Turkish Journal of Electrical Engineering & Computer Sciences

http://journals.tubitak.gov.tr/elektrik/

Research Article

Turk J Elec Eng & Comp Sci
(2014) 22: 620 – 636

doi:10.3906/elk-1209-23

Multiple-global-best guided artificial bee colony algorithm for induction motor parameter estimation

Abdul Ghani ABRO, Junita MOHAMAD-SALEH*
School of Electrical and Electronics Engineering, Engineering Campus, Universiti Sains Malaysia, Nibong Tebal, Seberang Perai Selatan, Penang, Malaysia

Received: 06.09.2012 • **Accepted:** 10.01.2013 • **Published Online:** 21.03.2014 • **Printed:** 18.04.2014

Abstract: An induction motor is the most commonly used motor in industry today. Motor circuit parameters are essential for designing, evaluating performance, and controlling the applications of the motor. However, it is difficult to measure the electric parameters, e.g., resistance and reactance, of induction motors accurately. Therefore, researchers have noted the parameter estimation of induction motors as an essential optimization problem. The artificial bee colony (ABC) algorithm is an efficient element of bioinspired optimization algorithms and has been successfully applied in numerous engineering applications. However, the ABC algorithm suffers from slow convergence and poor exploitation. Additionally, there are bleak chances of getting a fitter food source for scout bees using the the standard ABC algorithm scheme. Therefore, different solutions have already been proposed to avoid the flaws of the ABC algorithm. Nevertheless, the proposed solutions are either computationally intensive or prone to local optima traps or they require additional control variables to tune. Moreover, there is no systematic way to tune the additional control variables for yielding the optimal performance of the algorithms. Therefore, this research work proposes a novel variant of the ABC algorithm, which capitalizes on multiple global-best food sources rather than a single global-best food source. In addition, this research work proposes a novel scheme for enhancing the performance of the ABC algorithm's scout bee. Two modifications for the performance enhancement of the ABC algorithm are proposed in this research work. The proposed algorithm is compared with various recently proposed variants of the ABC algorithm and various other available methods for estimating induction motor parameters. The performance of the proposed algorithm is also assessed using the chaotic map initialization technique. The results prove that the proposed algorithm is able to achieve the best convergence among all of the compared algorithms.

Key words: Induction motor, parameter estimation, variant artificial bee colony, evolutionary algorithm, metaheuristic algorithms, computational intelligence

1. Introduction

Electrical energy is a multipurpose energy carrier and, hence, is primarily associated with society and economic development [1,2]. Electric motors consume almost 70% of the total electrical energy consumed by the industrial sector [3,4]. Therefore, the energy usage assessment of motors is an essential issue in industry for implementing energy-saving strategies. Induction motors are the most widely used motors in industries because of their lower maintenance requirement, lower space requirement, and easy controlling [1,5–7]. Moreover, most of the auxiliaries of thermal power plants are driven by induction motors. To ensure the successful restart of the plants, it is important to investigate the feasibility of the auxiliary motor start-ups [6,8]. Therefore, the exact

*Correspondence: jms@eng.usm.my

information of the motor circuit parameters is essential for the design, performance evaluation, and control applications of the motor.

Lindenmeyer et al. classified the methods proposed for induction motor parameter estimation into various classes and also gave the merits and demerits of the classes [8]. The classification is presented below.

1. The first-class methods calculate the parameters from the motor construction data. These methods require a huge dataset comprising the motor geometry and the material parameters. This is the most accurate method. However, it is the most costly method and the required data are not usually available in industry.

2. The second-class methods estimate the parameters from the steady-state motor models. These methods use iterative solution techniques for estimating the parameters. These are the most commonly used methods, as the required data are easily available in industry.

3. The third-class methods estimate the parameters based on the stand-still frequency response. As the stand still tests are not common in industry, the methods are not often used. Furthermore, these methods are expensive and require more time.

4. The fourth-class methods apply time-domain measurements for the parameter estimation. All of the parameters cannot be measured; therefore, the motor models are simplified and, as a result, the parameters are measured by sacrificing accuracy. Moreover, the method is costly and the required data are not commonly available.

This classification reveals that the most commonly used and most economical methods for estimating the motor parameters are the iterative methods. Bioinspired optimization algorithms also belong to the family of iterative techniques. These optimization algorithms resulted in far superior performances as compared to the conventional linear and nonlinear iterative methods [9]. Moreover, the authors in [10] rigorously compared the bioinspired optimization algorithms with a few conventional iterative methods for induction motor parameter estimation. Their work finally concluded that bioinspired optimization algorithms are far superior to conventional iterative methods. Hence, researchers have employed the optimization algorithms to determine the induction motors parameters [6,11] because the algorithms have an inherent capability to find either the optimal solution or very near optimal solutions of the problem at hand [2,12].

The artificial bee colony (ABC) is a member of the swarm intelligence-based bioinspired optimization algorithm family [13]. The ABC algorithm imitates the foraging phenomenon of honeybees for evolving optimal solutions of problems [14]. The primary goal of honeybees during foraging is to optimize the time spent on energy foraging and the foraged energy. The bee swarm distributes tasks among different honeybee colony members to accomplish a task efficiently [13–15]. The ABC algorithm performs better on a number of benchmark functions than many other optimization algorithms, such as particle swarm optimization (PSO) or differential evolution (DE) [14–16]. Moreover, the ABC algorithm was already successfully applied in numerous engineering applications [2,17–21].

Nevertheless, the ABC algorithm converges slowly and has a tendency towards local optima traps [22–25]. To overcome these pitfalls, researchers have proposed different solutions. To the best of our knowledge, the variants in [22–30] are among the most rigorously tested variants of the ABC algorithm. The ABC variant proposed in [26] was computationally intensive. The variants proposed in [22,23,27,28] were outperformed by that in [25] on various benchmark functions. Research results reveal that the variant proposed in [29] yielded an inferior response on a number of benchmark functions as compared to the standard ABC algorithm. The variant proposed in [25] had more exploration capability than exploitation and hence converged slowly [9]. Additionally, the variants proposed in [24,30] generate solutions only around the global best possible solution (GBPS), and therefore they were prone to local optima trappings [9]. Overall, the existing variants are either computationally intensive or could not avoid the flaws of the ABC optimization algorithm. The pitfalls of the existing variants guide us towards the proposed algorithm. The proposed variant is compared with the various existing variants of the ABC algorithm on 2 test cases for estimating induction motor parameters. The performance of the proposed algorithm is also assessed using the chaotic map initialization technique.

