed derivative of ABC named Enhanced-ABC (EABC) have been compared. The analysis has been carried out on eighteen commonly used high dimensional complex benchmark functions. Besides, performance of the algorithms has also been assessed on a real world application i.e. PID-AVR optimization. Performance assessment suggests decisive-impact of C-value and P-value on convergence of GABC and IABC respectively. Furthermore, there is no systematic way to decide upon the variable value. GABC has performed better on unimodal functions than multimodal functions. IABC has yielded better performance on multimodal functions than unimodal functions. BABC1 has performed better than BABC2 on all functions. BABC1 and BABC2 are highly prone to local optima traps, specifically, when the global optimum lies in very deep and curving valley. BABC1 and BABC2 have performed comparably better on the multi-dimensional functions than the unimodal functions. The analysis portrays that EABC has outperformed GABC and IABC on all functions. EABC has performed better than BABC1 on almost all functions. EABC has performed equally better on unimodal and multimodal functions. It reflects that the proposed algorithm, EABC, has the best capability to avert local optima than the other compared algorithms. Moreover, the proposed algorithm does not require any extra control variable unlike other variants of ABC algorithm.

Appendix

Appendix 1

Table 2 Performance results of GABC algorithms at four different values of C.

Functions	Algorithms	Max	Min	Avg	Std Dev
Griewank	GABC 0.5	9.99E-16	4.44E-16	6.11E-16	1.30E-16
	GABC 1.0	6.66E-16	4.44E-16	5.55E-16	5.05E-17
	GABC 1.5	**9.99E-16**	**4.44E-16**	**5.22E-16**	**1.29E-16**
	GABC 2.0	9.99E-16	4.44E-16	6.00E-16	1.29E-16
Rosenbrock	**GABC 0.5**	**3.1612**	**3.11E-02**	**5.63E-01**	**6.85E-01**
	GABC 1.0	4.4526	1.14E-02	7.43E-01	1.15E+00
	GABC 1.5	6.4158	8.44E-02	1.59E+00	1.76E+00
	GABC 2.0	20.0640	2.02E-01	2.23E+00	3.72E+00
Ackley	GABC 0.5	1.38E-13	5.60E-14	1.01E-13	1.92E-14
	GABC 1.0	4.17E-14	3.11E-14	3.75E-14	3.54E-15
	GABC 1.5	**4.17E-14**	**2.75E-14**	**3.50E-14**	**4.31E-15**
	GABC 2.0	4.53E-14	3.11E-14	3.87E-14	3.70E-15
Schwefel	GABC 0.5	2.86E-02	3.82E-04	1.32E-03	5.15E-03
	GABC 1.0	4.72E-04	3.82E-04	3.85E-04	1.65E-05
	GABC 1.5	**3.86E-04**	**3.82E-04**	**3.82E-04**	**6.92E-07**
	GABC 2.0	118.4387	3.82E-04	3.95E+00	2.16E+01
Himmelblau	GABC 0.5	-78.3323	-78.3323	-78.3323	**00**
	GABC 1.0	**-78.3323**	**-78.3323**	**-78.3323**	**00**
