Landmark operator inspired artificial bee colony algorithm for optimal vector control of induction motor

Fani Bhushan Sharma* and
Shashi Raj Kapoor

Department of Electrical Engineering,
Rajasthan Technical University,
Kota, India
Email: fbsharma.kota@gmail.com
Email: srkapoor@rtu.ac.in
*Corresponding author

Abstract: In recent years, soft computing strategies have played vital role to solve optimisation problems associated with real world. In this paper, an efficient soft computing strategy namely, artificial bee colony algorithm (ABC_{algo}) is modified with incorporating landmark operator. The proposed modified ABC algorithm is named as landmark inspired ABC (LMABC). The performance of LMABC is evaluated on benchmark functions. Further, the proposed LMABC is applied for vector control of induction motor (IM) and subsequently to improve its efficiency. The vector control of IM includes control of magnitude and phase of each phase current and voltage. In this research paper the field orientated control, a digital implementation which demonstrates the capability of performing direct torque control, of handling system limitations and of achieving higher power conversion efficiency is considered. The obtained outcomes are significantly better than other state-of-art algorithms available in literature.

Keywords: swarm intelligence; landmark; induction motor; metaheuristics; real world optimisation.

Reference to this paper should be made as follows: Sharma, F.B. and Kapoor, S.R. (2021) 'Landmark operator inspired artificial bee colony algorithm for optimal vector control of induction motor', *Int. J. Swarm Intelligence*, Vol. 6, No. 1, pp.1–23.

Biographical notes: Fani Bhushan Sharma received his BE in Electrical Engineering from Government Engineering College Kota, India and MTech in Power System from Rajasthan Technical University, Kota, India. He is a Research Scholar at the Department of Electrical Engineering, University Teaching Department, Rajasthan Technical University, Kota, India. His research areas include soft computing techniques for analysis and control of induction motors.

Shashi Raj Kapoor is a Professor at the Department of Electrical Engineering, University Teaching Department, Rajasthan Technical University, Kota, India. He has done received his BE in Electrical Engineering from Government Engineering College Kota, MTech from Indian Institute of Technology, Roorkee, India. He received his PhD from Rajiv Gandhi Praudyogiki Vishvidhyalay, Madhya Pradesh, India. He is the author of various international journal papers of repute. His research areas include estimation and control of induction motors.

1 Introduction

The induction motor (IM) is amongst significant creation in recent history. It directed the wheels of advancement at a fast speed and significantly contributed in launching off the second industrial revolution by remarkably enhancing energy producing ability and promising long distance distribution of electricity feasible. The IMs are broadly applicable in the field of electrical engineering from quite a long time. There are numerous domestic, industrial, and commercial utilities of IMs available practically. An IM is a kind of brushless electric motor in which an alternating supply (AC) sustained to the windings of the stator generates a magnetic field which incites a current in the rotor windings. For the wider applicability of single motor for different requirements, speed control of IMs is done to achieve an extent of operating speed. The machine speed is very firm concerning to load changes. The total speed alteration is only in the extent n_s to $(1 - s) * n_s$, n_s (IM speed) subjected to supply frequency and number of poles. The speed control of IMs is achieved by changing applied voltage, changing rotor resistance, mechanical coupling of shaft of two motors (cascade control), pole changing schemes, and stator frequency control. The speed control of IMs is done for constant voltage to frequency ratio (v/f) for normal frequency range. In low frequency operations for (v/f) control cogging of IM shafts take place which makes it typical to control IM. To avoid this cogging for low frequency operations vector control of IMs is applied. In the vector control the voltage and frequency can be separately manipulated which produces optimum (v/f) for maximum torque even for quite low frequency operations. The vector control provides stability for load and set point changes, short rise times for set point changes, short settling times for load alterations, acceleration as well as braking are attainable with the highest set table torque, motor safety due to changeable torque limitation in motor and regenerative mode, drive and braking torque controlled free of the speed, highest breakaway torque attainable, torque control is required in a higher control extent, acknowledges a designated and/or volatile torque for lower speeds. The vector control is split into torque/current and speed control.

The first implementations of vector control technology were essentially lab curiosities since the computing power did not exist in an embedded form to make this technique viable. But as fortune would have it, another technology was being developed that would change the face of motor control for ever the microprocessors. It was not long before several manufacturers began offering microprocessor-based vector control solutions over a wide range of power options. Soon the term vector control became synonymous with high performance, high tech motors. Later, a less ambiguous and more technically precise term was coined for it as field oriented control (FOC).

FOC is a high-performance technique for motor control that is becoming increasingly attractive for all kinds of applications. But FOC requires a more sophisticated shaft sensor, such as a resolver or encoder. The high cost of this sensor is one of the main reasons why more and more designs are migrating to sensor less control. If you can eliminate the shaft sensor, sensor less trapezoidal commutation is still less expensive than sensor less FOC because the processor requirements are lower. But the total system cost is almost identical. This is because in a sensor less system, most of the cost is in your power devices and bus capacitor(s), which is determined by the motor horsepower.

Considering the enhanced performance and flexibility possible with FOC, it may very well turn out to be the most cost effective solution for your application. This is why many low cost, high volume applications (such as appliances) are abandoning trapezoidal control altogether and embracing FOC faster than ever before. Various researchers have worked on the FOC as available in the literature. The control of IM was carried out for standard ac motor using microprocessors in 1980 (Gabriel et al., 1980). A methodology for enhancing of FOC IM was proposed in 1993 (Liu et al., 1993). Further, digital FOC for dual 3 phase IM was proposed in 2003 (Bojoi et al., 2003). In 2004, backstepping wavelet neural network (NN) control for indirect FOC of IM was presented (Wai and Chang, 2004). Fuzzy self tuning speed control of an indirect FOC IM was presented in 2008 (Masiala et al., 2008). In 2007, GA-PSO-based vector control of indirect three-phase IM was presented (Kim, 2007). FOC has been implemented on Stator-flux-oriented vector control for brushless doubly fed induction generator (Shao et al., 2009). Various techniques for energy efficient control of three-phase IM were discussed in 2009 (Raj et al., 2009). Genetic algorithms (GA)-based fuzzy speed controllers for indirect FOC of IM was proposed (Douiri et al., 2012). Artificial bee colony (ABC_{algo}) algorithm-based design of optimal online self tuning PID controller was proposed in 2014 (Ebrahim, 2014).

The outcomes obtained from soft computing techniques are motivating and in search of more accurate and efficient results in this paper a landmark operator (Duan and Qiao, 2014) inspired ABC namely, LMABC is proposed for the optimal solution of vector control (FOC) of IMs.

The paper is structured in following manner: vector control of IMs is presented in Section 2. Section 3 presents ABC_{algo} and its propound variant LMABC. The analysis of outcomes is discussed in Section 4. The work is wrapped up in Section 6.

2 Vector control of induction motors

Vector control, generally known as FOC, is a variable frequency drive (VFD) control strategy where the stator currents of an IM are determined as two orthogonal elements that can be anticipated with a vector. The first constituent delineates the magnetic flux of the motor, and the next one delineates the torque. The control system of the IM computes the correlated current component references from the flux and torque references given by the IMs speed control. Proportional-integral (PI) controllers generally applicable to maintain the obtained current elements at their standard values. The pulse width modulation (PWM) of the IMs delineates the transistor switching as per the stator voltage standards which are the outcomes of the PI current controllers. The block diagram of FOC is presented in Figure 1. The implementation of FOC block diagram is presented in Figure 2.

Figure 1 Block diagram of field oriented vector control (FOC)

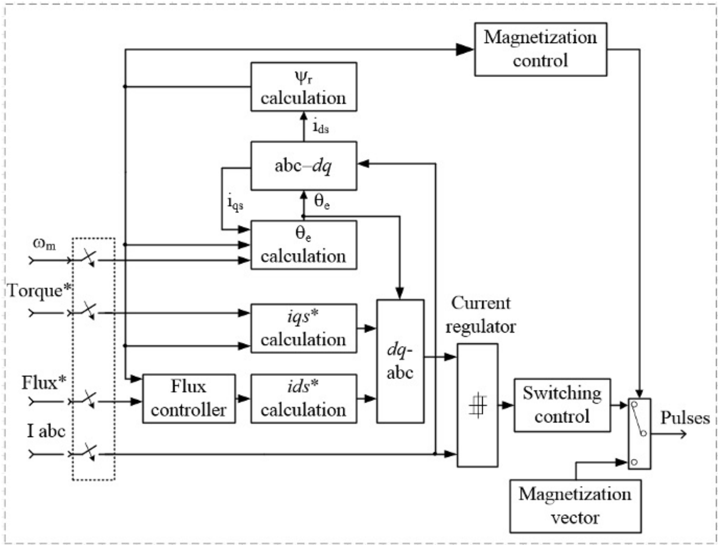
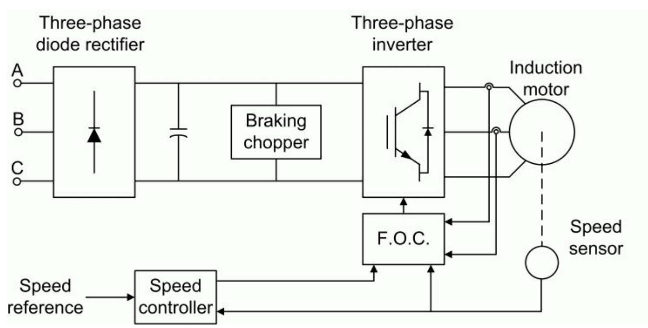