This paper is organized into 7 sections. The problem of induction motor parameter estimation is formulated in Section 2. Section 3 discusses the standard ABC algorithm and the proposed algorithm is presented in Section 4. Section 5 presents the experimental setup. The results are discussed in Section 6 and the performance of the proposed optimization algorithm with various other methods is compared. Finally, the conclusion is presented in Section 7.

2. Problem formulation

An induction motor is a highly nonlinear system and the electric variables of the rotor are not measurable [3,6,11]. Additionally, the skin effect and magnetic saturation complicate the modeling process of the machine even more. Therefore, the electromagnetic parameters of an induction motor are difficult to measure. Due to this, the parameters are commonly measured by indirect methods. The inputs are the voltage, speed, starting torque, full-load torque, and maximum torque, whereas the measured parameters are the rotor and stator resistance, reactance, and magnetizing reactance. The parameters can be determined using the approximate or exact model of an induction motor, discussed below.

2.1. Problem formulation based on approximate model of induction motor

The approximate model of the induction motor neglects the magnetizing reactance and rotor reactance; hence, it sacrifices accuracy a little. The objective function is given in Eq. (1) and was used in [11]:

$$\min(F) = f_1^2 + f_2^2 + f_3^2, \tag{1}$$

where

$$f_1 = \frac{K_t R_2}{s\left[(R_1 + \frac{R_2}{s})^2 + X_1^2\right]} - T_{FL}(mf),$$

$$f_2 = \frac{K_t R_2}{(R_1 + R_2)^2 + X_1^2} - T_{STR}(mf),$$

$$f_3 = \frac{K_t}{2\left[R_1 + \sqrt{R_1^2 + X_1^2}\right]} - T_{MAX}(mf),$$

$$K_t = \frac{3V_{ph}^2}{\omega_s}.$$

T_{FL} represents the full-load torque; T_{STR} is the starting torque; T_{MAX} is the maximum torque; R is the resistance; subscripts 1 and 2 symbolize the stator and rotor, respectively; X corresponds to the reactance; s

stands for the motor slip; mf is the manufacturer's data; ω_s is the synchronous speed; and V_{ph} symbolizes the phase voltage.

Minimization of the objective function is subjected to a few conditions, i.e. the values of the calculated parameters must lie within certain ranges. Moreover, the deviation between the estimated and manufacturer's values of the torque types must also be within an appropriately small range. Therefore, the objective function is multiobjective.

2.2. Problem formulation based on exact model of induction motor

The exact model of an induction motor has comparatively higher accuracy. The model considers the magnetizing and rotor reactances, along with the parameters conceived in the approximate model. The objective function is given in Eq. (2) and was used in [11]:

$$\min(F) = f_1^2 + f_2^2 + f_3^2 + f_4^2, \tag{2}$$

where

$$f_1 = \frac{K_t R_2}{s\left[(R_{th} + \frac{R_2}{s})^2 + X^2\right]} - T_{FL}(mf),$$

$$f_2 = \frac{K_t R_2}{(R_{th} + R_2)^2 + X^2} - T_{STR}(mf),$$

$$f_3 = \frac{K_t}{2\left[R_{th} + \sqrt{R_{th}^2 + X^2}\right]} - T_{MAX},$$

$$f_4 = \cos\left(\tan^{-1}\left(\frac{X}{R_{th} + \frac{R_2}{s}}\right)\right) - PF(mf),$$

$$V_{th} = \frac{V_{ph} X_m}{X_1 + X_m}, R_{th} = \frac{R_1 X_m}{X_1 + X_m}, X_{th} = \frac{X_1 X_m}{X_1 + X_m},$$

and

$$X = X_2 + X_{th}, K_t = \frac{3V_{th}^2}{\omega_s}.$$

T_{FL} represents the full-load torque; T_{STR} is the starting torque; T_{MAX} stands for the maximum torque; PF stands for the power factor; s stands for the motor slip; R is the resistance; X corresponds to the reactance; subscripts *1*, *2*, and m symbolize the stator, rotor, and magnetizing, respectively; mf stands for the manufacturer's data; ω_s is the synchronous speed; and V_{th} symbolizes Thévenin voltage.

Minimization of the exact model objective function has to meet a few conditions, i.e. the values of the calculated parameters must lie within certain ranges. Moreover, the deviation between the estimated and manufacturer's values of the torque and power factor must also be within a certain range.

3. ABC optimization algorithm

Bees in the ABC algorithm are divided into 3 different classes, i.e. employed bees (EBs), onlooker bees (OBs), and scout bees (SBs) [14]. The standard ABC optimization algorithm carries an equal number of EBs and OBs, and generally only 1 SB is used [15]. The EB searches for food sources around the hive. Each food source represents a possible solution of the problem and the hive symbolizes the search space. Once food sources are assigned to each EB, then the bees explore the neighborhood of the assigned food source using the following mutation equation [14,15]:

$$z_{ij} = y_{ij} + \phi_{ij}(y_{ij} - y_{kj}), \tag{3}$$

where y_{ij} symbolizes the jth dimension of the ith food source, y_{kj} represents the jth dimension of the kth food source, z_{ij} corresponds to the candidate solution of the jth dimension of the$i$th food source, i and k are mutually exclusive food sources, $j \in$ [1,2,.... D], D is the dimension of search space, j and k are randomly chosen numbers, and φ is a random number within [–1, 1].

The fitness of the modified food source, also called the candidate food source, is calculated using the following equation [13,22,29]:

$$fit_i = \begin{cases} \frac{1}{1+f_i}, f_i \geq 0, \\ 1 + abs(f_i), f_i < 0, \end{cases}, \tag{4}$$

where f_i represents the objective function value of the ith food source and fit_i is the corresponding fitness value after the transformation.