	GABC 1.5	-78.3323	-78.3323	-78.3323	**00**
	GABC 2.0	-78.3323	-78.3323	-78.3323	**00**
Step	GABC 0.5	8.71E-16	3.97E-16	6.46E-16	1.22E-16
	GABC 1.0	**7.35E-16**	**3.93E-16**	**5.80E-16**	**8.97E-17**
	GABC 1.5	7.29E-16	4.70E-16	5.92E-16	8.35E-17
	GABC 2.0	7.52E-16	4.37E-16	6.09E-16	1.08E-16
Bohachevsky	GABC 0.5	8.33E-16	2.22E-16	4.74E-16	1.25E-16
	GABC 1.0	5.55E-16	1.11E-16	4.22E-16	1.22E-16
	GABC 1.5	**5.55E-16**	**5.55E-17**	**3.74E-16**	**1.31E-16**
	GABC 2.0	5.55E-16	2.22E-16	4.46E-16	8.93E-17
Schwefel2	GABC 0.5	1.92E-14	6.09E-15	1.12E-14	3.28E-15
	GABC 1.0	1.80E-15	1.19E-15	1.51E-15	1.33E-16
	GABC 1.5	**1.65E-15**	**1.13E-15**	**1.42E-15**	**1.41E-16**
	GABC 2.0	2.26E-15	1.44E-15	1.84E-15	1.50E-16
Zakharov1	GABC 0.5	1.74E-12	3.16E-14	5.80E-13	3.82E-13
	GABC 1.0	**4.68E-14**	**2.26E-15**	**1.41E-14**	**1.12E-14**
	GABC 1.5	1.07E-13	5.83E-15	4.30E-14	3.00E-14
	GABC 2.0	3.26E-11	1.56E-12	9.47E-12	7.16E-12
Schwefel's Ridges	**GABC 0.5**	**384.9901**	**60.7910**	**255.1922**	**70.4696**
	GABC 1.0	472.0192	117.6052	282.9421	86.6242
	GABC 1.5	517.0625	133.9984	320.6369	99.9001
	GABC 2.0	472.4032	151.5902	359.0110	85.6502
Noisy Schwefel's Ridges	**GABC 0.5**	**80.2247**	**42.9922**	**60.9401**	**8.9862**
	GABC 1.0	75.0952	33.7006	62.7242	9.2184
	GABC 1.5	90.1739	55.5391	70.8466	8.6593
	GABC 2.0	90.8686	51.7526	73.9504	10.3576
Shifted Rotated Elliptic	GABC 0.5	-450	-450	-450	0
	GABC 1.0	-450	-450	-450	0
	GABC 1.5	**-450**	**-450**	**-450**	**0**
	GABC 2.0	-450	-450	-450	0
Random Shifted Sphere	GABC 0.5	7.69E-16	4.02E-16	6.14E-16	1.15E-16
	GABC 1.0	7.35E-16	4.00E-16	5.95E-16	9.77E-17
	GABC 1.5	**7.45E-16**	**4.33E-16**	**5.65E-16**	**9.16E-17**
	GABC 2.0	7.49E-16	4.02E-16	5.83E-16	9.10E-17
Random Shiftted Griewank	GABC 0.5	5.55E-16	4.44E-16	4.74E-16	4.99E-17
	GABC 1.0	5.55E-16	3.33E-16	4.63E-16	6.57E-17
	GABC 1.5	**5.55E-16**	**2.22E-16**	**4.51E-16**	**7.68E-17**
	GABC 2.0	5.55E-16	3.33E-16	4.63E-16	6.57E-17
Schafer	GABC 0.5	2.75E-01	0.00E+00	1.89E-02	6.07E-02
	GABC 1.0	**1.07E-02**	**3.16E-15**	**1.00E-03**	**2.17E-03**
	GABC 1.5	4.84E-01	1.67E-16	3.75E-02	1.22E-01
	GABC 2.0	4.67E-01	1.57E-13	4.24E-02	1.15E-01
Colvile	GABC 0.5	2.1277	6.38E-02	8.52E-01	5.75E-01
	GABC 1.0	**1.5924**	**2.64E-02**	**4.53E-01**	**3.92E-01**
	GABC 1.5	1.4699	2.25E-02	5.64E-01	3.62E-01
	GABC 2.0	1.5640	6.47E-02	5.28E-01	4.44E-01
Zakharov	GABC 0.5	2.97E-02	7.72E-09	1.37E-03	5.40E-03
	GABC 1.0	**4.07E-03**	**6.58E-07**	**6.52E-04**	**1.15E-03**