Figure 2 Implementation of FOC for IM



2.1 Induction motor model

The following fifth order model presents complete dynamics of IM with assumptions of each mutual inductances (MI) and linear magnetic circuit (Sajedi et al., 2011):

$$\frac{d\omega}{dt} = \mu\psi_d i_q - \frac{T_L}{J}, \tag{1}$$

$$\frac{d\psi_d}{dt} = -\alpha\psi_d + \alpha M i_d \tag{2}$$

$$\frac{d}{i_d}dt = -\gamma i_d + \alpha\beta\psi_d, \tag{3}$$

$$\frac{d}{i_q}dt = -\gamma i_q - \beta\eta_p\omega\psi_d - \eta_p\omega i_d - \alpha M\frac{i_d i_q}{\psi_d} + \frac{1}{\sigma L}u_q, \tag{4}$$

$$\frac{d\rho}{dt} = -\eta_p\omega + \alpha M \frac{i_q}{\psi_d} \tag{5}$$

$$T = \mu i_q \psi_d \tag{6}$$

The d-q axis segments of the motor flux are represented by ψ_d and ψ_q. The motor speed is shown by ω. The motor voltage's d-q axis segments are u_d and u_q and the stator current segments are i_d, i_q.

The motor pair poles is designated by n_p, the MI by M, the respective stator and rotor resistances are R_s and R_r. Further, the respective stator and rotor self inductances L_s and L_r. The load torque is designated by T_L.

$$\sigma = 1 - \frac{M^2}{L_r L_s}, \tag{7}$$

$$\alpha = \frac{R_r}{L_r},\ \beta = \frac{M}{\sigma L_s L_r},\ \mu = \frac{\eta_p M}{J L_r} \tag{8}$$

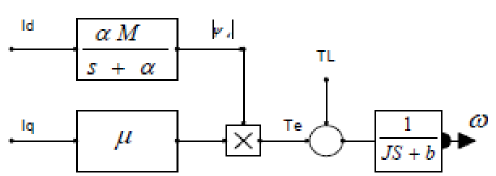
$$p = \arctan\frac{\psi_b}{\psi_a} \text{ and } \gamma = \frac{M R_r}{\sigma L_s {L_r}^2} + \frac{R_s}{\sigma L_s}, \tag{9}$$

2.2 Field oriented vector control model

The IMs are operated under different control strategies. The specific strategy to be embraced relies upon the type of the IM. The AC motor current will split into two particular segments: I_d or the flux generating current segment and I_q or the torque generating current segment. The vector sum of I_d and I_q current segments is the aggregate current. The torque developed within motor is relied upon the cross multiplication of above vectors (Kirschen et al., 1985). Distinct phenomena in drive system actualise distinct levels of controls over one or more of these segments and the vector angle amid them (Gastli and Matsui, 1992).

It is clear that the flux and torque in FOC are independent to each other. The outcome of the FOC are beneficial in terms of torque regulations, higher starting torque, smooth speed, higher low speed torque, and higher shock load capability. The decoupling control FOC is presented in Figure 3 (Sajedi et al., 2011).

Figure 3 FOC decoupling control



The application of voltage state feedback may introduce nonlinearities. The voltage is the directive action to wipe these nonlinearities. So V_d directly regulates ψ_d, once ψ_d becomes constant, the equation of speed becomes linear then voltage V_q

controls the speed ω straightforwardly. The voltage feedback equations are presented in equations (10) and (11) (Ho and Sen, 1988):

$$u_d = \frac{\beta}{L_r}\left(-p\omega i_q - \frac{M}{T_r}\frac{{i_q}^2}{\psi_d} - \frac{M}{\beta T_r} + V_d\right), \tag{10}$$

$$u_d = \frac{\beta}{L_r}\left(p\omega i_r + \frac{M}{T_r}\frac{I_d i_q}{\psi_d} + \frac{M}{T_r}(p\omega\psi_{rd}) + V_q\right), \tag{11}$$

The equations (12)–(29) represents closed loop system:

$$\frac{d\omega}{dt} = \mu\psi_d i_q - \frac{T_I}{J}, \tag{12}$$

$$\frac{d\psi}{dt} = -\alpha\psi_d + \alpha M i_d, \tag{13}$$

$$\frac{di_d}{dt} = -\alpha i_d + V_d, \tag{14}$$

$$\frac{di_d}{dt} = -\alpha i_q + V_q, \tag{15}$$

Where the time constant is $T_r = \frac{L_r}{R_r}$. The stator current is represented by equation (16),

$$I_s = \frac{T_r}{M}\frac{d\varphi}{dt} + + \frac{1}{M}\varphi, \tag{16}$$

Where, φ is the flux reference value. The voltage equations are mentioned as equations (17) and (18):

$$V_d = (I_{dr} - I_d)\left(K_6 + \frac{K_7}{s}\right), \tag{17}$$

$$V_q = (I_{qr} - I_q)\left(K_8 + \frac{K_9}{s}\right), \tag{18}$$

Where, I_d and I_q are the actual $d - q$ stator current segments respectively. The current equations in terms of flux and speed set points are as per the equations and :

$$I_{dr} = K_1 \int 0tot(f_{ref} - \psi_d)dt + K_2(f_{ref} - \psi_d) + \frac{\psi_d}{M}, \tag{19}$$

$$I_{qr} = K_3(\theta_{ref} - \theta) + K_4 \int 0tot(\theta_{ref} - \theta)dt + K_5(\omega_{ref} - \omega_d) + {\alpha_{ref}}^{\frac{1}{\mu}}, \tag{20}$$

The target is to optimise the execution of the FOC strategy by enhancing the motor efficiency by finding the optimum reference flux. Further, it is to select the optimal flux set point using optimal selection of the controller gains $(K_1, K_2, K_3, K_4$ and $K_5)$. This optimisation is done using landmark ABC.

3 Landmark artificial bee colony algorithm

The available literature reports that the convergence speed of ABC_{algo} is low. To embellish the convergence capability of the basic version of ABC_{algo}, a landmark operator is incorporated with it and the proposed algorithm is titled as landmark ABC (LMABC). The ABC_{algo} and proposed LMABC are discussed in following subsections:

3.1 Artificial bee colony algorithm

The ABC_{algo} inspired by the collective well-informed food foraging activities of the natural bees is a kind of the swarm intelligence (SI)-based algorithms (Karaboga and Basturk, 2007). In ABC_{algo} possible solution for the optimisation problem is represented by food source's (F_{Source}) position and the nectar amount of a (F_{Source}) resembles and correlates to the fitness of the solution (Karaboga and Akay, 2009).

The artificial bees are separated into three groups of a colony that is employed bees, onlooker bees, and scout bees. The onlooker bees or employed bees in number are similar to the F_{Source}. The employed bees arbitrarily search for the positions of the F_{Source} and propagate its information with the onlooker bee which stays at hive to follow information from the employed bees. At the exhaust situation of existing F_{Source} scout bees searches the new F_{Source} arbitrarily (Abu-Mouti and El-Hawary, 2012; Karaboga, 2005).

similar to other populous relied upon metaheuristic algorithms ABC_{algo} is also an iterative process. It performs cycles of the four phases titled as Initialisation of the populous phase $(Init_{phase})$, employed bees phase (EB_{phase}), onlooker bees phase (OB_{phase}) and scout bees phase (SB_{phase}) (Akay and Karaboga, 2012). The explanation of the phases is given below:

- $Init_{phase}$: initially ABC_{algo} generates an evenly scattered initial populous of SN solutions where each solution x_i $(i = 1, 2, \ldots, SN)$ is a D-dimensional vector. Here D is the number of variables in the optimisation problem and x_i is the i^{th} F_{Source} in the populous. Generation of each F_{Source} is as follows:

  $$x_{ij} = x_{minj} + rand[0, 1](x_{maxj} - x_{minj}) \tag{21}$$

  where x_{minj} and x_{maxj} are limits of x_i in j^{th} direction and rand [0, 1] is an evenly scattered arbitrary number in the range [0, 1].

