

Loughborough University
Institutional Repository

*Coevolutionary particle
swarm optimization using
AIS and its application in
multiparameter estimation
of PMSM*

This item was submitted to Loughborough University's Institutional Repository by the/an author.

Citation: LIU, Z-H. ...et al., 2013. Coevolutionary particle swarm optimization using AIS and its application in multiparameter estimation of PMSM. IEEE Transactions on Cybernetics, 43(6), pp. 1921-1935.

Additional Information:

- Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

Metadata Record: <https://dspace.lboro.ac.uk/2134/25039>

Version: Accepted for publication

Publisher: © IEEE

Please cite the published version.

Coevolutionary Particle Swarm Optimization Using AIS and Its Application in Multi-Parameter Estimation of PMSM

Zhao-Hua Liu, Jing Zhang, Shao-Wu Zhou, Xiao-Hua Li, and Kan Liu

Abstract—In this paper, a coevolutionary particle swarm optimization algorithm associating with the artificial immune principle is proposed. In the proposed algorithm, the whole population is divided into two kinds of subpopulations consisting of one elite subpopulation and several normal subpopulations. The best individual of each normal subpopulation will be memorized into the elite subpopulation, during the evolution process. A hybrid method, which creates new individuals by using three different operators, is presented to ensure the diversity of all the subpopulations. Furthermore, a simple adaptive wavelet learning operator is utilized for accelerating the convergence speed of the *pbest* particles. The improved immune clonal selection operator is employed for optimizing the elite subpopulation while the migration scheme is employed for the information exchange between elite subpopulation and normal subpopulations. The performance of the proposed algorithm is verified by testing on a suite of standard benchmark functions, which shows a faster convergence and global search ability. Its performance is further evaluated by its application to multi-parameter estimation of permanent magnet synchronous machines, which shows that its performance significantly outperforms existing PSOs. The proposed algorithm can estimate the machine dq-axis inductances, stator winding resistance and rotor flux linkage simultaneously.

Index Terms—artificial immune system(AIS), particle swarm optimization, Coevolutionaion, permanent magnet synchronous machines(PMSM), parameter estimation, migration scheme, elite population, global search

I. INTRODUCTION

Particle swarm optimization (PSO) was firstly presented by Kennedy and Eberhart in 1995[1]-[2], which imitates the behavior of swarms such as birds flocking and fish schooling. It has been greatly developed in recent years and is widely employed for solving problems in different areas such as neural networks [3], intelligent control [4][5], real optimization problems [6] and system identification [7]-[9]. However, since the basic PSO is based on iterative computation, which is similar to other evolutionary algorithms, it is report in literature

that the basic PSO is easy to lose diversity at the later stage of evolution [10]-[11] and suffer from trapping in local optima. Further, as detailed in [11], although each particle of basic PSO moves in a random direction, it has a potential trend of clustering together and may lose its diversity in the later stage of evolution computation and suffer from a premature convergence problem. Therefore, the recent literatures about PSO mainly focus on how to simultaneously accelerate the convergence speed and avoid the loss of diversity of the population, which can be reviewed as follows:

In some literatures, it was proposed to improve the algorithm performance by using a hybrid of PSO and other evolutionary algorithms (EA). It shows that these hybrid PSOs have the ability to jump out of the local optima thanks to the added operators, for example, the mutation operator. For instance, Angeline [12] firstly proposed a hybrid PSO method in which the standard selection mechanism of evolutionary computation is employed for improving the performance of PSO. Ahmed et al. [10] proposed to introduce the genetic mutation mechanism into the particle swarm optimization, which is with a constant mutating space. Juang et al. [11] proposed a hybrid PSO algorithm associating with the genetic algorithm (GA), in which the classical GA operators such as crossover, mutation, and reproduction improved the diversity of PSO significantly. However, since the factors of utilized operators such as the mutation rate, crossover rate and reproduction rate are usually set to be constants in these methods, their performances are still sensitive to the variation of operator setups. In order to overcome this flaw, Ling et al. [13] proposed a hybrid PSO which included a wavelet function based mutation operation. Compared with basic PSO, the hybrid PSO in [13] can dynamically vary with the value of wavelet function in each generation of optimization and its ability in exploring the solution space is enhanced thanks to the dynamic mutating space. However, since the mutation amplitude of used wavelet function is usually very small, the convergence speed of the hybrid PSO in [13] is too low and its accuracy cannot be ensured.