Afterwards, the fitness values of the candidate possible solution and the old possible solution are compared. The food source that has the higher fitness value is retained [25,28]. This is called greedy selection. The OBs wait in the dancing area of the hive for receiving information about the explored food sources from the EBs [15,23]. The OBs probably select food sources that have a higher nectar amount [14,15]. The probability of a food source having a higher fitness value is calculated using the following equation [13,24,25]:

$$p_i = \frac{fit_i}{\sum\limits_{j=i}^{NS} fit_j}, \tag{5}$$

where NS stands for the maximum number of food sources and i is the selected food source.

If any food source is explored for the maximum number of times without success, then it is abandoned [25]. Next, the SB flies around the hive to replace the abandoned food source with a completely new food source. A control variable 'limit' controls the maximum number of food source explorations [14,15]. The SB flies around the hive to pick a random food source. Hence, the SB induces a search-space exploring capability in the ABC algorithm by replacing a well-tried and unsuccessful food source with a completely new food source. The SB uses the following equation to pick a new food source [26]:

$$y_{ij} = y_j^{\min} + rand(0,1)(y_j^{\max} - y_j^{\min}), \tag{6}$$

where y_{min} is the lower limit of the search space, y_{max} represents the upper limit of the search space, *rand* is a random number within [0, 1], and j is the jth index of the food source.

The pseudocode of the standard ABC algorithm is given below.

1. START
2. Initialization of the control variables
3. Random initialization of the food sources
4. CYCLE = 1

5. REPEAT while a preset number of generations is reached
6. Explore the neighborhood of the food sources by EBs using Eq. (3)
7. Calculate the fitness of the explored food source using Eq. (4)
8. Apply greedy selection
9. Select food sources for OBs to explore using Eq. (5)
10. Explore the neighborhood of the selected food sources by OBs using Eq. (3)
11. Calculate the fitness of the explored food source using Eq. (4)
12. Apply greedy selection
13. Memorize the best food source found so far
14. IF an abandoned food source exists
15. Assign a food source to the SB using Eq. (6)
16. INCREASE cycle counter
17. TERMINATE if the cycle is equal to the maximum number of cycles
18. END

4. Proposed enhanced ABC algorithm

A literature review reveals that the induction of the GBPS's influence in the mutation equation [Eq. (3)] of the standard ABC algorithm increases the convergence rate of the algorithm. However, it has a tendency for premature convergence due to local minima traps. The local optima trapping tendency is tackled by introducing additional control variables, but there is no systematic way to decide upon the optimal values of the variables. This research work proposes an enhanced ABC (EABC) algorithm.

4.1. Proposal of a novel mutation equation

The proposed algorithm capitalizes on multiple GBPSs rather than a single GBPS. The reliance on the single GBPS accumulates the swarm at one location of the search space. Nevertheless, the application of the multiple GBPSs averts the accumulation of the swarm at any single location of the search space. Thus, the algorithm capitalizes on GBPSs for an accelerating convergence rate. Simultaneously, the algorithm has the ability to avert local optima traps, as it explores around the multiple best-found locations of the search space. The mutation equation of the proposed algorithm is given below:

$$z_{ij} = y_{ij} + \varphi_{ij}\,(y_{ij} - y_{best/second-best/\ldots/\,n,\,j}), \tag{7}$$

where y_{ij} symbolizes the jth dimension of the ith food source; $y_{best/second-best/\ldots/n,j}$ is the jth index of one of the food sources having a higher fitness value; z_{ij} corresponds to the candidate solution of the jth dimension of the ith food source; $j \in$ [1,2,.... D], D is the dimension of the search space; and φ is a random number within [–1, 1].

The EABC algorithm divides the whole population of possible solutions into different groups. It then assigns one of the selected GBPSs for enhancing the fitness of the possible solutions present in a selected group, regardless of the GBPS's presence in the group.

Now, the question is: The division of food sources should be into how many groups to be enough for obtaining the optimal performance of the proposed algorithm? The authors' experiences suggest that a higher number of groups would yield a better performance of the proposed algorithm. Each group capitalizes on one of the selected GBPSs; therefore, more search-space locations will be explored simultaneously using more groups. However, it is also important that the bee swarm must be divided in such a way that each group contains a good number of bees. Hence, the number of GBPSs is limited by the population size of the swarm, i.e. number of bees. Generally, the population size is limited either by the dimension of the problem or by the search space capacity.

4.2. Convergence enhancement of the proposed algorithm

DE is a simple yet effective evolutionary algorithm. One among the various variants of DE capitalizes on the following mutation equation;

$$DE/rand/1 : Z_i = Y_{r1} + F(Y_{r2} - Y_{r3}), \tag{8}$$

where Z_i corresponds to the candidate solution of a possible solution; i ğ [1,2,.... D], D is the dimension of the search space; Y_{r1} and Y_{r2} represent mutually different random integer indices selected from a population of possible solutions; and F is a positive real number of generally less than 1.00.

DE and the ABC algorithm generate solutions on the basis of the vector difference and, hence, the mutation equations of one algorithm are applicable to the other. However, Eq. (8) is good at exploration but poor at exploitation. Moreover, a properly balanced mutation equation in terms of exploration and exploitation is immensely important. Therefore, this research work proposes the replacement of the third term of Eq. (8) with the global-best food source. The modified form of the mutation equation is given below:

$$z_{ij} = y_{mj} + \varphi_{ij}\,(y_{ij} - y_{best\,j}), \tag{9}$$

where y_{mj} symbolizes the jth dimension of the mth food source; $y_{best,\,j}$ represents the jth dimension of the global-best food source; z_{ij} corresponds to the candidate solution of the jth dimension of the ith food source; m and i are mutually exclusive food sources; j ğ [1,2,.... D], D is the dimension of the search space; and φ is a random number within [–1, 1].

However, the application of Eq. (9) is subjected to 2 conditions: if Eq. (7) fails to produce a better solution than the existing solution and if the randomly generated number is less than Selection (S). Selection (S) is a user-defined control variable. A higher value of S will yield better convergence.