- EB_{phase}: the individual's knowledge and the fitness value (FV) of the new solution, i.e., nectar amount is computed to decide the updating of present solution. The bee modifies its position with the new one and discards the old one (Akay and Karaboga, 2012), if the FV of the new solution is greater than that of the old solution. For i^{th} candidate the position modifying equation in this phase is:

  $$v_{ij} = x_{ij} + \phi_{ij}(x_{ij} - x_{kj}) \tag{22}$$

  where $k \in \{1, 2, \ldots, SN\}$ and $j \in \{1, 2, \ldots, D\}$ are randomly chosen indices. k must be different from i. ϕ_{ij} is a random number amid [–1, 1].

- OB_{phase}: the employed bees transfer the knowledge related to the new fitness. It communicates about nectar of the new solutions (F_{Source}) to onlooker bees. After evaluating this information onlooker bees select a solution with a probability, about its fitness. The probability p_i is calculated using following expression which is a a function of fitness:-

$$p_i = \frac{fit_i}{\sum_{i=1}^{SN} fit_i} \tag{23}$$

The FV of the solution i is fit_i. Like EB_{phase}, it memorises a modification in the position and checks for the fitness of the candidate source. If the new fitness is greater than that of the earlier one using greedy selection mechanism (GSM), the bee memorises the new acquired position and forgets the old one.

- SB_{phase}: if an F_{Source} does not modify its position up to a predefined limit, i.e., number of cycles, the F_{Source} is assumed to be discarded and then SB_{phase} starts.

In this phase F_{Source} is replaced by a randomly chosen F_{Source} with in the particularised area. Assume that the discarded source is x_i and $j \in \{1, 2, \ldots, D\}$ then the scout bee exchanges this F_{Source} with x_i. This process is mathematically presented as follows:

$$x_i^j = x_{min}^j + rand[0,1](x_{max}^j - x_{min}^j) \tag{24}$$

where x_{min}^j and x_{max}^j are bounds of x_i in j^{th} direction.

The above analysis reveals three control parameters in ABC_{algo}: first the number of F_{Source}, SN (equal to number of onlooker or employed bees), second the value of $limit$, and third the maximum number of cycles MCN.

In the ABC_{algo}, employed bees and onlooker bees are responsible for the exploitation is process while the scout bees perform the exploration process.

3.2 Landmark artificial bee colony algorithm

The optimisation algorithm's performance depends upon the two basic concepts, i.e., exploration and exploitation. The exploration is used to inspect the entire area of the foraging region to find out the promising solution. While exploitation ability uses the previous knowledge and intelligence to refine the already explored areas to discover the quality solution. For the efficient execution of any optimisation algorithm, there is always a requirement to maintain an optimum balance amid both the above mentioned concepts. The review of literature show that ABC_{algo} is favourable at exploration but bad at exploitation (Zhu and Kwong, 2010).

To enhance the exploitation capability of the basic version of ABC_{algo}, a landmark operator-based phase is incorporated. The proposed phase is termed as landmark phase and the proposed algorithm is termed as landmark ABC (LMABC).

The complete working of the landmark phase is as follows: to enhance the convergence ability of ABC_{algo}, a landmark (Centre solution) is discovered and all the solutions attract towards that landmark. The centre for the search space is calculated using the following equation:

$$x_{ct} = \frac{x_i t * fit_i}{\sum_{i=1}^{SN} fit_i} \tag{25}$$

Here, fit_i is the fitness of the i^{th} solution of the swarm. $x_i t$ is the position of i^{th} solution in the swarm. SN is the size of the swarm.

Here, the centre of the search region is calculated using equation (25). After discovering the centre of the search region, each solution is updated by using equation (26):

$$x_i(t+1) = x_i t + R(x_c t - x_i t) \tag{26}$$

Here, $R \subseteq (0, 1)$ The fitness of the newly generated solution is evaluated using GSM which is applied amid the old solution and the newly generated solution. In every iteration, subsequent to decide the centre of the solutions, the number of swarm size is cut to a half. The solutions who are at a distance from the target, are supposed to follow near the target ones. The updating rule is given by equation (27):

$$SN = SN \frac{(T-1)}{2} \tag{27}$$

The proposed LMABC algorithm is branched in to four phases. Landmark phase is embedded after the SB_{phase} of the algorithm. Relied upon the above analysis, the pseudo code of the propound LMABC algorithm is shown in Algorithm 1.

Algorithm 1 Landmark artificial bee colony ($LMABC$)

Initialise the parameters: MCN (maximum number of cycles), D (dimension of the problem), SN (swarm size), R;
$Init_{phase}$, x_i where (i = 1, 2, ..., SN) by using equation (21);
cycle = 1;
while cycle <> MCN **do**
  EB_{phase};
  OB_{phase};
  SB_{phase};
  Landmark phase: /* Explained as follows:*/
  for each solution **do**
    Evaluate the centre of the solutions applying the equation (25)
    Modify the position of the solutions applying the equation (26);
    Exercising the GSM between the old position and the new position of the F_{Source};
    $(SN = SN \div 2)$
  end for
  Memorise the best F_{Source} found as yet;
  cycle=cycle+1;
end while
Output the best solution found so far.

4 Comparison and analysis of result

The execution of propound algorithm LMABC is accessed on 25 different continuous optimisation benchmark functions (f_1 to f_{25}) persisting non-similar degrees of complexity and multi modality as shown in Table 1. To check the competitiveness of LMABC, it is compared with, ABC_{algo} (Karaboga, 2005), particle swarm optimisation ($PSO-2011$) (Clerc and Kennedy, 2011), differential evolution (DE) algorithms (Price, 1996) and six significant variants of ABC_{algo} algorithm namely, Gbest-guided ABC ($GABC$) algorithm (Zhu and Kwong, 2010), modified ABC ($MABC$) algorithm (Bansal et al., 2013), and lévy flight ABC ($LFABC$) algorithm (Sharma et al., 2015), disruption ABC algorithm ($DiABC$), black hole ABC algorithm ($BHABC$), and ($HABC$). The experimental setting is depicted in Subsection 4.1.

4.1 Experimental setting

The experimental setting adopted is as follows:

- the number of simulations/run =100
- colony size NP = 50 and number of F_{Source} SN = NP/2,
- $C_0 = 60$
- $\rho = 10^{-10}$
- $\phi_{ij} = rand[-1, 1]$ and limit = dimension × number of F_{Source} = $D \times SN$ (Akay and Karaboga, 2012)
- parameter setting for other considered algorithms are identical to their legitimate work (Banharnsakun et al., 2011; Bansal et al., 2013; Kennedy and Eberhart, 1995; Storn and Price, 1995; Sharma et al., 2015; Zhu and Kwong, 2010).

4.2 Results comparison

The availed outcomes are shown in Table 2 expressed and evaluated on four analytical parameters. These are success rate (SR), average number of function evaluations (AFE), mean error (ME), and standard deviation (SD).

The LMABC is compared with ABC_{algo} and its significant variants, it is also compared with DE and PSO. The results are tabulated in Table 2. The obtained outcomes reveal that LMABC is a competing algorithm and performs better for majority of the optimisation functions regardless of their characteristics.

Mann-Whitney U rank sum test (MWU) (Mann and Whitney, 1947), acceleration rate (AR), and boxplot analysis (BP) are also applied on the considered algorithms. MWU test is performed on AFEs. For all the examined algorithms the test is executed at 5% significance level ($\alpha = 0.05$) and the obtained outcomes for 100 simulations are tabulated in Table 4. In this table, '↑' symbol displays that $LMABC$ is quite better as other investigated algorithm while '↓' symbol represents that the other examined algorithm is better. The $LMABC$ dominates as accessed with all other examined algorithms for eight functions including f_2, f_5, f_{15}–f_{20}, and f_{25}. Execution of $LMABC$ is better than basic ABC_{algo} for all 25 functions f_1–f_{25}. The $LMABC$

executes better than $MABC$ for 24 functions f_1–f_{18} and f_{20}–f_{25}. In comparison with $BSFABC$, $LMABC$ shows better results for 24 functions f_1–f_{13}, f_{15}–f_{25}. In comparison with $LFABC$, the $LMABC$ performs better for 15 functions f_2, f_4, f_8, f_{12}–f_{18}, f_{20}–f_{22}, f_{24} and f_{25}. The $LMABC$ shows better results for 23 functions when compared with $HABC$ algorithm f_1–f_{20}, f_{22}, f_{23} and f_{25}. The $LMABC$ shows better results for 21 functions when accessed with $BHABC$ algorithm f_1–f_7, f_9, f_{10}, f_{13}–f_{21}, and f_{23}–f_{25}. The $LMABC$ shows better results for 20 functions when accessed with $DiABC$ algorithm f_1–f_2, f_5, f_6–f_{18}, f_{20}–f_{23} and f_{25}. The $LMABC$ shows better results for 21 functions when compared with $PSO-2011$ and DE algorithm, f_1–f_{10}, f_{12}–f_{18}, f_{20}, f_{21}, f_{23}–f_{25}.

The above analysis shows that $LMABC$ will be among an important member in the field of SI-based algorithms.