In some literatures, it was proposed to change the particle behaviors to improve the performance of PSO. For instance, in [14] and [15], Shi et al. proposed to adjust the convergence speed of particles to improve their dynamic performance for the first time, where a linear function [14] and a fuzzy method [15] were utilized to change the convergence factors of PSO, respectively. Zhan et al. [16] proposed an adaptive PSO which can accelerate the convergence speed and jump out of the local

1 Manuscript received September 1,2011; revised March 12,2012; July 11,2012and November 29,2012; accepted December 5,2012.This work was supported in part by the National Natural Science Foundation of China under Grant (61174140, 61203309), the Research Fund for the Doctoral Program of Higher Education of China under Grant 20110161110035 and Key Projects in the National Science & Technology Pillar Program under Grant(2012BAH09B00, 2012BAH09B02).

Z.-H. Liu, S.-W. Zhou and X.-H.Li are with School of Information and Electrical Engineering, Hunan University of Science and Technology, Xiangtan, 411201, China (email: zhaohualiu2009@hotmail.com, swzhou@hnust.edu.cn, lixiaohua_0227@163.com.).

J. Zhang is with College of Electrical and Information Engineering, Hunan University ,Chang Sha ,410082, China(email: zhangj@hnu.edu.cn.).

K.Liu is with the Department of Electronic and Electrical Engineering, University of Sheffield, UK(email: lkan@live.cn)

optima by using a proposed parameter adaptive regulation scheme and an elitist learning strategy. The utilized adaptive regulation scheme in [16] still needs expert experiences to judge the evolutionary state and there exists a potential risk that these experiences may not be appropriate in solving other problems and will influence on the population diversity. In addition, different from changing the convergence speed of particles, Liang et al. [17] proposed a comprehensive learning PSO, in which all the flying particles are randomly updated by each other in each step of evolution. Although the method in [17] is superior in keeping diversity, it does not take into account how to jump out of the local optima and will cause the algorithm with low convergence speed. Similar problem also exists in other PSO using varying *pbests* [18]. In addition, some researchers proposed to design different types of topologies to improve the effectiveness of the PSO. For instance, in [19] and [20], Kennedy and Mendes proposed two new PSOs for solving multimodal problems, which are with ring topology and von Neumann topology (VPSO), respectively.

In some other literatures, it was proposed to generate multi-swarm to ensure the diversity of PSO. For instance, since the algorithm based on multi-population or co-evolutionary scheme is usually of better performance in maintaining the diversity compared with the algorithm based on a single population because there existing multiple global optimums in all the subpopulations, in other words, each subpopulation has its own global optimum. It has been reported in some literatures that it is applicable to use multi-population or co-evolutionary scheme to improve the diversity of the basic PSO [21]-[24]. For instance, Bergh et al. [21] proposed to use multiple swarms to optimize different components of the solution vector, which showed better performance than basic PSO. However, there is no information interaction between these isolated swarms, which may influence on the convergence speed of the algorithm. In [22], a hybrid method is proposed, in which a truncated Gaussian distribution function is utilized to accelerate the convergence speed and a co-evolutionary scheme for PSO is utilized to ensure the diversity of population. In [23], a PSO with dynamic multi-swarm was proposed, in which the number of swarms was adaptively adjusted throughout the search process. Further, Niu et al. [24] proposed to use the concept of master-slave mode to improve the PSO performance, where the population consists of one master swarm and several slave swarms. The slave swarms will explore the search space independently to maintain the diversity of particles while the master swarm evolves based on its own knowledge. However, it does not take into account how to escape from the local optimum and the convergence speed of each swarm may become slow at the later stage of evolution.

All in all, existing literatures mainly focus on using the foregoing three methods to improve the performance of classical PSO. However, from the analysis, there exist merit and flaw simultaneously in these methods. Thus, if the merits of these methods can be utilized to improve each other, it is possible to design an algorithm with better performance. In this paper, a novel co-evolutionary PSO called Immune Co-evolution Particle Swarm Optimization algorithm (ICPSO) is proposed.