	GABC 1.5	1.12E-02	2.10E-05	1.16E-03	2.26E-03
	GABC 2.0	9.97E-02	2.58E-05	1.14E-02	2.03E-02
Quartic	GABC 0.5	1.18E-02	6.92E-04	4.35E-03	2.69E-03
	GABC 1.0	6.59E-03	8.06E-04	3.39E-03	1.79E-03
	GABC 1.5	**7.23E-03**	**4.89E-04**	**2.77E-03**	**1.60E-03**
	GABC 2.0	9.59E-03	8.19E-04	4.33E-03	2.48E-03

Appendix 2

Table3 Performance results of IABC algorithms at three different values of Probability (*P*).

Functions	Algorithm	Max	Min	Avg	Std Dev
Griewank	**IABC 0.15**	**8.88E-16**	**4.44E-16**	**5.77E-16**	**1.14E-16**
	IABC 0.25	9.99E-16	4.44E-16	5.77E-16	1.18E-16
	IABC 0.35	8.88E-16	4.44E-16	5.96E-16	1.07E-16
Rosenbrock	**IABC 0.15**	**7.4408**	**0.5550**	**2.5837**	**1.7811**
	IABC 0.25	7.0467	0.3711	2.6007	1.8262
	IABC 0.35	10.3054	0.4061	2.6268	2.0762
Ackley	IABC 0.15	4.17E-14	2.75E-14	3.50E-14	4.00E-15
	IABC 0.25	**3.82E-14**	**2.75E-14**	**3.39E-14**	**4.00E-15**
	IABC 0.35	3.82E-14	2.75E-14	3.51E-14	3.58E-15
Schwefel	IABC 0.15	3.82E-04	3.82E-04	3.82E-04	8.18E-13
	IABC 0.25	3.82E-04	3.82E-04	3.82E-04	3.12E-10
	IABC 0.35	**3.82E-04**	**3.82E-04**	**3.82E-04**	**7.82E-13**
Himmelblau	IABC 0.15	-78.3323	-78.3323	-78.3323	0.0000
	IABC 0.25	-78.3323	-78.3323	-78.3323	0.0000
	IABC 0.35	**-78.3323**	**-78.3323**	**-78.3323**	**0.0000**
Step	IABC 0.15	7.50E-16	3.28E-16	5.55E-16	1.01E-16
	IABC 0.25	**7.26E-16**	**2.89E-16**	**5.52E-16**	**1.21E-16**
	IABC 0.35	7.38E-16	4.19E-16	6.11E-16	8.35E-17
Bohachevsky	**IABC 0.15**	**5.55E-16**	**2.22E-16**	**4.02E-16**	**1.17E-16**
	IABC 0.25	5.55E-16	2.78E-16	4.35E-16	9.90E-17
	IABC 0.35	5.55E-16	1.11E-16	4.11E-16	1.29E-16
Schwefel2	IABC 0.15	1.65E-15	1.08E-15	1.42E-15	1.42E-16
	IABC 0.25	**1.82E-15**	**1.17E-15**	**1.41E-15**	**1.55E-16**
	IABC 0.35	1.65E-15	1.18E-15	1.46E-15	1.20E-16
Zakharov1	IABC 0.15	2.58E-10	3.46E-13	4.07E-11	5.66E-11
	IABC 0.25	3.12E-11	5.42E-13	8.07E-12	8.32E-12
	IABC 0.35	**2.67E-11**	**5.55E-14**	**4.17E-12**	**5.91E-12**
Schwefel's Ridges	**IABC 0.15**	**807.9164**	**500.1765**	**669.7189**	**73.9467**
	IABC 0.25	885.8818	494.6965	704.0401	96.5522
	IABC 0.35	859.9889	421.4624	670.6579	100.7596
Noisy Schwefel's Ridges	IABC 0.15	87.7583	58.8921	76.5929	7.4268
	IABC 0.25	**93.4944**	**57.1121**	**74.7421**	**8.4245**
	IABC 0.35	91.9252	58.8699	75.5614	9.4278
Shifted Rotated Elliptic	IABC 0.15	-450	-450	-450	0
	IABC 0.25	**-450**	**-450**	**-450**	**0**
	IABC 0.35	-450	-450	-450	0
Random Shifted Sphere	**IABC 0.15**	**9.12E-16**	**4.76E-16**	**5.89E-16**	**9.45E-17**
	IABC 0.25	7.40E-16	4.73E-16	5.98E-16	8.58E-17
	IABC 0.35	7.46E-16	4.12E-16	5.98E-16	1.04E-16
Random Shiftted Griewank	**IABC 0.15**	**5.55E-16**	**3.33E-16**	**4.51E-16**	**5.78E-17**
	IABC 0.25	5.55E-16	3.33E-16	4.85E-16	6.83E-17
	IABC 0.35	5.55E-16	2.22E-16	4.55E-16	6.74E-17
Schafer	IABC 0.15	0.4041	0.0025	0.1105	0.1092
	IABC 0.25	**0.3521**	**0.0026**	**0.1070**	**0.1113**
	IABC 0.35	0.4843	0.0028	0.1100	0.1378
Colvile	IABC 0.15	9.7536	0.6855	2.7620	1.9363
	IABC 0.25	9.4860	0.1740	2.5284	2.3077
	IABC 0.35	**4.2606**	**0.1855**	**1.8564**	**1.0155**
Zakharov	IABC 0.15	2.1533	0.0296	0.6175	0.6139
	IABC 0.25	1.7082	0.0258	0.3933	0.4239
	IABC 0.35	**1.4740**	**0.0107**	**0.3195**	**0.3749**
Quartic	IABC 0.15	9.02E-03	1.04E-03	4.43E-03	2.23E-03
	IABC 0.25	8.69E-03	1.61E-03	3.87E-03	1.82E-03
	IABC 0.35	**8.91E-03**	**6.89E-04**	**3.70E-03**	**2.09E-03**

Appendix 3

Table 4 Performance results of the compared optimization algorithms on all considered benchmark functions.