Further, the convergence speed (C_{Speed}) of examined algorithms is accessed by analysis of AFEs. There is a contrary relationship amid AFEs and C_{Speed}, for smaller AFEs the C_{Speed} will be higher and vice-versa. To curtail the effects of stochastic nature of algorithm, the AFEs are averaged for 100 runs for each examined test problems. The C_Speed is accessed using AR for the examined algorithms. The AR which is evaluated as follows:

$$AR = \frac{AFE_{ALGO}}{AFE_{LMABC}}, \tag{28}$$

Here, $AR > 1$, represents $LMABC$ is speedier as compared to examined algorithm. The AR outcomes are shown in Table 3. The outcomes in Table 3 presents that for most of the examined benchmark test functions, $LMABC$ converge speedier than the examined algorithms.

The BP analysis has also been processed out for all examined algorithms for assessing in terms of overall performance. In BP analysis tool (Williamson et al., 1989), graphical distribution of empirical data is properly shown. The BP for $LMABC$ and other investigated algorithms are presented in Figure 4. It is evident from this figure that $LMABC$ dominates than the other examined strategies as interquartile range and median are quite low.

Figure 4 Boxplots graphs for average number of function evaluation (see online version for colours)

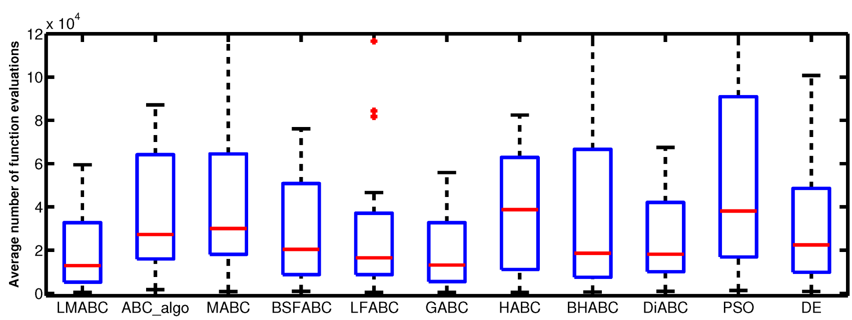


Table 1 TP: the benchmark test problems, D: dimensions, C: characteristic, U: unimodal, M: multimodal, S: separable, N: non-separable, AE: acceptable error

TP	*Optimisation function*	*Search range*	*Optimum Value*	*D*	*AE*	*C*
Sphere	$f_1(x) = \sum_{i=1}^{D} x_i^2$	[–5.12, 5.12]	$f(\vec{0}) = 0$	30	1.0E-05	S, U
De Jong f4	$f_2(x) = \sum_{i=1}^{D} i \cdot (x_i)^4$	[–5.12, 5.12]	$f(\vec{0}) = 0$	30	1.0E-05	S, M
Ackley	$f_3(x) = -20 + e + exp(-\frac{0.2}{D}\sqrt{\sum_{i=1}^{D} {x_i}^3})$ $- \exp(\frac{1}{D}\sum_{i=1}^{D}\cos(2\pi x_i)x_i)$	[–1 1]	$f(\vec{0}) = 0$	30	1.0E-05	M, N
Alpine	$f_4(x) = \sum_{i=1}^{D} \lvert x_i \sin x_i + 0.1x_i \rvert$	[–10, 10]	$f(\vec{0}) = 0$	30	1.0E-05	M, S
Michalewicz	$f_5(x) = -\sum_{i=1}^{D} \sin x_i (\sin(\frac{i{x_i}^2}{\pi})^{20})$	$[0, \pi]$	$f_{min} = -9.66015$	10	1.0E-05	N, M
Exponential	$f_6(x) = -(exp(-0.5\sum_{i=1}^{D} {x_i}^2)) + 1$	[–1, 1]	$f(\vec{0}) = -1$	30	1.0E-05	N, M
Schewel	$f_7(x) = \sum_{i=1}^{D} \lvert x_i \rvert + \prod_{i=1}^{D} \lvert x_i \rvert$	[–10, 10]	$f(\vec{0}) = 0$	30	1.0E-05	N, U
Levy montalvo 1	$f_8(x) = \frac{\pi}{D}(10\sin^2(\pi y_1) + \sum_{i=1}^{D-1}(y_i - 1)^2(1 + 10\sin^2(\pi y_{i+1}))$ $+ (y_D - 1)^2)$, where $y_i = 1 + \frac{1}{4}(x_i + 1)$	[–10, 10]	$f(\vec{-1}) = 0$	30	1.0E-05	N, M
Levy montalvo 2	$f_9(x) = 0.1(\sin^2(3\pi x_1) + \sum_{i=1}^{D-1}(x_i - 1)^2 \times (1 + \sin^2(3\pi x_{i+1}))$ $+ (x_D - 1)^2(1 + \sin^2(2\pi x_D))$	[–5, 5]	$f(\vec{1}) = 0$	30	1.0E-05	N, M
Ellipsoidal	$f_{10}(x) = \sum_{i=1}^{D}(x_i - i)^2$	[–30, 30]	$f(1, 2, 3, \ldots, D) = 0$	30	1.0E-05	U, S
Beale function	$f_{11}(x) = [1.5 - x_1(1 - x_2)]^2 + [2.25 - x_1(1 - x_2^2)]^2$ $+ [2.625 - x_1(1 - x_2^3)]^2$	[–4.5, 4.5]	$f(3, 0.5) = 0$	2	1.0E-05	N, M
Colville function	$f_{12}(x) = 100[x_2 - x_1^2]^2 + (1 - x_1)^2 + 90(x_4 - x_3^2)^2$ $+ (1 - x_3)^2 + 10.1[(x_2 - 1)^2 + (x_4 - 1)^2]$ $+ 19.8(x_2 - 1)(x_4 - 1)$	[–10, 10]	$f(\vec{1}) = 0$	4	1.0E-05	N, M
Branins's function	$f_{13}(x) = a(x_2 - bx_1^2 + cx_1 - d)^2 + e(1 - f)\cos x_1 + e$	$-5 \le x_1 \le 10$, $0 \le x_2 \le 15$	$f(-\pi, 12.275) = 0.3979$	2	1.0E-05	N, M
2D tripod function	$f_{14}(x) = p(x_2)(1 + p(x_1)) + \lvert(x_1 + 50p(x_2)(1 - 2p(x_1)))\rvert$ $+ \lvert(x_2 + 50(1 - 2p(x_2)))\rvert$	[–100, 100]	$f(0, -50) = 0$	2	1.0E-04	N, M
Shifted sphere	$f_{15}(x) = \sum_{i=1}^{D} z_i^2 + f_{bias}$, $z = x - o$, $x = [x_1, x_2, \ldots, x_D]$, $o = [o_1, o_2, \ldots o_D]$	[–100, 100]	$f(o) = f_{bias} = -450$	10	1.0E-05	S, M
Shifted Griewank	$f_{16}(x) = \sum_{i=1}^{D} \frac{z_i^2}{4000} - \prod_{i=1}^{D} \cos(\frac{z_i}{\sqrt{i}}) + 1 + f_{bias}$, $z = (x - o)$, $x = [x_1, x_2, \ldots x_D]$, $o = [o_1, o_2, \ldots, o_D]$	[–600, 600]	$f(o) = f_{bias} = -180$	10	1.0E-05	M, N
Shifted Ackley	$f_{17}(x) = -20\exp(-0.2\sqrt{\frac{1}{D}\sum_{i=1}^{D} z_i^2})$ $- \exp(\frac{1}{D}\sum_{i=1}^{D}\cos(2\pi z_i)) + 20 + e + f_{bias}$, $z = (x - o)$, $x = (x_1, x_2, \ldots, x_D)$, $o = (o_1, o_2, \ldots, o_D)$	[–32, 32]	$f(o) = f_{bias} = -140$	10	1.0E-05	S, M

Table 1 TP: the benchmark test problems, D: dimensions, C: characteristic, U: unimodal, M: multimodal, S: separable, N: non-separable, AE: acceptable error (continued)