Compared with classical PSO, the proposed ICPSO is associated with the co-evolution theory and artificial immune system (AIS) theory [25]-[26]. The whole population of ICPSO is divided into two kinds of subpopulations including one elite subpopulation and several normal subpopulations. In each generation of the algorithm, the best individual of each normal subpopulation will be memorized into the elite subpopulation. A hybrid method, creates new individuals by using operators such as elitist reservation, immune network and Cauchy mutation, is proposed to ensure the diversity of all the subpopulations. For accelerating the convergence speed of *pbests* in normal subpopulations, a simple wavelet learning operator is employed to adjust the convergence factor. In addition, since the artificial immune system (AIS) theory has a powerful intelligent information processing capability, it is verified in literatures that some operators based on AIS can be used for improving other algorithms [27]-[29]. Therefore, the immune clonal selection operator is employed for optimizing the elite subpopulation and the migration scheme is employed for the information exchange between elite and normal subpopulations. Compared with other hybrid PSOs, the performance of proposed ICPSO is verified by solving some standard benchmark functions, high dimension functions, and shift functions, which show better performance in convergence, global search and solution stability. Its performance is further verified by its application to multi-parameter estimation of permanent magnet synchronous machines.

II. IMMUNE CO-EVOLUTION PARTICLE SWARM OPTIMIZATION ALGORITHM FRAMEWORK

In basic PSO, a swarm of particles are represented as potential solutions, and each particle i consists of two vectors, which are the velocity vector $V_i = \{V_{i1}, V_{i2}, \dots, V_{id}\}$ and the position vector $X_i = \{X_{i1}, X_{i2}, \dots, X_{id}\}$. Assuming that $Pbest_{id}$ represents the best position found by the i -th particle so far and $Gbest_d$ is the best position in the entire group, the velocity and position of particle i in dimension d are updated during the search process by the following functions.

$$V_{id}(t+1) = wV_{id} + c_1 * rand_1() (Pbest_{id}(t) - X_{id}(t)) + c_2 * rand_2() (Gbest_d(t) - X_{id}(t)) \quad (1)$$

$$X_{id}(t+1) = X_{id}(t) + V_{id}(t+1) \quad (2)$$

Where w is the inertia weight decreasing linearly from 0.9 to 0.4; c_1 and c_2 are the acceleration coefficients; $rand_1$ and $rand_2$ are two uniformly distributed random numbers generated in the interval $[0,1]$ and $v_{id} \in [v_{min}, v_{max}]$, respectively, where v_{min} and v_{max} are the designated minimum velocity and maximum velocity, respectively. As can be seen from (1) and (2), if the $Pbest_{id}$ and $Gbest_d$ are close to X_{id} , and also w is less than one, the diversity will be low during the process of evolution, which may halt the cycle of evolutionary computing. Thus, an improved PSO, which is called immune co-evolution particle swarm optimization, is proposed in this section to improve the

performance of basic PSO and shown as follows.

In natural system, the evolution levels of different biomes are usually different from each other and there usually exist information and energy interactions among these biomes. In addition, there also exists stratification phenomenon in biomes. These activities and phenomenon, namely co-evolution [30], can be employed for improving the performance of PSO. As can be seen from Fig.1, the proposed ICPSO consists of one memorized elite subpopulation and several normal subpopulations and has taken into account the co-evolution which can inspire the diversity of the whole population. The pseudo code of ICPSO algorithm is shown in Fig.3.