Functions	Algorithms	Max	Min	Avg	Std Dev	Test
Griewank	ABC	9.99E-16	5.55E-16	7.44E-16	1.70E-16	1
	BABC2	8.88E-16	4.44E-16	5.59E-16	7.98E-17	1
	BABC1	**3.33E-16**	**1.11E-16**	**2.26E-16**	**6.83E-17**	**0**
	GABC-1.5	9.99E-16	4.44E-16	5.22E-16	1.29E-16	1
	IABC-0.15	8.88E-16	4.44E-16	5.77E-16	1.14E-16	1
	EABC	**3.33E-16**	**1.11E-16**	**2.21E-16**	**5.93E-17**	**-**
Rosenbrock	ABC	2.1034	0.0779	0.5480	0.4930	1
	BABC2	19.3961	0.1756	5.0233	5.2076	1
	BABC1	14.6555	0.1419	3.8966	3.3735	1
	GABC-0.5	3.1612	0.0311	0.5627	0.6848	1
	IABC-0.15	7.4408	0.5550	2.5837	1.7811	1
	EABC	**0.9331**	**0.0069**	**0.2622**	**0.2499**	**-**
Ackley	ABC	2.33E-09	3.93E-10	9.78E-10	4.19E-10	1
	BABC2	4.17E-14	2.75E-14	3.48E-14	3.90E-15	1
	BABC1	**3.11E-14**	**2.04E-14**	**2.39E-14**	**3.55E-15**	**0**
	GABC-1.5	4.17E-14	2.75E-14	3.50E-14	4.31E-15	1
	IABC-0.25	3.82E-14	2.75E-14	3.39E-14	4.00E-15	1
	EABC	**3.05E-14**	**1.96E-14**	**2.30E-14**	**3.09E-15**	**-**
Schwefel	ABC	118.4388	3.82E-04	18.8873	43.1578	1
	BABC2	3.82E-04	3.82E-04	3.82E-04	4.78E-13	0
	BABC1	3.82E-04	3.82E-04	3.82E-04	8.92E-13	0
	GABC-1.5	3.86E-04	3.82E-04	3.82E-04	6.92E-07	0
	IABC-0.35	3.82E-04	3.82E-04	3.82E-04	7.82E-13	0
	EABC	3.82E-04	3.82E-04	3.82E-04	1.94E-13	0
Himmelblau	ABC	-78.3323	-78.3323	-78.3323	0.00	0
	BABC2	-78.3323	-78.3323	-78.3323	0.00	0
	BABC1	-78.3323	-78.3323	-78.3323	0.00	0
	GABC-1.0	-78.3323	-78.3323	-78.3323	0.00	0
	IABC-0.35	-78.3323	-78.3323	-78.3323	0.00	0
	EABC	-78.3323	-78.3323	-78.3323	0.00	0
Step	ABC	9.66E-16	4.77E-16	6.79E-16	1.48E-16	1
	BABC2	9.42E-16	3.24E-16	6.86E-16	1.09E-16	1
	BABC1	**3.12E-16**	**1.44E-16**	**2.52E-16**	**4.43E-17**	**0**
	GABC-1.0	7.35E-16	3.93E-16	5.80E-16	8.97E-17	1
	IABC-0.25	7.26E-16	2.89E-16	5.52E-16	1.21E-16	1
	EABC	**2.95E-16**	**1.40E-16**	**2.41E-16**	**8.62E-17**	**0**
Bohachevsky	ABC	6.77E-15	7.77E-16	1.45E-15	1.05E-15	1
	BABC2	7.22E-16	1.11E-16	4.79E-16	1.22E-16	1
	BABC1	**2.78E-16**	**0.00E+00**	**5.74E-17**	**6.60E-17**	**0**
	GABC-1.5	5.55E-16	5.55E-17	3.74E-16	1.31E-16	1
	IABC-0.15	5.55E-16	2.22E-16	4.02E-16	1.17E-16	1
	EABC	**1.94E-16**	**0.00E+00**	**5.61E-17**	**1.19E-17**	**-**
Schwefel2	ABC	5.99E-10	1.63E-10	3.61E-10	1.10E-10	1
	BABC2	1.87E-15	1.35E-15	1.52E-15	1.32E-15	1
	BABC1	**1.64E-15**	**6.48E-16**	**1.16E-15**	**2.51E-16**	**0**
	GABC-1.5	1.65E-15	1.13E-15	1.42E-15	1.41E-15	1
	IABC-0.25	1.82E-15	1.17E-15	1.41E-15	1.55E-16	1
	EABC	**1.85E-15**	**9.90E-16**	**1.17E-15**	**2.00E-16**	**-**
Zakharov1	ABC	1.08E-08	3.16E-10	2.95E-09	2.21E-09	1
	BABC2	2.69E-02	3.97E-03	1.19E-02	5.94E-03	1
	BABC1	2.16E-03	1.43E-04	7.23E-04	5.94E-04	1
	GABC-1.0	4.68E-14	2.26E-15	1.41E-14	1.12E-14	1
	IABC-0.35	2.67E-11	5.55E-14	4.17E-12	5.91E-12	1
	EABC	**1.22E-15**	**6.53E-16**	**1.01E-15**	**1.78E-16**	**-**
Schwefel's Ridges	ABC	4.38E+02	1.15E+02	3.05E+02	6.97E+01	1
	BABC2	9.76E+02	5.06E+02	7.39E+02	1.14E+02	1
	BABC1	8.39E+02	4.53E+02	6.26E+02	9.26E+01	1
	GABC-0.5	**3.85E+02**	**6.08E+01**	**2.55E+02**	**7.05E+01**	**1**
	IABC-0.15	8.08E+02	5.00E+02	6.70E+02	7.39E+01	1
	EABC	**3.03E+02**	**8.21E+01**	**2.06E+02**	**5.22E+01**	**-**
Noisy Schwefel's Ridges	ABC	97.3855	50.9731	77.5277	11.1642	1
	BABC2	99.7449	66.8569	80.6434	8.8235	1
	BABC1	90.5750	59.8665	74.8630	8.2968	1
	GABC-0.5	80.2247	42.9922	60.9401	8.9862	1