Test problem	*Objective function*	*Search range*	*Optimum value*	*D*	*AE*	*C*
Six-hump camel back	$f_{18}(x) = (4 - 2.1x_1^2 + x_1^4/3)x_1^2 + x_1x_2 + (-4 + 4x_2^2)x_2^2$	[–5, 5]	$f(-0.0898, 0.7126) = -1.0316$	2	1.0E-05	N, M
Easom's function	$f_{19}(x) = -\cos x_1 \cos x_2 e^{((-(x_1-\pi)^2-(x_2-\pi)^2))}$	[–10, 10]	$f(\pi, \pi) = -1$	2	1.0E-13	S, M
Dekkers and Aarts	$f_{20}(x) = 10^5 x_1^2 + x_2^2 - (x_1^2 + x_2^2)^2 + 10^{-5}(x_1^2 + x_2^2)^4$	[–20, 20]	$f(0, 15) = f(0, -15) = -24777$	2	5.0E-01	N, M
McCormick	$f_{21}(x) = \sin(x_1 + x_2) + (x_1 - x_2)^2 - \frac{3}{2}x_1 + \frac{5}{2}x_2 + 1$	$-1.5 \leq x_1 \leq 4$, $-3 \leq x_2 \leq 3$	$f(-0.547, -1.547) = -1.9133$	30	1.0E-04	N, M
Meyer and Roth Problem	$f_{22}(x) = \sum_{i=1}^{5}(\frac{x_1x_3t_i}{1+x_1t_i+x_2v_i} - y_i)^2$	[–10, 10]	$f(3.13, 15.16, 0.78) = 0.4E\text{-}04$	3	1.0E-03	U, N
Shubert	$f_{23}(x) = -\sum_{i=1}^{5} i\cos((i+1)x_1 + 1)\sum_{i=1}^{5} i\cos((i+1)x_2 + 1)$	[–10, 10]	$f(7.0835, 4.8580) = -186.7309$	2	1.0E-05	S, M
Sinusoidal	$f_{24}(x) = -[A\prod_{i=1}^{n}\sin(x_i - z) + \prod_{i=1}^{n}\sin(B(x_i - z))]$, $A = 2.5, B = 5, z = 30$	[0, 180]	$f(90 + z) = -(A + 1)$	10	1.0E-02	N, M
Moved axis parallel hyper-ellipsoid	$f_{25}(x) = \sum_{i=1}^{D} 5i \times x_i^2$	[–5.12, 5.12]	$f(x) = 0; x(i) = 5 \times i, i = 1 : D$	30	1.0E-15	U, S

Table 2 Assessment of the results of test problems

Test function	*Measure*	*LMABC*	*ABC*	*MABC*	*BSFABC*	*LFABC*	*GABC*	*HABC*	*BHABC*	*DiABC*	*PSO-2011*	*DE*
f_1	SD	1.86E-06	9.48E-07	2.15E-06	2.02E-06	1.73E-06	1.81E-06	1.97E-06	1.44E-06	2.51E-06	6.10E-07	8.24E-07
	ME	8.06E-06	8.95E-06	7.49E-06	8.17E-06	8.39E-06	8.11E-06	7.33E-06	8.53E-06	7.34E-06	9.33E-06	9.06E-06
	AFE	14530.91	22359	30063	20409	16733.85	14347.5	39341.69	22304.92	16823	38101.5	22444
	SR	100	100	100	100	100	100	100	100	100	100	100
f_2	SD	2.86E-06	1.27E-06	3.12E-06	3.11E-06	3.02E-06	2.72E-06	1.66E-06	2.63E-06	2.88E-06	8.62E-07	8.51E-07
	ME	6.08E-06	8.63E-06	5.31E-06	4.90E-06	6.62E-06	5.51E-06	4.49E-07	5.75E-06	5.52E-06	9.03E-06	9.01E-06
	AFE	8338.84	22584	24524.5	9578.5	9556.12	8388	44055.41	8687.05	19853	32596.5	20859.5
	SR	100	100	100	100	100	100	100	100	100	100	100
f_3	SD	1.05E-06	4.31E-07	2.03E-06	1.54E-06	2.92E-04	1.12E-06	1.47E-06	1.99E-06	1.64E-06	3.66E-07	4.42E-07
	ME	9.05E-06	9.51E-06	7.71E-06	8.63E-06	8.95E-06	8.82E-06	8.62E-06	8.05E-06	8.55E-06	9.69E-06	9.46E-06
	AFE	30460.11	43333	72833	48726.5	36042.22	30438.5	38523.15	105493.04	29537	77352	43100.5
	SR	100	100	100	100	100	100	100	100	100	100	100
f_4	SD	1.79E-06	7.61E-04	6.18E-06	2.01E-06	5.12E-05	2.22E-06	6.44E-04	1.63E-06	2.58E-06	3.23E-07	4.48E-07
	ME	8.33E-06	1.01E-03	7.68E-06	8.10E-06	1.61E-05	7.78E-06	2.04E-04	8.62E-06	6.72E-06	9.63E-06	9.50E-06
	AFE	59491.15	199704.63	148190	76098.5	81698.1	55904	175856.35	60825.96	46753	93046.5	61895.5
	SR	99	2	99	100	94	100	57	100	100	100	100
f_5	SD	3.38E-06	3.28E-06	3.19E-06	3.65E-06	2.57E-02	3.67E-06	3.55E-06	3.63E-06	3.69E-06	2.34E-01	4.26E-02
	ME	4.28E-06	6.70E-06	3.21E-06	3.84E-06	6.08E-03	3.79E-06	4.44E-06	4.43E-06	3.99E-06	3.12E-01	4.21E-02
	AFE	20075.34	36367.7	43095.01	29192.27	40306.64	21914.86	28237.43	27594.86	58170.32	198326	171411
	SR	100	100	100	100	89	100	100	100	100	3	20
f_6	SD	1.89E-06	6.96E-07	2.26E-06	2.34E-06	3.10E-02	1.75E-06	3.96E-07	2.34E-06	2.50E-06	6.15E-07	9.15E-07
	ME	8.02E-06	9.11E-06	7.36E-06	7.12E-06	8.20E-06	8.13E-06	2.31E-07	7.49E-06	7.53E-06	9.33E-06	8.98E-06
	AFE	11816.77	16647.5	18707	16974	14033.43	11746	40644.92	18629.47	10872	28227.5	17269
	SR	100	100	100	100	100	100	100	100	100	100	100
f_7	SD	9.16E-07	3.85E-07	1.10E-06	1.03E-06	1.12E-04	6.90E-07	8.41E-07	9.22E-07	1.71E-06	3.09E-07	5.99E-07
	ME	9.23E-06	9.49E-06	8.99E-06	9.16E-06	8.98E-06	9.30E-06	9.18E-06	9.39E-06	8.32E-06	9.61E-06	9.39E-06
	AFE	27648.92	32888.5	52966	41646	31122.67	27636	62156.74	122748.31	28319	70794.5	44943
	SR	100	100	100	100	100	100	100	100	100	100	100

Table 2 Assessment of the results of test problems (continued)

Test function	*Measure*	*LMABC*	*ABC*	*MABC*	*BSFABC*	*LFABC*	*GABC*	*HABC*	*BHABC*	*DiABC*	*PSO-2011*	*DE*
f_8	SD	1.91E-06	6.67E-07	2.78E-06	2.13E-06	2.10E-02	1.97E-06	2.25E-06	1.81E-06	2.08E-06	1.77E-02	9.48E-07
	ME	7.98E-06	9.27E-06	6.78E-06	7.32E-06	8.48E-06	7.92E-06	1.65E-06	7.66E-06	7.44E-06	3.12E-03	9.02E-06
	AFE	12902.57	22791.5	26672	19816	15182.17	13160	45830.27	12124.57	18183.5	37764.5	19550
	SR	100	100	100	100	100	100	100	100	100	97	100
f_9	SD	1.87E-06	7.64E-07	2.48E-06	2.24E-06	2.10E-04	2.03E-06	2.80E-06	1.90E-06	2.63E-06	3.58E-03	1.87E-03
	ME	7.99E-06	9.21E-06	6.89E-06	7.54E-06	8.44E-06	7.56E-06	3.29E-06	8.26E-06	6.96E-06	1.33E-03	3.38E-04
	AFE	14363.22	20902.5	28876	21940	16459.73	14290	38765.3	16425.69	17578.5	56513.5	25989
	SR	100	100	100	100	100	100	100	100	100	88	97
f_{10}	SD	1.67E-06	6.73E-07	2.34E-06	2.35E-06	2.78E-06	1.89E-06	1.41E-06	2.05E-06	2.55E-06	5.56E-07	8.74E-07
	ME	8.24E-06	9.24E-06	7.05E-06	7.52E-06	8.49E-06	8.05E-06	8.72E-06	8.08E-06	7.22E-06	9.33E-06	8.93E-06
	AFE	16667.33	26766	40934.5	24219.5	18646.7	16665	65207.11	23001.64	26228	44306	27365.5
	SR	100	100	100	100	100	100	100	100	100	100	100
f_{11}	SD	2.92E-06	2.90E-06	3.18E-06	1.83E-06	2.94E-06	3.00E-06	3.01E-06	2.87E-06	2.85E-06	2.81E-06	2.81E-06
	ME	5.36E-06	5.32E-06	9.64E-06	8.29E-06	7.42E-06	6.06E-06	4.70E-06	5.65E-06	7.67E-06	4.96E-06	4.74E-06
	AFE	9386.96	9803.94	50415.4	16954.05	4274.48	9135.64	14879.66	7524.87	23170.97	2753.5	1415.5
	SR	100	100	98	100	100	100	100	100	99	100	100
f_{12}	SD	2.05E-02	1.27E-02	3.49E-02	1.08E-01	1.58E-04	1.64E-02	3.25E-02	1.68E-03	3.72E-02	2.24E-04	1.66E-01
	ME	1.92E-02	1.36E-02	2.15E-02	1.58E-01	9.30E-04	1.83E-02	2.64E-02	8.36E-03	4.29E-02	8.13E-04	5.34E-02
	AFE	146271.11	197786.36	200028.8	200025.94	116636.98	200027.39	200241.95	76010.84	174609.95	48776.5	36451.5
	SR	46	3	0	0	97	0	2	100	22	100	84
f_{13}	SD	6.62E-06	7.64E-06	6.67E-06	7.10E-06	1.15E-05	6.66E-06	6.41E-06	6.49E-06	2.72E-02	3.28E-06	6.34E-06
	ME	5.98E-06	6.67E-06	5.68E-06	6.32E-06	5.91E-06	5.57E-06	6.35E-06	5.53E-06	2.74E-03	5.81E-06	5.52E-06
	AFE	1074.84	34517.47	21661.45	2128.77	14812.69	1185.37	32597.49	1613.58	1576	17240	25736
	SR	100	84	90	100	93	100	84	100	100	93	88