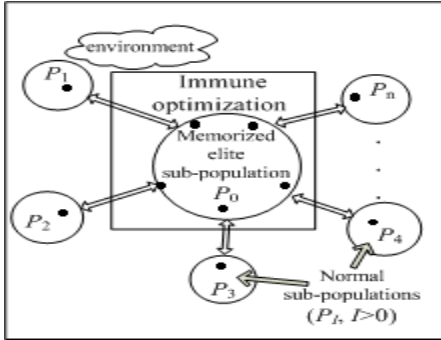


Fig. 1. Elite and normal subpopulations of ICPSO

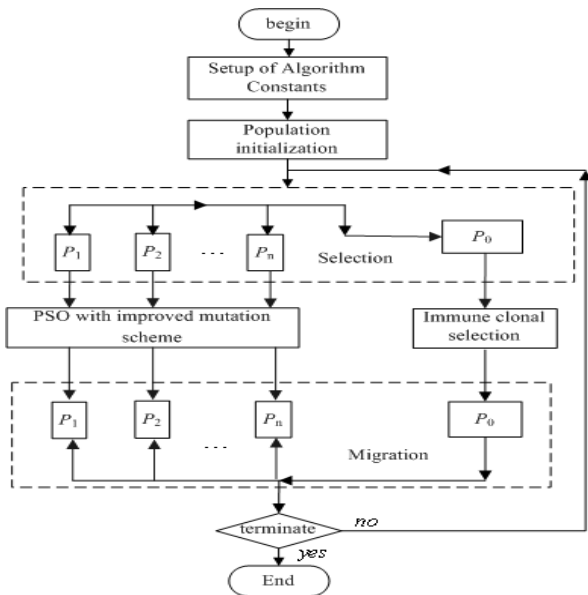


Fig. 2. Immune co-evolution particle swarm optimization algorithm flow

In Fig.1, P_I ($I > 0$) represents the I -th normal sub-population of the whole population and P_0 represents the memorized elite subpopulation. Furthermore, P_{IJ} represents the J -th dominant particle of the I -th subpopulation and the memorized P_0 is equivalent to the dominant antibody population in immune system. The individuals in P_0 are dominant particles selected from all the normal subpopulations. The population size of the elite subpopulation is fixed to three multiply normal subpopulations, named best individuals and two secondary individuals are selected from normal subpopulations. The best individuals and only one of each normal subpopulation P_i with highest fitness are memorized into P_0 and will replace the

existing elite individuals with low fitness in each generation. Part of the secondary individuals (two numbers) of each normal subpopulation P_i with high fitness are randomly selected and memorized in P_0 in each generation.

Fig.2 shows the schematic diagram of proposed ICPSO and its each step of evolution is detailed as follows.

1) Setup of algorithm constants.

Constants such as c_1, c_2 (each subpopulation have the same c_1 and c_2), P_i , mutation and clonal selection rates are initialized in this step.

2) Population initialization.

As can be seen from Fig.1, the whole population is divided into two groups, which are named elite and normal subpopulations. The initialization of population is through randomly initialized.

3) Immune clonal selection.

The memorized elite subpopulation will be optimized by the algorithm of immune clonal selection [25] and the obtained global best individuals will migrate to these normal subpopulations to replace their worst individuals. During this step, all the individuals in elite subpopulation will be optimized by using hyper mutation and the whole elite subpopulation will be updated by the optimized results.

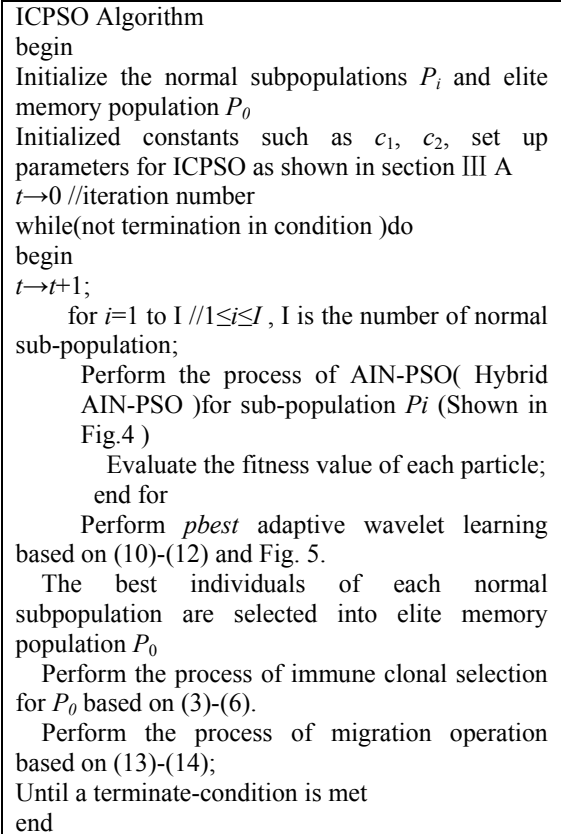


Fig. 3. Pseudo code for ICPSO

The whole process is detailed as follows.

a. Clone.