	IABC-0.25	93.4944	57.1121	74.7421	8.4245	1
	EABC	**78.2952**	**40.4345**	**57.2374**	**8.8122**	**-**
Shifted Rotated Elliptic	ABC	-450	-450	-450	0.00	0
	BABC2	-450	-450	-450	0.00	0
	BABC1	-450	-450	-450	0.00	0
	GABC-1.5	-450	-450	-450	0.00	0
	IABC-0.35	-450	-450	-450	0.00	0
	EABC	-450	-450	-450	0.00	-
Random Shifted Sphere	ABC	9.18E-16	4.57E-16	7.02E-16	1.22E-16	1
	BABC2	7.53E-16	4.45E-16	6.53E-16	9.69E-17	1
	BABC1	**4.23E-16**	**1.57E-16**	**2.69E-16**	**5.25E-17**	**0**
	GABC-1.5	7.45E-16	4.33E-16	5.65E-16	9.16E-17	1
	IABC-0.35	7.46E-16	4.12E-16	5.98E-16	1.04E-16	1
	EABC	**3.93E-16**	**1.38E-16**	**2.65E-16**	**4.83E-17**	**0**
Random Shiftted Griewank	ABC	6.66E-16	3.33E-16	5.33E-16	7.38E-17	1
	BABC2	5.55E-16	3.33E-16	4.18E-16	6.31E-17	1
	BABC1	**3.33E-16**	**1.11E-16**	**1.85E-16**	**7.34E-17**	**0**
	GABC-1.5	5.55E-16	2.22E-16	4.51E-16	7.68E-17	1
	IABC-0.35	5.55E-16	2.22E-16	4.55E-16	6.74E-17	1
	EABC	**3.33E-16**	**1.11E-16**	**1.82E-16**	**7.30E-17**	**-**
Schafer	ABC	2.86E-02	3.56E-05	5.15E-03	6.43E-03	1
	BABC2	4.84E-01	3.44E-04	8.22E-02	1.29E-01	1
	BABC1	4.84E-01	0.00E+00	5.91E-02	1.30E-01	1
	GABC-1.0	1.07E-02	3.16E-15	1.00E-03	2.17E-03	1
	IABC-0.25	3.52E-01	2.65E-03	1.07E-01	1.11E-01	1
	EABC	**4.05E-03**	**0**	**1.41E-04**	**7.39E-04**	**-**
Colvile	ABC	7.00E+00	1.88E-01	2.03E+00	1.83E+00	1
	BABC2	4.29E+00	2.52E-01	1.02E+00	8.46E-01	1
	BABC1	**1.24E+00**	**5.64E-03**	**3.05E-01**	**2.64E-01**	**0**
	GABC-1.0	1.59E+00	2.64E-02	4.53E-01	3.92E-01	1
	IABC-0.35	4.26E+00	1.85E-01	1.86E+00	1.02E+00	1
	EABC	**1.25E+00**	**4.43E-03**	**2.66E-01**	**2.94E-01**	**-**
Zakharov	ABC	2.42E-02	1.70E-05	4.80E-03	6.16E-03	1
	BABC2	1.66E+00	8.96E-02	6.15E-01	4.29E-01	1
	BABC1	4.41E-01	5.50E-04	6.22E-02	8.64E-02	1
	GABC-1.0	4.07E-03	6.58E-07	6.52E-04	1.15E-03	1
	IABC-0.35	1.47E+00	1.07E-02	3.20E-01	3.75E-01	1
	EABC	**2.18E-05**	**4.28E-15**	**1.20E-06**	**4.67E-06**	**-**
Quartic	ABC	1.58E-02	1.45E-03	6.48E-03	3.34E-03	1
	BABC2	7.47E-03	4.03E-04	3.48E-03	2.27E-03	1
	BABC1	**3.64E-03**	**2.90E-04**	**1.63E-03**	**1.05E-03**	**0**
	GABC-1.5	7.23E-03	4.89E-04	2.77E-03	1.60E-03	1
	IABC-0.35	8.91E-03	6.89E-04	3.70E-03	2.09E-03	1
	EABC	**3.42E-03**	**3.01E-04**	**1.64E-03**	**1.07E-03**	**-**