Table 2 Assessment of the results of test problems (continued)

Test function	*Measure*	*LMABC*	*ABC*	*MABC*	*BSFABC*	*LFABC*	*GABC*	*HABC*	*BHABC*	*DiABC*	*PSO-2011*	*DE*
f_{14}	SD	2.48E-05	8.98E-04	1.30E-04	2.39E-05	3.32E-01	2.49E-05	2.33E-05	2.32E-05	2.36E-05	2.71E-01	2.71E-01
	ME	6.50E-05	2.51E-04	8.48E-05	6.43E-05	1.00E-01	6.51E-05	6.18E-05	6.06E-05	5.88E-05	8.01E-02	8.01E-02
	AFE	7990.98	51848.2	10833.17	7927.03	22793.84	8315.55	8640.31	10093	17764.14	29745.5	19150.5
	SR	100	92	96	100	91	100	100	100	100	92	92
f_{15}	SD	2.37E-06	1.51E-06	2.37E-06	2.61E-06	3.90E+00	2.19E-06	2.22E-06	2.40E-06	2.58E-06	1.50E-06	1.71E-06
	ME	7.13E-06	8.10E-06	6.65E-06	6.97E-06	7.67E-06	6.94E-06	7.59E-06	7.40E-06	6.67E-06	8.29E-06	7.95E-06
	AFE	5539.72	8704.5	18137.5	9042.5	6249.21	5568	16722.88	8505.59	8631	15785.5	10353.5
	SR	100	100	100	100	100	100	100	100	100	100	100
f_{16}	SD	7.35E-04	1.62E-03	6.15E-03	1.87E-03	2.60E+02	9.80E-04	1.85E-03	2.21E-03	2.62E-03	2.87E-02	1.53E-02
	ME	7.88E-05	3.79E-04	4.98E-03	4.73E-04	2.53E-04	1.03E-04	4.24E-04	6.96E-04	8.67E-04	4.05E-02	1.42E-02
	AFE	43290.24	87135.2	121497.59	67879.63	46629.99	44474.24	76828.96	66261.07	67534.17	197491	160664.5
	SR	99	95	54	94	97	99	95	91	90	2	25
f_{17}	SD	1.51E-06	8.93E-07	1.71E-06	1.98E-06	8.05E-01	1.44E-06	1.52E-06	2.00E-06	3.97E+00	1.05E-06	1.17E-06
	ME	8.28E-06	8.90E-06	8.06E-06	7.76E-06	8.31E-06	8.18E-06	8.39E-06	7.92E-06	3.99E-01	8.93E-06	8.81E-06
	AFE	9259.66	14030.53	31305	16704.5	10926.37	9317	11648.2	67768.62	10577	24630	15564.5
	SR	100	100	100	100	100	100	100	100	100	100	100
f_{18}	SD	1.14E-05	1.53E-05	1.48E-05	1.10E-05	3.00E-02	1.14E-05	8.72E-06	1.10E-05	1.05E-05	1.18E-05	1.49E-05
	ME	1.32E-05	1.40E-05	1.70E-05	1.20E-05	1.53E-05	1.42E-05	2.63E-05	1.33E-05	1.22E-05	1.75E-05	1.67E-05
	AFE	559.35	76995.51	102387.92	1017	84353.75	595	180402.84	809.71	1038	105570.5	100761
	SR	100	62	49	100	58	100	10	100	100	48	50
f_{19}	SD	4.85E-10	2.37E-03	3.02E-14	8.37E-05	1.00E-02	3.01E-14	4.26E-12	3.14E-14	8.29E-13	2.92E-14	2.80E-14
	ME	4.88E-11	9.94E-04	4.40E-14	3.09E-05	4.32E-14	4.92E-14	5.29E-13	4.58E-14	2.18E-13	4.82E-14	4.17E-14
	AFE	55221.35	199252.34	4578.41	186124.2	13861.99	42967.95	57131.78	83816.76	52418.01	9796.5	4798.5
	SR	99	1	100	16	100	100	98	100	93	100	100

Table 2 Assessment of the results of test problems (continued)

Test function	*Measure*	*LMABC*	*ABC*	*MABC*	*BSFABC*	*LFABC*	*GABC*	*HABC*	*BHABC*	*DiABC*	*PSO-2011*	*DE*
f_{20}	SD	5.70E-03	5.34E-03	5.18E-03	5.76E-03	1.15E-02	5.10E-03	5.30E-03	5.26E-03	5.35E-03	5.55E-03	4.80E-03
	ME	4.90E-01	4.90E-01	4.89E-01	4.91E-01	4.92E-01	4.89E-01	4.89E-01	4.91E-01	4.89E-01	4.92E-01	4.89E-01
	AFE	754.14	2376.59	2838.51	1407.52	756.4	760.5	1007.34	887.75	1411.5	5050	2123
	SR	100	100	100	100	100	100	100	100	100	100	100
f_{21}	SD	6.39E-06	6.57E-06	6.65E-06	6.36E-06	2.50E+00	6.92E-06	5.76E-06	5.94E-06	4.30E-02	6.86E-06	6.54E-06
	ME	8.93E-05	8.92E-05	8.92E-05	8.80E-05	9.02E-05	8.83E-05	8.26E-05	8.84E-05	4.41E-03	8.84E-05	8.78E-05
	AFE	611.55	1761.7	930.03	1176.04	552.29	620	511.44	738.1	1103	1445	971.5
	SR	100	100	100	100	100	100	100	100	100	100	100
f_{22}	SD	2.87E-06	2.87E-06	2.97E-06	2.61E-06	4.41E-02	3.01E-06	3.14E-06	2.66E-06	1.90E-01	2.93E-06	1.30E-05
	ME	1.95E-03	1.95E-03	1.95E-03	1.95E-03	1.95E-03	1.95E-03	1.94E-03	1.95E-03	2.10E-02	1.95E-03	1.95E-03
	AFE	4380.2	10949.9	17861.63	28795.15	4317.51	5316.83	9347.19	4174.71	15827.47	3092	3667.5
	SR	100	100	100	100	100	100	100	100	98	100	99
f_{23}	SD	5.63E-06	5.51E-06	5.33E-06	5.92E-06	4.22E-04	5.69E-06	5.78E-06	5.77E-06	5.67E-06	1.37E-03	5.16E-06
	ME	5.05E-06	4.58E-06	4.71E-06	5.36E-06	4.93E-06	5.07E-06	4.92E-06	5.18E-06	5.17E-06	3.12E-04	4.48E-06
	AFE	2406.37	27282.44	8956.03	4802.25	1594.27	2270.24	2662.65	7427.58	2781.62	90199	8287
	SR	100	100	100	100	100	100	100	100	100	71	100
f_{24}	SD	2.17E-03	9.37E-02	1.87E-03	2.00E-03	1.87E+00	2.11E-03	2.35E-03	2.17E-03	2.52E-03	2.94E-01	2.51E-01
	ME	7.62E-03	6.32E-01	7.86E-03	7.70E-03	8.01E-03	7.68E-03	6.28E-03	7.87E-03	6.97E-03	4.39E-01	5.49E-01
	AFE	42627.64	200034.58	62308.31	56952.5	23383.62	44057.89	4442.32	42781.85	40547.07	181097.5	199100
	SR	100	0	100	100	99	99	100	100	100	19	2
f_{25}	SD	6.87E-17	4.99E-17	2.56E-16	6.25E-17	5.36E-02	7.61E-17	1.00E-16	3.07E-11	3.47E-01	6.12E-17	8.35E-17
	ME	9.31E-16	9.34E-16	6.95E-16	9.36E-16	9.31E-16	9.27E-16	9.15E-16	1.73E-11	3.49E-02	9.29E-16	8.99E-16
	AFE	39572.19	59849.5	71130.5	62885	46127.25	39645.5	82468.5	200026.01	53220.5	104872.5	59436
	SR	100	100	100	100	100	100	100	0	100	100	100