In each generation, the particles are regarded as antibodies in the elite subpopulation, whose clonal scale is proportional to its affinity. The clonal scale (N_c) of the whole elite subpopulation is

shown in (3).

$$N_c = \sum_i^N \text{round}(\frac{\beta N}{i} + b_1) \quad (3)$$

Where N is the population size and clonal scale coefficient $\beta \in (0,1)$, β is set to be 0.8. In addition, b_1 is setup to be larger than 1, which can keep the diversity of the whole population.

b. Hyper mutation.

Since the particles are regarded as antibodies in the elite subpopulation, the mutation operator can be expressed as follows.

$$P_{id}^{new} = P_{id} + [(rd > P_m)] * \eta * P_{id} * U(0,1) - [(rd \leq P_m)] * \eta * P_{id} * U(0,1) \quad (4)$$

$$[rd > P_m] = \begin{cases} 1 & \text{if } rd > P_m \\ 0 & \text{else} \end{cases} \quad (5)$$

Where rd is a randomly generated number and P_m is set to be 0.5. η is calculated by (6).

$$\eta(t) = 1 - r^{[1-(t/T)]^b} \quad (6)$$

Where T is the maximum evolution generation and t is the current generation number. Nonlinear variation coefficient b is a formal parameter and is set to be 2. r and U are randomly generated during the initialization and in each generation, respectively, which are both uniformly distributed in $(0,1)$. As can be seen from (6), $\eta(t)$ will decrease to 0 when $t=T$.

c. Clonal selection.

After the hyper mutation, the best individuals will be selected according to the affinity, and used for updating the elite subpopulation, which is named clonal selection [25].

d. Hybrid AIN-PSO.

The technologies of artificial immune network (AIN)[26] and PSO will be simultaneously employed to optimize the normal subpopulations, which are schematically shown in Fig.4. Prior to the optimization, the normal subpopulation will be ranked into three levels according to their fitness; the population is divided into three parts: the elite individuals (half of the population), the better individuals (forty percent of the population) and the worst individuals (one-tenth of the population). The new groups are generated through the following hybrid strategies.

In addition, an artificial immune network (AIN) based PSO will be employed for generating new middle-level individuals. From Fig.4, during the evolution, the middle-level individuals will be updated by comparing with the selected elite individuals. The AIN can explain the immune system dynamics phenomenon, which maintains interactions between not only an antibody and an antigen, but also antibodies themselves. Inspired by the idiotypic network theory, the principium of the AIN is as follows:

Assuming that C , C^* , α and C_{Ag} are the vector of AIN cells, the updated vector of AIN cells, AIN mutation rate and the vector of antigens, respectively, the AIN cells can be updated by (7).

$$C^* = C - \alpha \cdot (C - C_{Ag}) \quad (7)$$

(7) is the equation of mutation for a normal AIN. Thus, similar to (7), the equation of mutation for AIN based PSO can be derived as follows.

$$X_{id}(t+1) = X_{id}(t) - r_1 \cdot (X_{id}(t) - X_{kd}(t)) + r_2 \cdot \text{cauchy} \cdot X_{id}(t) \quad (8)$$

Where, $i \neq k$, r_1 , r_2 are random numbers in $(0, 1)$. (8) is used for the mutation operation of middle-level individuals, in which X_{id} is the i -th particle of P_i and X_{kd} is the k -th particle of the selected elite-level individuals. The term $r_1 \cdot (X_{id}(t) - X_{kd}(t))$ can gather the particles of middle-level individuals around the particles of the selected elite individuals. The term $r_2 \cdot \text{cauchy} \cdot X_{id}(t)$ can activate the mutation of the middle-level individuals, so offspring can inherit excellent variation information from parents. *cauchy* is the density function, which can be expressed as follows.

$$\text{cauchy}_t(x) = \frac{1}{\pi} \frac{t}{t^2 + x^2}, -\infty < x < \infty, t > 0 \quad (9)$$

Comparing with the traditional random mutation, Cauchy mutation performs better because of its higher probability of making longer jumps [32].

The lowest-level individuals which belong to the worst individuals will be reinitialized, and then the dynamic performance of particle is enhanced, and also can avoid premature convergence for swarms. Finally, the updated three-level individuals will be memorized in sequence and used for the evolution of next generation. From the description above, it is obvious that the proposed AIN-PSO operation can keep the diversity of the whole population and accelerate the convergence speed simultaneously.