ACKNOWLEDGMENT

The authors greatly acknowledge Ministry of Higher Education (MOHE) FRGS Grants No: 203/PELECT/6071247 and Universiti Sains Malaysia for the financial and facilities support.

REFERENCES

[1] F. Kang, J. Li, and Z. Ma, "Rosenbrock artificial bee colony algorithm for accurate global optimization of numerical functions," *Information Sciences*, vol. 181, no. 16, pp. 3508-3531, 2011.

[2] W. Yingxu, W. Kinsner, and Z. Du, "Contemporary cybernetics and its facets of cognitive informatics and computational intelligence," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 39, no. 4, pp. 823-833, 2009.

[3] M. Heydar, and A. Yousefli, "An Optimization model for market basket analysis with allocation considerations: a genetic algorithm based approach," *International Journal of Mathematics and Computers in Simulation*, vol. 11, pp. 159-163, 2017.

[4] A. Osareh, B. Shadgar, and R. Markham, "A computational-intelligence-based approach for detection of exudates in diabetic retinopathy images," *IEEE Transactions on Information Technology in Biomedicine*, vol. 13, no. 4, pp. 535-545, 2009.

[5] J. Sevcik, T. Urbanek, and J. Sipek, "Novel approach to utilization of differential evolution for camera placement problem," *International Journal of Mathematics and Computers in Simulation*, vol. 9, pp. 136-140, 2015.

[6] S. X. Wu, and W. Banzhaf, "The use of computational intelligence in intrusion detection systems: A review," *Applied Soft Computing*, vol. 10, no. 1, pp. 1-35, 2010.

[7] A. Brabazon, M. O'Neill, and I. Dempsey, "An introduction to evolutionary computation in finance," *IEEE Computational Intelligence Magazine*, vol. 3, no. 4, pp. 42-55, 2008.

[8] N. J. van Eck, L. Waltman, J. den Berg, and U. Kaymak, "Visualizing the computational intelligence field," *IEEE Computational Intelligence Magazine*, vol. 1, no. 4, pp. 6-10, 2006.

[9] I. Juniku, and P. Marango, "PID design with bio-inspired intelligent algorithms for high order systems," *International Journal of Mathematics and Computers in Simulation*, vol. 9, pp. 44-52, 2015.

[10]L. N. d. Castro, "Fundamentals of natural computing; basic concepts, algorithms and applications," in *Computer and Information Science Series*, New York: Taylor & Francis, 2006.

[11] B. Akay, and D. Karaboga, "A modified artificial bee colony algorithm for real-parameter optimization," *Information Sciences*, vol. 192, no. 6, pp. 120-142, 2012.

[12] E. Bonabeau, M. Dorigo, and G. Theraulaz, *Swarm Intelligence: From Natural to Artificial Systems*. New-York: Oxford University Press, 1999.