4.3. Enhancement of the SB stage

In the standard ABC algorithm and in its various variants, the SB is assigned a randomly initialized food source using Eq. (6). Hence, there are very bleak chances of getting a food source with a better fitness value. This research work proposes a novel scheme for assigning a food source to the SB. The scheme capitalizes on the GBPS for assigning a food source to the SB. The following is the equation to assign a food source to the SB:

$$z_{nj} = (y_{best,\,j})\beta_{nj}\,, \tag{10}$$

where z_{nj} is the jth dimension magnitude of the newly assigned food source, $x_{best;j}$ is the jth dimension magnitude of the global-best food source; and β_{nj} is a random number within [0.90 1.10].

The limits of β are chosen so that the dimension magnitudes of the new food source may not go out of the chosen boundaries of the function and the global-best food source may not be distorted considerably; otherwise, the global-best food source may lose quality. The pseudocode of the standard ABC algorithm is given below.

1. START
2. Initialization of the control variables
3. Random initialization of the food sources
4. Divide the food sources into a user-defined number of groups
5. CYCLE = 1
6. REPEAT while a preset number of generations is reached
7. Explore the neighborhood of the food sources by EBs using Eq. (7)
8. Calculate the fitness of the explored food source using Eq. (4)
9. IF (fitness of the explored food source > fitness of the existing food source)
10. Update the food source
11. ELSE explore the neighborhood of the food sources by EBs using Eq. (9)
12. Apply greedy selection
13. Select food sources for OBs to explore using Eq. (5)
14. Explore the neighborhood of the selected food sources by OBs using Eq. (7)
15. Calculate the fitness of the explored food source using Eq. (4)
16. IF (fitness of the explored food source > fitness of the existing food source)
17. Update the food source
18. ELSE explore the neighborhood of the food sources by OBs using Eq. (9)
19. Apply greedy selection
20. Memorize the best food source found so far
21. IF an abandoned food source exists
22. Assign a food source to the SB using Eq. (10)
23. INCREASE cycle counter
24. TERMINATE if the cycle is equal to the maximum number of cycles
25. END

5. Experimental setup

This research work considers the standard ABC, global-best-guided ABC (GABC) [22], improved ABC (IABC) [25], modified ABC (MABC) [30], best-so-far ABC (BSFABC) [27], and modified ABC (ModABC) [29] algorithms for evaluating the performance of the proposed EABC algorithm. The proposed algorithm is also compared with the PSO algorithm, genetic algorithm (GA), and a conventional nonlinear iterative method [8]. Moreover, the performance of the proposed algorithm is also assessed using the chaotic map initialization technique. The EABC algorithm initialized using a chaotic map is named the chaotic EABC (CH-EABC) algorithm. The performances of the algorithms are analyzed using 2 different motors. The specification data of the motors are given in Table 1.

Table 1. Data of the motors used to analyze the performances of the optimization algorithms.

Specifications	Motor 1	Motor 2
Capacity (HP)	5	40
Voltage (V)	3000	3000
Frequency (Hz)	50	50
No. of poles	4	4
Full-load slip	0.07	0.07
Starting torque (Nm)	15	260
Maximum torque (Nm)	42	370
Full-load torque (Nm)	25	190

The colony size is fixed at 30, the number of generations is limited to 50, and '*limit*' is set to 20. As suggested in [22], the GABC algorithm is run on 3 values of the *C-parameter*, i.e. 0.5, 1.5, and 2.5, to analyze the impact of C on the GABC algorithm's performance. As directed in [25], the IABC algorithm is run on 3 different values of *P-variable*, i.e. 0.15, 0.25, and 0.35, for the performance analysis. The proposed algorithm is divided into 5 groups and S is set to 0.50. In groups of less than 5, the proposed algorithm did not perform the best among all of the algorithms for all of the test cases. Each algorithm is run 30 times on each objective function to analyze the robustness and convergence of the optimization algorithms. Possible solutions for all of the algorithms are initialized using Gaussian-generated random numbers, except for the CH-EABC algorithm. A sinusoidal $[\chi_{n+1} = \sin(\pi\chi_n)]$ chaotic map is considered for the CH-EABC algorithm's initialization. The chaotic map was also used in .[28].

The parameter estimation of an induction motor is a multiobjective problem. In this work, the multiobjective problem has been converted to a single-objective problem using the penalty technique. After obtaining the estimated values of the torque, the difference between the estimated and the manufacturer's values is calculated. As mentioned in Section 2, there are 3 different types of torque, i.e. starting torque, full-load torque, and maximum torque. The final comparison of the compared algorithms is carried out on the basis of the sum of absolute difference between the estimated and the manufacturer's values. The performances of the algorithms are analyzed on the basis of the least error, largest error, average error, and standard deviation among 30 final values of error. The standard deviation among 30 outputs predicts variability, whereas an average value over 30 outputs prophesizes the convergence of an algorithm. The least value gives the best convergence, while the largest value represents the worst convergence.

6. Results and discussion

This section compares the performance of the proposed optimization algorithm with various other optimization algorithms and the methods applied for the parameter estimation of the induction motor.

6.1. Comparison with variants of ABC optimization algorithm

Tables 2 and 3 show the results obtained by running the GABC algorithm using 3 different values of C. The results clearly depict that there is no single optimal value of C for yielding the optimal performance of the GABC algorithm. Moreover, there is no systematic way to calculate the optimal value of C. Hence, only on the basis of trial and error is the optimal C value to be decided. The GABC algorithm producing the minimum least value of error is considered for the performance comparison with the rest of the algorithms.

Table 2. Performance comparison of the GABC algorithm on the different values of C for motor 1.

Model (motor 1)	Algorithms	Largest	Least	Average	St. deviation
Approximate model	GABC 0.5	12.2979	6.18599	8.5481	1.2193
	GABC 1.5	10.1000	7.5342	8.7315	0.8149
	GABC 2.5	10.4396	7.75949	8.5589	0.7928
Exact model	GABC 0.5	17.2429	6.1170	10.0793	3.3707
	GABC 1.5	21.9223	6.39667	10.0939	3.4739
	GABC 2.5	18.1222	5.9079	9.1555	2.5836

Table 3. Performance comparison of the GABC algorithm on the different C values for motor 2.