Table 3 Assessment relied upon acceleration rate (AR)

TP	*LMABC vs. ABC_{algo}*	*LMABC vs. MABC*	*LMABC vs. BSFABC*	*LMABC vs. LFABC*	*LMABC vs. GABC*	*LMABC vs. HABC*	*LMABC vs. BHABC*	*LMABC vs. DiABC*	*LMABC vs. PSO-2011*	*LMABC vs. DE*
f_1	1.54	2.07	1.40	1.15	0.99	2.71	1.53	1.16	2.62	1.54
f_2	2.71	2.94	1.15	1.15	1.01	5.28	1.04	2.38	3.91	2.50
f_3	1.42	2.39	1.60	1.18	1.00	1.26	3.46	0.97	2.54	1.41
f_4	3.36	2.49	1.28	1.37	0.94	2.96	1.02	0.79	1.56	1.04
f_5	1.81	2.15	1.45	2.01	1.09	1.41	1.37	2.90	9.88	8.54
f_6	1.41	1.58	1.44	1.19	0.99	3.44	1.58	0.92	2.39	1.46
f_7	1.19	1.92	1.51	1.13	1.00	2.25	4.44	1.02	2.56	1.63
f_8	1.77	2.07	1.54	1.18	1.02	3.55	0.94	1.41	2.93	1.52
f_9	1.46	2.01	1.53	1.15	0.99	2.70	1.14	1.22	3.93	1.81
f_{10}	1.61	2.46	1.45	1.12	1.00	3.91	1.38	1.57	2.66	1.64
f_{11}	1.04	5.37	1.81	0.46	0.97	1.59	0.80	2.47	0.29	0.15
f_{12}	1.35	1.37	1.37	0.80	1.37	1.37	0.52	1.19	0.33	0.25
f_{13}	32.11	20.15	1.98	13.78	1.10	30.33	1.50	1.47	16.04	23.94
f_{14}	6.49	1.36	0.99	2.85	1.04	1.08	1.26	2.22	3.72	2.40
f_{15}	1.57	3.27	1.63	1.13	1.01	3.02	1.54	1.56	2.85	1.87
f_{16}	2.01	2.81	1.57	1.08	1.03	1.77	1.53	1.56	4.56	3.71
f_{17}	1.52	3.38	1.80	1.18	1.01	1.26	7.32	1.14	2.66	1.68
f_{18}	137.65	183.05	1.82	150.81	1.06	322.52	1.45	1.86	188.74	180.14
f_{19}	3.61	0.08	3.37	0.25	0.78	1.03	1.52	0.95	0.18	0.09
f_{20}	3.15	3.76	1.87	1.00	1.01	1.34	1.18	1.87	6.70	2.82
f_{21}	2.88	1.52	1.92	0.90	1.01	0.84	1.21	1.80	2.36	1.59
f_{22}	2.50	4.08	6.57	0.99	1.21	2.13	0.95	3.61	0.71	0.84
f_{23}	11.34	3.72	2.00	0.66	0.94	1.11	3.09	1.16	37.48	3.44
f_{24}	4.69	1.46	1.34	0.55	1.03	0.10	1.00	0.95	4.25	4.67
f_{25}	1.51	1.80	1.59	1.17	1.00	2.08	5.05	1.34	2.65	1.50

Note: TP – test problem.

Table 4 Assessment relied upon Mann-Whitney U rank sum test at significant level $\alpha = 0.05$ and average number of function evaluations

TP	*LMABC vs. ABC_{algo}*	*LMABC vs. MABC*	*LMABC vs. BSFABC*	*LMABC vs. LFABC*	*LMABC vs. GABC*	*LMABC vs. HABC*	*LMABC vs. BHABC*	*LMABC vs. DiABC*	*LMABC vs. PSO-2011*	*LMABC vs. DE*
f_1	↑	↑	↑	↑	↓	↑	↑	↑	↑	↑
f_2	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑
f_3	↑	↑	↑	↑	↓	↑	↑	↓	↑	↑
f_4	↑	↑	↑	↑	↓	↑	↑	↓	↑	↑
f_5	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑
f_6	↑	↑	↑	↑	↓	↑	↑	↓	↑	↑
f_7	↑	↑	↑	↑	↓	↑	↑	↑	↑	↑
f_8	↑	↑	↑	↑	↑	↑	↓	↑	↑	↑
f_9	↑	↑	↑	↑	↓	↑	↑	↑	↑	↑
f_{10}	↑	↑	↑	↑	↓	↑	↑	↑	↑	↑
f_{11}	↑	↑	↑	↓	↓	↑	↓	↑	↓	↓
f_{12}	↑	↑	↑	↓	↑	↑	↓	↑	↓	↓
f_{13}	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑
f_{14}	↑	↑	↓	↑	↑	↑	↑	↑	↑	↑
f_{15}	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑
f_{16}	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑
f_{17}	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑
f_{18}	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑
f_{19}	↑	↓	↑	↓	↓	↑	↑	↓	↓	↓
f_{20}	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑
f_{21}	↑	↑	↑	↓	↑	↓	↑	↑	↑	↑
f_{22}	↑	↑	↑	↓	↑	↑	↓	↑	↓	↓
f_{23}	↑	↑	↑	↓	↓	↑	↑	↑	↑	↑
f_{24}	↑	↑	↑	↓	↑	↓	↑	↓	↑	↑
f_{25}	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑
Total no. of + signs	25	24	24	19	15	23	21	20	21	21

Note: TP – test problem.

5 Induction motor vector control using LMABC

The three standard phases of ABC_{algo} namely, $EB_p hase$, $OB_p hase$, and $SB_p hase$ are carried out to generate a new origination. These operations are executed until pre-specified number of generation is attained or the needed accuracy is achieved. In this research the optimisation process is accomplished using MATLAB/SIMULINK parameters as follows: There are five optimal control gain variables, populous size is 20, the initial range of variables for K_1, K_2 are [200–600], for K_3, [0–20], and K_4, K_5 are [500–5,000].

The fitness functions are following:

- To improve the efficiency of the motor, the fitness function will be motor input power.

$$\begin{aligned} f_2 &= max(p_{in}) \\ &= max(I_{sa} \times u_a + I_{sb} \times u_b) \end{aligned} \tag{29}$$

- To obtain optimum control gains, the fitness function will be:

$$f_1 = \int [(f_{ref} - f_r)^2 + (\omega_{ref} - \omega_r)^2] dt \tag{30}$$

Such that f_{ref} is the reference flux and f_r is the motor rotor flux, ω_{ref} is the reference speed, and ω_r is the motor speed. The parameter of flux reference will be selected in the extent from [0.1, 2]. The proposed LMABC algorithm is applied on vector control of IM in following manner:

1. Step 1 (initialisation): this is the initialisation step, as there are five variables in this optimisation problem $(K_1–K_5)$, that are initialised within the search space in the following range: K_1, K_2 [200 500], K_3 [0 20], K_4, and K_5 [500 5,000]. For efficiency improvement of the motor and optimal control gains the fitness function are defined as equations (29) and (30) respectively. The F_{Source} in the search region are initialised using equation (21) and the FV of each F_{Source} is calculated accordingly.
2. Step 2 (EB_{phase}): During this step, the position of the each F_{Source} is modified using equation (22). The FV for the newly generated solution is obtained. The GSM is applied to select the solution for the next generation. If the newly generated solution is having better FV then it is selected for the next generation and the old solution is discarded.
3. Step 3 (OB_{phase}): during this phase the position of a solution is modified as per the equation (22) and the solutions are selected as per equation (23). In this step GSM is applied again between amid the newly generated solution and the old solution. The solution that is having higher FV is selected for the next generation.
4. Step 4 (SB_{phase}): if a particular solution is not updating its position upto a predefined threshold limit then that particular F_{Source} is reinitialised in the search space using equation (21) and the fitness of the newly generated solution is evaluated.
5. Step 5 (landmark phase).

5.1 Result analysis and discussions

In this paper, the ABC_{algo} and its proposed variant LMABC are operated to access the optimal gains of the FOC of the motor using equation (30). The optimal flux set point is obtained with one variable (r_1) which is the flux standard, one constant (c_1) to be selected as $(c_1) = 0.12$, $\omega = 0.9$, $n = 50$, maximum number of bees is 30, and the variable boundary are in range [0.1 to 2]. The objectives of simulation are as follows:

- to find optimum flux reference value which has the minimum input power using behaviour of the motor without any control during the speed and flux manual changes
- to find optimum flux reference value which has the minimum input power using the optimum flux reference implicitly by ABC_{algo} and LMABC
- to select optimum gains for the controller of FOC as shown in equations (17) and (18).