[13] L. N. d. Castro, "Fundamentals of natural computing: an overview," *Physics of Life Reviews*, vol. 4, no. 1, pp. 1-36, 2007.

[14] D. Karaboga, and B. Basturk, "A powerful and efficient algorithm for numerical function optimization: artificial bee colony algorithm," *Journal of Global Optimization*, vol. 39, no. 3, pp. 459-471, 2007.

[15] E.-A. Mohammed, "Performance assessment of foraging algorithms vs. evolutionary algorithms," *Information Sciences*, vol. 182, no. 1, pp. 243-263, 2012.

[16] D. Karaboga, and B. Akay, "On the performance of artificial bee colony algorithm," *Applied Soft Computing*, vol. 8, no. 1, pp. 687-697, 2008.

[17] 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.

[18] D. Karaboga, and C. Ozturk, "A novel clustering approach: Artificial Bee Colony (ABC) algorithm," *Applied Soft Computing*, vol. 11, no. 1, pp. 652-657, 2011.

[19] K. Nurhan, "A new design method based on artificial bee colony algorithm for digital IIR filters," *Journal of the Franklin Institute*, vol. 346, no. 4, pp. 328-348, 2009.

[20] S. L. Sabat, S. K. Udgata, and A. Abraham, "Artificial bee colony algorithm for small signal model parameter extraction of MESFET," *Engineering Applications of Artificial Intelligence*, vol. 23, no. 5, pp. 689-694, 2010.

[21] S. Samanta, and S. Chakraborty, "Parametric optimization of some non-traditional machining processes using artificial bee colony algorithm," *Engineering Applications of Artificial Intelligence*, vol. 24, no. 6, pp. 946-957, 2011.

[22] A. Ş. Şahin, B. Kılıç, and U. Kılıç, "Design and economic optimization of shell and tube heat exchangers using artificial bee colony (ABC) algorithm," *Energy Conversion and Management*, vol. 52, no. 11, pp. 3356-3362, 2011.

[23] W. -F. Gao, S. -Y. Liu, and F. Jiang, "An improved artificial bee colony algorithm for directing orbits of chaotic systems," *Applied Mathematics and Computation*, vol. 218, no. 7, pp. 3868-3879, 2011.

[24] W. Gao, and S. Liu, "Improved artificial bee colony algorithm for global optimization," *Information Processing Letters*, vol. 111, no. 17, pp. 871-882, 2011.

[25] W. Gao, S. Liu, and L. Huang, "A global best artificial bee colony algorithm for global optimization," *Journal of Computational and Applied Mathematics*, vol. 236, no. 11, pp. 2741-2753, 2012.

[26] Y. -F. Liu, and S. -Y. Liu, "A hybrid discrete artificial bee colony algorithm for permutation flowshop scheduling problem," *Applied Soft Computing*, vol. 13, no. 3, pp. 1459-1463, 2013.

[27] M. H. Kashan, N. Nahavandi, and A.H. Kashan, "DisABC: A new artificial bee colony algorithm for binary optimization," *Applied Soft Computing*, vol. 12, no. 1, pp. 342-352, 2012.

[28] B. Alatas, "Chaotic bee colony algorithm for global numerical optimization," *Expert systems with applications*, vol. 37, no. 8, pp. 5682-5687, 2010.