Model (motor 2)	Algorithms	Largest	Least	Average	St. deviation
Approximate model	GABC 0.5	32.6699	2.1249	10.8135	6.5791
	GABC 1.5	24.0137	0.8138	9.1223	6.1021
	GABC 2.5	35.3529	2.7354	15.1881	10.7547
Exact model	GABC 0.5	33.7232	3.6348	18.3668	9.0155
	GABC 1.5	33.1505	5.4103	18.8873	8.9211
	GABC 2.5	94.6508	2.0094	24.2350	17.4698

Tables 4 and 5 show the results obtained by running the IABC algorithm at 3 different values of P. The results given in Tables 4 and 5 reveal the strong dependency of the IABC algorithm on the P value. There is no method that can be used to decide the optimal value of P, except trial and error. The IABC algorithm yielding the minimum least value of error is taken for the performance comparison with the rest of the algorithms.

Table 4. Performance comparison of the IABC algorithm on the different values of P for motor 1.

Model (motor 1)	Algorithms	Largest	Least	Average	St. deviation
Approximate model	IABC 0.15	11.7979	6.2574	8.5526	1.2196
	IABC 0.25	10.0680	6.3810	8.3756	0.9182
	IABC 0.35	10.2156	6.1109	8.6426	0.9181
Exact model	IABC 0.15	19.6349	6.3063	11.9035	3.9119
	IABC 0.25	25.9837	6.0472	10.3644	4.0277
	IABC 0.35	22.1956	6.2015	11.0079	3.9192

Table 5. Performance comparison of the IABC algorithm on the different values of P for motor 2.

Model (motor 2)	Algorithms	Largest	Least	Average	St. deviation
Approximate model	IABC 0.15	35.4138	5.5487	18.9372	10.4941
	IABC 0.25	37.0796	4.9802	16.5791	8.3424
	IABC 0.35	33.4255	2.9909	14.4684	9.4563
Exact model	IABC 0.15	37.1660	3.0891	17.6610	10.2937
	IABC 0.25	38.2471	3.1542	20.5923	9.0574
	IABC 0.35	46.3544	3.5544	19.7316	11.0145

Tables 6 and 7 give the output comparison of all of the ABC algorithm variants. The BSFABC algorithm results in the worst convergence among all of the compared algorithms. This reveals the highly local nature of the BSFABC algorithm. The standard ABC algorithm produces better convergence only in comparison to the BSFABC algorithm. The proposed algorithm produces the least minimum value in all of the considered cases. This shows the superiority of the proposed algorithm in finding the optimal solution in comparison to all of the other algorithms. The proposed algorithm produces the least average value in comparison to the compared algorithms in all of the cases. This clearly shows that the proposed algorithm possesses the best convergence among the compared algorithms.

For motor 1, the results produced by the CH-EABC algorithm are better than those produced by the EABC algorithm. However, the results produced by the CH-EABC algorithm are worse than those produced by the EABC algorithm for motor 2. Hence, it can be concluded that the application of chaotic maps does not necessarily enhance the performance of an algorithm. The CH-EABC algorithm might produce a better response on any chaotic map other than a sinusoidal.

Table 6. Performance comparison of the algorithms for motor 1.

Model (motor 1)	Algorithms	Largest	Least	Average	St. deviation
Approximate model	ABC	14.3899	6.3508	8.9274	1.7412
	GABC 0.5	12.2979	6.1860	8.5481	1.2193
	IABC 0.35	10.2156	6.1109	8.6426	0.9181
	MABC	10.3548	6.1225	8.4452	1.0597
	ModABC	18.1016	6.2670	8.8612	2.1904
	BSFABC	25.7823	6.3810	10.7352	4.3427
	EABC	9.0120	5.9816	7.0756	0.5085
	CH-EABC	6.8327	5.9966	6.2963	0.1282
Exact model	ABC	16.8181	6.0348	9.9764	2.8355
	GABC 0.5	17.2429	6.1170	10.0793	3.3707
	IABC 0.25	25.9837	6.0472	10.3644	4.0277
	MABC	17.8565	6.1690	9.7704	2.6455
	ModABC	37.3012	7.0232	13.1735	5.9250
	BSFABC	66.8101	6.2758	13.8546	10.7790
	EABC	10.6793	6.0301	7.2270	1.3867
	CH-EABC	12.2055	5.9650	6.9977	1.4477

The induction motor parameters yielding the least error value on the motor 1 dataset are shown in Tables 8 and 9, where the estimated parameters and calculated percentage error of motor 1 are given using the approximate and exact models. The results replicate the aforementioned discussion.

Table 7. Performance comparison of the algorithms for motor 2.

Model (motor 2)	Algorithms	Largest	Least	Average	St. deviation
Approximate model	ABC	92.6851	5.3122	23.1755	17.7932
	GABC 1.5	24.0137	0.8138	9.1223	6.1021
	IABC 0.35	33.4255	2.9909	14.4684	9.4563
	MABC	31.9732	4.1975	15.1538	8.5779
	ModABC	34.2653	6.2620	17.6784	9.4771
	BSFABC	60.3106	13.3382	29.0371	11.4128
	EABC	19.2781	0.2780	6.8780	5.5091
	CH-EABC	22.4605	0.8172	11.0624	6.8240
Exact model	ABC	63.0924	7.2409	28.8224	13.1456
	GABC 2.5	94.6508	2.0094	24.2350	17.4698
	IABC 0.15	37.1660	3.0891	17.6610	10.2937
	MABC	33.6194	4.0267	17.9831	10.3962
	ModABC	55.3698	3.9132	25.2119	13.8454
	BSFABC	84.8549	13.3887	45.4431	18.4830
	EABC	15.7527	0.1557	7.8365	7.3006
	CH-EABC	14.5529	0.5293	6.4054	4.9628

Table 8. Estimated parameters and percentage error using the approximate model of motor 1.

Algorithms	T_{MAX} (Nm)	T_{STR} (Nm)	T_{FL} (Nm)	Percentage error			R1 (Ω)	R2 (Ω)	X1 (Ω)
Manufacturer	42	15	25	T_{MAX}	T_{STR}	T_{FL}			
ABC	39.819	15.074	25.166	5.192	-0.495	-0.663	0.001	7.072	35.971
GABC 0.5	39.745	15.078	25.075	5.370	-0.518	-0.299	0.060	7.178	36.596
IABC 0.35	39.720	15.059	25.072	5.428	-0.396	-0.286	0.001	7.101	36.060
MABC	39.735	15.040	25.116	5.393	-0.263	-0.466	0.001	7.085	36.047
ModABC	39.691	15.102	24.978	5.498	-0.682	0.088	0.001	7.134	36.087
BSFABC	39.789	15.165	25.005	5.265	-1.097	-0.019	0.001	7.129	35.998
EABC	39.634	15.046	24.990	5.634	-0.308	0.040	0.001	7.126	36.139
CH-EABC	39.682	15.034	25.063	5.519	-0.226	-0.252	0.001	7.102	36.095

Table 9. Estimated parameters and percentage error using the exact model of motor 1.