The simulation is carried out with standard speed 100 r.p.m., the speed is raised from zero to 100 seconds. Further, it is raised from 100 seconds time to 200 seconds. The standard flux is augmented at a pace from 0.15 weber to 0.45 weber. The motor input power is starting with high value then decreasing with flux standard incrementing to reach its minimum value then increasing. The excitation due to I_d and I_q currents outcomes designates that the optimum reference flux is 0.24 weber which has the minimum motor input power. Further, using ABC_{algo} and LMABC, there will be just one variable in the propound flux fitness function presented in equation (29), this value resembles the optimal flux standard value. The outcomes depict that by using ABC_{algo} and LMABC the flux standard will be $f_{ref} = 0.2429$ weber and $f_{ref} = 0.2409$ weber respectively. So the LMABC standard value is quite near to the target optimum standard value that was deduced without control. The input power using LMABC attains superior motor efficiency than ABC_{algo} and other cutting edge strategies with similar previous parameters and speed standards.

While selecting the optimal values of the controller gains K_1, K_2, K_3, K_4, and K_5 by using ABC_{algo} and LMABC with the similar motor parameters and with the similar speed standard using the gains fitness function as in equation (30). The optimum gains using GA are $K_1 = 450.5$, $K_2 = 510$, $K_3 = 9.8$, $K_4 = 11$, and $K_5 = 960$, while their value by using PSO are $K_1 = 231.9$, $K_2 = 355.5$, $K_3 = 20.6$, $K_4 = 534.5$, and $K_5 = 737.5$. Further, the optimum gains using ABC are $K_1 = 221.7$, $K_2 = 346.5$, $K_3 = 19.3$, $K_4 = 519.5$, and $K_5 = 729.5$, while their value by using LMABC are $K_1 = 219.6$, $K_2 = 326.4$, $K_3 = 18.9$, $K_4 = 517.1$, and $K_5 = 703.5$

6 Conclusions

This work introduces an efficient variant of artificial bee colony (ABC_{algo}) algorithm titled as landmark ABC (LMABC) algorithm. The propound variant is relied upon landmark operator. The LMABC enhances exploration and exploitation as well as maintains optimum balance between these two. Further, the ABC and its proposed variant are applied for vector control of induction motor. On comparing with the other existing cutting edge methods available writings, it is found that LMABC is a better choice for induction motor (IM) vector control.

References

Abu-Mouti, F.S. and El-Hawary, M.E. (2012) 'Overview of artificial bee colony (ABC) algorithm and its applications', *2012 IEEE International Systems Conference (SysCon)*, pp.1–6, IEEE.

Akay, B. and Karaboga, D. (2012) 'A modified artificial bee colony algorithm for real-parameter optimization', *Information Sciences*, Vol. 192, pp.120–142, doi: 10.1016/j.ins.2010.07.015.

Banharnsakun, A., Achalakul, T. and Sirinaovakul, B. (2011) 'The best-so-far selection in artificial bee colony algorithm', *Applied Soft Computing*, Vol. 11, No. 2, pp.2888–2901.

Bansal, J.C., Sharma, H., Arya, K. and Nagar, A. (2013) 'Memetic search in artificial bee colony algorithm', *Soft Computing*, Vol. 17, No. 10, pp.1911–1928.

Bojoi, R., Lazzari, M., Profumo, F. and Tenconi, A. (2003) 'Digital field-oriented control for dual three-phase induction motor drives', *IEEE Transactions on Industry Applications*, Vol. 39, No. 3, pp.752–760.

Clerc, M. and Kennedy, J. (2011) *Standard PSO 2011*, Particle Swarm Central Site [online] http://www.particleswarm.info (accessed 3 January 2019).

Douiri, M.R., Cherkaoui, M. and Essadki, A. (2012) 'Genetic algorithms based fuzzy speed controllers for indirect field oriented control of induction motor drive', *International Journal of Circuits, Systems and Signal Processing*, Vol. 6, No. 1, pp.21–28.

Duan, H. and Qiao, P. (2014) 'Pigeon-inspired optimization: a new swarm intelligence optimizer for air robot path planning', *International Journal of Intelligent Computing and Cybernetics*, Vol. 7, No. 1, pp.24–37.

Ebrahim, E.A. (2014) 'Artificial bee colony-based design of optimal on-line self-tuning PID-controller fed AC drives', *Int. J. Eng. Res.*, Vol. 3, No. 12, pp.807–811.

Gabriel, R., Leonhard, W. and Nordby, C.J. (1980) 'Field-oriented control of a standard AC motor using microprocessors', *IEEE Transactions on Industry Applications*, Vol. IA-16, No. 2, pp.186–192, doi: 10.1109/tia.1980.4503770.

Gastli, A. and Matsui, N. (1992) 'Stator flux controlled v/f PWM inverter with identification of IM parameters (induction motors)', *IEEE Transactions on Industrial Electronics*, Vol. 39, No. 4, pp.334–340.

Ho, E.Y. and Sen, P.C. (1988) 'Decoupling control of induction motor drives', *IEEE Transactions on Industrial Electronics*, Vol. 35, No. 2, pp.253–262.

Karaboga, D. (2005) *An Idea Based on Honey Bee Swarm for Numerical Optimization*, Techn. Rep. TR06, Erciyes Univ. Press, Erciyes.

Karaboga, D. and Akay, B. (2009) 'A comparative study of artificial bee colony algorithm', *Applied Mathematics and Computation*, Vol. 214, No. 1, pp.108–132.

Karaboga, D. and Basturk, B. (2007) 'A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm', *Journal of Global Optimization*, Vol. 39, No. 3, pp.459–471.

Kennedy, J. and Eberhart, R. (1995) 'Particle swarm optimization', *IEEE International Conference on Neural Networks, Proceedings*, Vol. 4, pp.1942–1948, IEEE.

Kim, D.H. (2007) 'GA-PSO based vector control of indirect three phase induction motor', *Applied Soft Computing*, Vol. 7, No. 2, pp.601–611.

Kirschen, D.S., Novotny, D.W. and Lipo, T.A. (1985) 'On-line efficiency optimization of a variable frequency induction motor drive', *IEEE Transactions on Industry Applications*, No. 3, pp.610–616.

Liu, T-H., Fu, J-R. and Lipo, T.A. (1993) 'A strategy for improving reliability of field-oriented controlled induction motor drives', *IEEE Transactions on Industry Applications*, Vol. 29, No. 5, pp.910–918.

Mann, H. and Whitney, D. (1947) 'On a test of whether one of two random variables is stochastically larger than the other', *The Annals of Mathematical Statistics*, Vol. 18, No. 1, pp.50–60.

Masiala, M., Vafakhah, B., Salmon, J. and Knight, A.M. (2008) 'Fuzzy self-tuning speed control of an indirect field-oriented control induction motor drive', *IEEE Transactions on Industry Applications*, Vol. 44, No. 6, pp.1732–1740.

Price, K. (1996) 'Differential evolution: a fast and simple numerical optimizer', *1996 Biennial Conference of the North American Fuzzy Information Processing Society, NAFIPS*, pp.524–527, IEEE.

Raj, C.T., Srivastava, S. and Agarwal, P. (2009) 'Energy efficient control of three-phase induction motor – a review', *International Journal of Computer and Electrical Engineering*, Vol. 1, No. 1, p.61.

Sajedi, S., Khalifeh, F., Khalifeh, Z. and Karimi, T. (2011) 'Application of particle swarm optimization and genetic algorithm methods for vector control of induction motor', *Australian Journal of Basic and Applied Sciences*, Vol. 5, No. 12, pp.1697–1706.

Shao, S., Abdi, E., Barati, F. and McMahon, R. (2009) 'Stator-flux-oriented vector control for brushless doubly fed induction generator', *IEEE Transactions on Industrial Electronics*, Vol. 56, No. 10, pp.4220–4228.

Sharma, H., Bansal, J.C., Arya, K. and Yang, X-S. (2015) 'Lévy flight artificial bee colony algorithm', *International Journal of Systems Science*, Vol. 3, pp.1–19, ahead-of-print.

Storn, R. and Price, K. (1995) *Differential Evolution – A Simple and Efficient Adaptive Scheme for Global Optimization over Continuous Spaces*, TR, International Computer Science Institute.

Wai, R-J. and Chang, H-H. (2004) 'Backstepping wavelet neural network control for indirect field-oriented induction motor drive', *IEEE Transactions on Neural Networks*, Vol. 15, No. 2, pp.367–382.

Williamson, D.F., Parker, R.A. and Kendrick, J.S. (1989) 'The box plot: a simple visual method to interpret data', *Annals of Internal Medicine*, Vol. 110, No. 11, pp.916–921.

Zhu, G. and Kwong, S. (2010) 'Gbest-guided artificial bee colony algorithm for numerical function optimization', *Applied Mathematics and Computation*, Vol. 217, No. 7, pp.3166–3173.