[29] W. -F. Gao, and S. -Y. Liu, "A modified artificial bee colony

algorithm," *Computers & Operations Research*, vol. 39, no. 3, pp. 687-697, 2012.
[30] L. Elliott, *et al*., "Reaction mechanism reduction and optimisation for modelling aviation fuel oxidation using standard and hybrid genetic algorithms," *Computers & Chemical Engineering*, vol. 30, no. 5, pp. 889-900, 2006.
[31] G. Zhu, and S. Kwong, "Gbest-guided artificial bee colony algorithm for numerical function optimization," *Applied Mathematics and Computation*, vol. 217, no. 7, pp. 3166-3173, 2010.
[32] G. Li, P. Niu, and X. Xiao, "Development and investigation of efficient artificial bee colony algorithm for numerical function optimization," *Applied Soft Computing*, vol. 12, no. 1, pp. 320-332, 2012.
[33] 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.
[34] K. Sathish Kumar, and T. Jayabarathi, "Power system reconfiguration and loss minimization for an distribution systems using bacterial foraging optimization algorithm," *International Journal of Electrical Power & Energy Systems*, vol. 36, no. 1, pp. 13-17, 2012.
[35] E. S. Ali, and S. M. Abd-Elazim, "Coordinated design of PSSs and TCSC via bacterial swarm optimization algorithm in a multimachine power system," *International Journal of Electrical Power & Energy Systems*, vol. 36, no. 1, pp. 84-92, 2012.
[36] G. Nicosia, S. Rinaudo, and E. Sciacca, "An evolutionary algorithm-based approach to robust analog circuit design using constrained multi-objective optimization," *Knowledge-Based Systems*, vol. 21, no. 3, pp. 175-183, 2008.
[37] M. Wan, C. Wong, L. Li, and Y. Yang, "Chaotic ant swarm approach for data clustering," *Applied Soft Computing*, vol 12, no. 8, pp. 2387-2393, 2012.
[38] T. K. Das, G. K. Venayagamoorthy, and U. O. Aliyu, "Bio-inspired algorithms for the design of multiple optimal power system stabilizers: SPPSO and BFA," *IEEE Transactions on Industry Applications*, vol. 44, no. 5, pp. 1445-1457, 2008.
[39] M. Ramirez-Gonzalez, and O. P. Malik, "Power system stabilizer design using an online adaptive neurofuzzy controller with adaptive input link weights," *IEEE Transactions on Energy Conversion*, vol. 23, no. 3, pp. 914-922, 2008.
[40] G. K. Venayagamoorthy, and R. G. Harley, "A continually online trained neurocontroller for excitation and turbine control of a turbogenerator," *IEEE Transactions on Energy Conversion*, vol. 16, no. 3, pp. 261-269, 2001.
[41] P. Kundur, *et al*., "Definition and classification of power system stability IEEE/CIGRE joint task force on stability terms and definitions," *IEEE Transactions on Power Systems*, vol. 19, no. 3, pp. 1387-1401, 2004.
[42] V. Vittal, "Consequence and impact of electric utility industry restructuring on transient stability and small-signal stability analysis," *Proceedings of the IEEE*, vol. 88, no. 2, pp. 196-207, 2000.
[43] H. Gozde, and M. C. Taplamacioglu, "Comparative performance analysis of artificial bee colony algorithm for automatic voltage regulator (AVR) system," *Journal of the Franklin Institute*, vol. 348, no. 8, pp. 1927-1946, 2011.
[44] A. K. Goswami, C.P. Gupta, and G.K. Singh, "Minimization of voltage sag induced financial losses in distribution systems using FACTS devices," *Electric Power Systems Research*, vol. 81, no. 3, pp. 767-774, 2011.
[45] M. J. Basler, and R. C. Schaefer, "Understanding power-system stability," *IEEE Transactions on Industry Applications*, vol. 44, no. 2, pp. 463-474, 2008.
[45] G. Zwe-Lee, "A particle swarm optimization approach for optimum design of PID controller in AVR system," *IEEE Transactions on Energy Conversion*, vol. 19, no. 2, pp. 384-391, 2004.
[47] D. Devaraj, and B. Selvabala, "Real-coded genetic algorithm and fuzzy logic approach for real-time tuning of proportional-integral - derivative controller in automatic voltage regulator system," *IET Generation, Transmission & Distribution*, vol. 3, no. 7, pp. 641-649, 2009.
[48] K. Dong Hwa, "Hybrid GA–BF based intelligent PID controller tuning for AVR system," *Applied Soft Computing*, vol. 11, no. 1, pp. 11-22, 2011.
[49] D. Manjarres, *et al*., "A novel heuristic approach for distance and connectivity-based multihop node localization in wireless sensor networks", *Soft Computing*, vol. 17, no. 1, pp. 17-28, 2013.

Abdul G. Abro received Ph.D. degree in 2013 from Universiti Sains Malaysia. He is currently working as Assistant Professor in the College of Applied Engineering, Al Muzahimiyah Campus, King Saud University. His research interests include bio-inspired computation technique and their applications in power system engineering.

Junita Mohamad-Saleh received her B.Sc (in Computer Engineering) degree from the Case Western Reserve University, USA in 1994, the M.Sc. degree from the University of Sheffield, UK in 1996 and the Ph.D. degree from the University of Leeds, UK in 2002. She is currently an Associate Professor in the School of Electrical and Electronic Engineerinf, Universiti Sains Malaysia. Her research interests include computational intelligence, tomograpic & medical imaging and soft computing. .

Noorazliza Sulaiman received her Ph.D. in 2017 from Universiti Sains Malaysia. She is currently working as Post-doctoral fellow in the School of Electrical & Electronic Engineering, Universiti Sains Malaysia. His research interests include computational intelligence and its applications.