Algorithms	T_{MAX} (Nm)	T_{STR} (Nm)	T_{FL} (Nm)	Percentage error			R1	R2	X1	X2	Xm
Manufacturer	42	15	25	T_{MAX}	T_{STR}	T_{FL}	(Ω)	(Ω)	(Ω)	(Ω)	(Ω)
ABC	39.683	15.068	25.017	5.518	0.450	0.067	0.001	6.37	20.00	13.38	350.00
GABC 0.5	39.716	15.073	25.048	5.437	0.487	0.193	0.001	6.48	16.72	16.89	350.00
IABC 0.25	39.698	15.046	25.065	5.480	0.307	0.261	0.001	6.26	20.00	13.01	304.93
MABC	39.729	14.992	25.176	5.408	0.055	0.706	0.001	6.36	18.78	14.65	350.00
ModABC	39.700	14.917	25.248	5.477	0.555	0.992	0.001	6.35	18.27	15.22	350.00
BSFABC	39.795	15.056	25.163	5.251	0.374	0.651	0.001	6.36	18.95	14.41	349.07
EABC	39.706	15.013	25.120	5.461	0.089	0.480	0.001	6.34	18.35	14.93	322.54
CH-EABC	39.616	14.988	25.052	5.675	0.083	-0.207	0.001	5.93	19.80	12.09	209.05

The induction motor parameters yielding the least error values for the motor 2 dataset are shown in Tables 10 and 11, where the estimated parameters and calculated percentage error of motor 2 are given using the approximate and exact models. The results replicate aforementioned discussion.

Table 10. Estimated parameters and percentage error using the approximate model of motor 2.

Algorithms	T_{MAX} (Nm)	T_{STR} (Nm)	T_{FL} (Nm)	Percentage error			R1 (Ω)	R2 (Ω)	X1 (Ω)
Manufacturer	370	260	190	T_{MAX}	T_{STR}	T_{FL}			
ABC	360.576	260.656	194.774	2.547	-0.252	-2.513	1.693	0.759	1.526
GABC 1.5	369.302	260.765	189.371	0.189	-0.294	0.331	1.424	0.824	1.999
IABC 0.35	361.081	260.793	189.477	2.411	-0.305	0.275	1.606	0.799	1.731
MABC	374.994	262.557	193.542	-1.350	-0.983	-1.864	1.391	0.805	1.991
ModABC	365.1242	251.5160	193.1943	1.3178	3.263	-1.681	1.443	0.7982	2.017
BSFABC	343.0155	264.3033	181.6591	7.2931	-1.655	4.390	1.968	0.7984	1.000
EABC	369.850	259.483	189.927	0.041	0.199	0.039	1.403	0.823	2.033
CH-EABC	368.945	261.010	190.272	0.285	-0.389	-0.143	1.449	0.816	1.956

Table 11. Estimated parameters and percentage error using the exact model of motor 2.

Algorithms	T_{MAX} (Nm)	T_{STR} (Nm)	T_{FL} (Nm)	Percentage error			R1	R2	X1	X2	Xm
Manufacturer	370	260	190	T_{MAX}	T_{STR}	T_{FL}	(Ω)	(Ω)	(Ω)	(Ω)	(Ω)
ABC	353.752	261.725	185.846	4.391	-0.664	2.186	1.732	0.803	0.10	1.44	400
GABC 2.5	372.785	259.467	188.002	-0.753	0.205	1.052	1.311	0.846	0.10	2.06	400
IABC 0.15	368.416	263.277	192.661	0.428	-1.260	-1.401	1.526	0.792	0.10	1.70	315
MABC	367.383	262.135	194.746	0.707	-0.821	-2.498	1.563	0.776	0.17	1.56	400
ModABC	373.518	259.820	184.503	-0.951	0.069	2.893	1.241	0.867	0.88	1.37	200
BSFABC	358.528	266.194	174.979	3.101	-2.382	7.906	1.508	0.892	1.84	0.12	400
EABC	370.075	260.013	189.752	-0.020	-0.005	0.131	1.403	0.824	0.10	1.93	400
CH-EABC	369.601	260.091	190.734	0.108	-0.035	-0.386	1.429	0.815	0.10	1.89	400

6.2. Comparison with the PSO and GA algorithms

The performance of the proposed optimization algorithm is compared with those of the PSO and GA for the parameter estimation. PSO is a prominent element of the swarm intelligence-based optimization algorithms, whereas GA is an efficient element of the evolutionary optimization algorithms [11]. The results of the PSO and GA for the parameter estimation of the induction motor were taken from [11]. The data of the induction motor and parameter settings of the PSO and GA can also be seen in [11]. The PSO and GA are run for 200 generations, whereas the proposed EABC algorithm is run for 100 generations, for the same population size. As the authors in [11] only reported the best results among 30 runs, the comparison is also carried out only on the basis of the best results.

The results reported in Tables 12 and 13 state that the proposed optimization algorithm (EABC) results in the most optimal results among the compared optimization algorithms, even though the proposed algorithm is run for half (100) of the total number of generations (200) that the PSO and GA are run for. This proves that the proposed optimization algorithm is the most balanced optimization algorithm in terms of the exploration and the exploitation capabilities.

6.3. Comparison with the artificial immune algorithm

The performance of the proposed optimization algorithm is compared with that of the artificial immune (AI) optimization algorithm. The AI algorithm is one of various recently proposed efficient optimization algorithms. The results of the AI algorithm for the parameter estimation of an induction motor were taken from [6]. The data of the induction motor and the parameter settings of the AI algorithm can also be seen in [6]. The AI algorithm is run for 50 iterations. The proposed algorithm is also run for the same number of iterations. The authors in [6] only reported the best results among 20 runs. Therefore, the performance of the proposed algorithm is compared with the AI algorithm on the basis of the best results among 20 runs.

Table 12. Comparative results of the PSO, GA, and EABC optimization algorithms for motor 1.

Algorithms	Approximate model						Exact model					
	T_{MAX} (Nm)	T_{STR} (Nm)	T_{FL} (Nm)	Percentage error			T_{MAX} (Nm)	T_{STR} (Nm)	T_{FL} (Nm)	Percentage error		
Manufacturer	42	15	25	T_{MAX}	T_{STR}	T_{FL}	42	15	25	T_{MAX}	T_{STR}	T_{FL}
PSO	39.5	14.99	22.41	5.95	0.60	10.36	40.97	17.6	22.11	2.45	17.36	11.56
GA	37.8	17.88	21.04	10.06	19.2	15.86	35.98	16.73	20.09	14.33	11.53	19.64
EABC	39.66	15.03	25.03	5.58	0.22	0.13	39.75	15.04	25.14	5.35	0.26	0.56

Table 13. Comparative results of the PSO, GA, and EABC optimization algorithms on motor 2.

Algorithms	Approximate model						Exact model					
	T_{MAX} (Nm)	T_{STR} (Nm)	T_{FL} (Nm)	Percentage error			T_{MAX} (Nm)	T_{STR} (Nm)	T_{FL} (Nm)	Percentage error		
Manufacturer	*370*	*260*	*190*	T_{MAX}	T_{STR}	T_{FL}	370	260	190	T_{MAX}	T_{STR}	T_{FL}
PSO	321	205.8	175	13.22	20.85	7.77	381.63	255.55	222.78	3.14	1.7	17.25
GA	315	150	166.79	14.86	42.31	12.21	355.48	285.7	200.99	3.92	0.5	5.78
EABC	370.5	260.68	188.80	0.13	0.26	0.63	369.79	260.19	190.22	0.06	0.07	0.11

The results of the AI and EABC algorithms are tabulated in Tables 14 and 15 for motors 1 and motor 2, respectively. The reported results show that the proposed EABC algorithm outperforms the AI optimization algorithm for the parameter estimation of an induction motor. The proposed algorithm outperforms the AI algorithm on both motors' datasets.

Table 14. Comparative results of the PSO, GA, and EABC optimization algorithms for motor 1.

Algorithms	Approximate model						Exact model					
	T_{MAX} (Nm)	T_{STR} (Nm)	T_{FL} (Nm)	Percentage error			T_{MAX} (Nm)	T_{STR} (Nm)	T_{FL} (Nm)	Percentage error		
Manufacturer	42	15	25	T_{MAX}	T_{STR}	T_{FL}	42	15	25	T_{MAX}	T_{STR}	T_{FL}
AI	38.44	15.44	20.36	8.00	2.9	18	41.8	16.03	27.44	0.4	7.0	9.7
EABC	39.634	15.046	24.99	5.63	-0.31	0.04	39.71	15.01	25.1	5.46	0.089	0.48

Table 15. Comparative results of the PSO, GA, and EABC optimization algorithms on motor 2.

Algorithms	Approximate Model						Exact Model					
	T_{MAX} (Nm)	T_{STR} (Nm)	T_{FL} (Nm)	Percentage error			T_{MAX} (Nm)	T_{STR} (Nm)	T_{FL} (Nm)	Percentage error		
Manufacturer	370	260	190	T_{MAX}	T_{STR}	T_{FL}	370	260	190	T_{MAX}	T_{STR}	T_{FL}
AI	376	272.36	170.44	1.6	4.8	10.5	377.9	255.93	178.56	2	0.16	6
EABC	369.85	259.48	189.93	0.04	0.20	0.04	370.08	260.0	189.8	0.02	0.005	0.13

6.4. Comparison with the conventional method

As clearly mentioned in Section 1, the iterative methods are the most commonly used and the most economical methods among the various available induction motor parameter estimation methods. Therefore, in this research work, the performance of the proposed optimization algorithm is compared with an enhanced nonlinear iterative method for induction motor parameter estimation. The enhanced nonlinear iterative method (SOLNP) was proposed in [8]. The induction motor data can also be seen in [8], although the authors did not mention the number of iterations for which the SOLNP was run. However, the proposed EABC algorithm is run for 50 iterations.

Table 16 presents the best results of the proposed algorithm and the SOLNP algorithm for comparison, and the induction motor parameters in per-unit (pu) values are presented. Table 16 suggests that the proposed algorithm results in a lower percentage of error than the SOLNP. Hence, the results clearly establish the superiority of the proposed optimization algorithm for the parameter estimation of an induction motor.

Table 16. Comparative results of the SOLNP and the proposed optimization algorithms.

	Algorithm	R1	X1	R2	X2	Xm
Magnitude (pu)	SOLNP	0.02411	0.76779	0.27389	0.070127	3.2316
	EABC	0.02412	0.76739	0.26544	0.066411	3.1405
Percent error	SOLNP	0.0005	0.0006	0.0271	0.0415	0.0293
	EABC	0.0005	0.0001	0.0046	0.0136	0.0003

7. Conclusion

This research work has proposed an efficient variant of the ABC algorithm, the EABC. The proposed algorithm was extensively compared with the standard ABC algorithm, its variants, and other commonly used methods for estimating the parameters of an induction motor. The comparative analysis proved that the proposed algorithm has the best convergence (shown by the least average error) and also possesses the best capability to find the most optimal induction motor parameters (shown by the minimum least error) compared to the other algorithms, as it is able to capitalize on multiple so-far best-found locations of the search space. Furthermore, the comparative analysis revealed that the initialization of the food sources using chaotic maps may enhance the searching ability of the optimization algorithms. Nevertheless, the performance enhancement of the bioinspired optimization algorithms through the chaotic map initialization technique is subject to the chosen chaotic map.

Acknowledgment

The authors acknowledge the fellowship scheme of IPS-Universiti-Sains Malaysia (USM) and Fundamental Research Grant Scheme (FRGS) from the Malaysian Ministry of Higher Education No: 203/PELECT/6071247 for the financial support.

References

[1] A.A. Bazmi, G. Zahedi, “Sustainable energy systems: role of optimization modeling techniques in power generation and supply—a review”, Renewable and Sustainable Energy Reviews, Vol. 15, pp. 3480–3500, 2011.

[2] A.G. Abro, J. Mohamad-Saleh, “Control of power system stability - reviewed solutions based on intelligent systems”, International Journal of Innovative Computing, Information and Control, Vol. 10, pp. 6643–6666, 2012.

[3] V.P. Sakthivel, R. Bhuvaneswari, S. Subramanian, “An accurate and economical approach for induction motor field efficiency estimation using bacterial foraging algorithm”, Measurement, Vol. 44, pp. 674–684, 2011.

[4] H. Çalış, A. Çakır, E. Dandıl, “Artificial immunity based induction motor bearing fault diagnosis”, Turkish Journal of Electrical Engineering & Computer Sciences, Vol. 21, pp. 1–25, 2012.

[5] V. Prakash, S. Baskar, S. Sivakumar, S.K. Krishna, “A novel efficiency improvement measure in three-phase induction motors, its conservation potential and economic analysis”, Energy for Sustainable Development, Vol. 12, pp. 78–87, 2008.

[6] V.P. Sakthivel, R. Bhuvaneswari, S. Subramanian, "Artificial immune system for parameter estimation of induction motor", Expert Systems with Applications, Vol. 37, pp. 6109–6115, 2010.

[7] M. Akar, İ. Çankaya, "Broken rotor bar fault detection in inverter-fed squirrel cage induction motors using stator current analysis and fuzzy logic", Turkish Journal of Electrical Engineering & Computer Sciences, Vol. 20, pp. 1077–1089, 2012.

[8] D. Lindenmeyer, H.W. Dommel, A. Moshref, P. Kundur, "An induction motor parameter estimation method", International Journal of Electrical Power & Energy Systems, Vol. 23, pp. 251–262, 2001.

[9] L.N. de Castro, Fundamentals of Natural Computing: Basic Concepts, Algorithms and Applications, New York, Taylor & Francis, 2006.

[10] R.K. Ursem, P. Vadstrup, "Parameter identification of induction motors using stochastic optimization algorithms", Applied Soft Computing, Vol. 4, pp. 49–64, 2004.

[11] V.P. Sakthivel, R. Bhuvaneswari, S. Subramanian, "Multi-objective parameter estimation of induction motor using particle swarm optimization", Engineering Applications of Artificial Intelligence, Vol. 23, pp. 302–312, 2010.

[12] Z. Bingül, O. Karahan, "Fractional PID controllers tuned by evolutionary algorithms for robot trajectory control", Turkish Journal of Electrical Engineering & Computer Sciences, Vol. 20, pp. 1123–1136, 2012.

[13] D. Karaboga, B. Basturk, "A powerful and efficient algorithm for numerical function optimization: artificial bee colony algorithm", Journal of Global Optimization, Vol. 39, pp. 459–471, 2007

[14] D. Karaboga, B. Basturk, "On the performance of artificial bee colony (ABC) algorithm", Applied Soft Computing, Vol. 8, pp. 687–697, 2008.

[15] D. Karaboga, B. Akay, "A comparative study of artificial bee colony algorithm", Applied Mathematics and Computation, Vol. 214, pp. 108–132, 2009.

[16] E.A. Mohammed, "Performance assessment of foraging algorithms vs. evolutionary algorithms", Information Sciences, Vol. 182, pp. 243–263, 2012.

[17] S. Samanta, S. Chakraborty, "Parametric optimization of some non-traditional machining processes using artificial bee colony algorithm", Engineering Applications of Artificial Intelligence, Vol. 24, pp. 946–957, 2011.

[18] H. Gozde, M.C. Taplamacioglu, "Comparative performance analysis of artificial bee colony algorithm for automatic voltage regulator (AVR) system", Journal of the Franklin Institute, Vol. 348, pp. 1927–1946, 2011.

[19] H. Gozde, M.C. Taplamacioglu, I. Kocaarslan, "Comparative performance analysis of artificial bee colony algorithm in automatic generation control for interconnected reheat thermal power system", International Journal of Electrical Power & Energy Systems, Vol. 42, pp. 167–178, 2012.

[20] K. Chandrasekaran, S. Hemamalini, S.P. Simon, N.P. Padhy, "Thermal unit commitment using binary/real coded artificial bee colony algorithm", Electric Power Systems Research, Vol. 84, pp. 109–119, 2012.

[21] Ö. Erçin, R. Çoban, "Identification of linear dynamic systems using the artificial bee colony algorithm", Turkish Journal of Electrical Engineering & Computer Sciences, Vol. 20, pp. 1175–1188, 2012.

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

[23] F. Kang, J. Li, Z. Ma, "Rosenbrock artificial bee colony algorithm for accurate global optimization of numerical functions", Information Sciences, Vol. 181, pp. 3508–3531, 2011.

[24] W.F. Gao, S.Y. Liu, L. Huang, "A global best artificial bee colony algorithm for global optimization", Journal of Computational and Applied Mathematics, Vol. 236, pp. 2741–2753, 2012.

[25] W.F. Gao, S.Y. Liu, "Improved artificial bee colony algorithm for global optimization", Information Processing Letters, Vol. 111, pp. 871–882, 2011.

[26] G. Li, P. Niu, X. Xiao, "Development and investigation of efficient artificial bee colony algorithm for numerical function optimization", Applied Soft Computing, Vol. 12, pp. 320–332, 2012.

[27] A. Banharnsakun, T. Achalakul, B. Sirinaovakul, "The best-so-far selection in artificial bee colony algorithm", Applied Soft Computing, Vol. 11, pp. 2888–2901, 2011.

[28] B. Alatas, "Chaotic bee colony algorithms for global numerical optimization", Expert Systems with Applications, Vol. 37, pp. 5682–5687, 2010.

[29] B. Akay, D. Karaboga, "A modified artificial bee colony algorithm for real-parameter optimization", Information Sciences, Vol. 192, pp. 120–142, 2012.

[30] W.F. Gao, S.Y. Liu, "A modified artificial bee colony algorithm", Computers & Operations Research, Vol. 39, pp. 687–697, 2012.