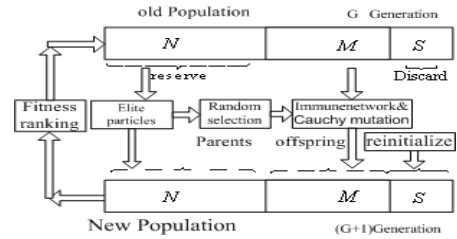


Fig. 4. Hybrid AIN-PSO in sub-population

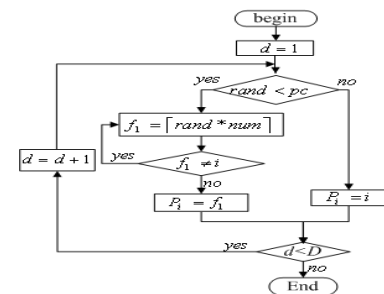


Fig. 5. P_i selection flow chart

e. $pbest$ learning with adaptive wavelet

As detailed in Fig.4, the elite individuals with high fitness will be optimized by a PSO algorithm with improved $pbest$ learning method, which is shown as follows.

As detailed in the first section, the wavelet function can nonlinearly adjust the mutation speed of PSO by following the increasing of generation number. Thus, similarly, the wavelet function is employed for adjusting the learning speed of $pbest$. The Choose Morlet wavelet function is employed and shown as

follows [13], [31].

$$\sigma = \frac{1}{\sqrt{a}} e^{-\left(\frac{\varphi}{a}\right)^2 / 2} \cos\left(5\left(\frac{\varphi}{a}\right)\right) \quad (10)$$

Where $\varphi \in [-2.5a, 2.5a]$ and will be randomly generated during the evolution. a will linearly increase with the generation number and can be expressed as follows.

$$a = a_{min} + (a_{max} - a_{min}) \frac{t}{T} \quad (11)$$

Where a_{max} and a_{min} are the upper and lower boundaries of a , fixed ($a_{max}=1000$, $a_{min}=10$). By using (10), all the $pbests$ will be updated by (12) in each generation, which is shown as follows.

$$Pbest_{id(j)} = Pbest_{id(j)} + \sigma \cdot (gbest_{P_i(d)} - Pbest_{id(j)}) \quad (12)$$

Where $Pbest_{id(j)}$ represents the best position found by the i -th particle so far in subpopulations P_j , where $P_i(d)$ defines which subpopulation's $gbest$ particle corresponding dimension, where $gbest_{P_i(d)}$ represents the dimension of the $gbest$ of P_j and the number of i is randomly selected in each generation, which is schematically shown in Fig.4. By using (12), perturbations related to different $gbest_{P_i(d)}$ will be employed for adjusting all the $pbests$ of P_i . Each $pbest$ may learn the global best particles that come from different subpopulations randomly through the adaptive wavelet learning mechanism. So, the $pbest$ learning scheme will maintain population diversity and help the particles jumping out of the potential local optima, simultaneously, can improve the convergence speed of PSO.

f. Migration.

The operation of migration is not conducted in every generation and it will be activated by the migration signal sig . In the proposed ICPSO, the migration operation will be executed if $sig=1$. Part of the antibodies (ten percent) with high affinity in the elite subpopulation will be copied and migrate to P_i to replace the worst antibodies if $sig=1$, which can improve the average fitness of the normal subpopulations. The pseudo code of migration procedures is shown in Fig.6.

The used migration operation can further accelerate the convergence rate and the interactions between the elite and normal subpopulations can make a balance between searching roughly and accurately.

```

Migration procedures
For each normal sub-population  $p_i$ 
  if  $sig==1$ 
    A new best individual in the  $p_0$ 
    (elite population) with high fitness
    is selected randomly and migrated
    to  $p_i$  and replace the worst
    individual
  End if
End For

```

Fig. 6. Pseudo code for migration procedures

g. Termination or jump to 3).

The evolution process will be terminated if the generation (t) has reached the maximum generation (T).

III. BENCHMARK TESTS AND COMPARISONS

A. Benchmark Function

A set of standard benchmark test functions [27] are employed to validate the performance of ICPSO and for the comparison between ICPSO and existing hybrid PSOs. The employed benchmark functions are corresponding to different optimization problems and can be divided into two categories, which are unimodal functions (f_1 - f_4) and multimodal functions being with several local minima f_5 - f_6 . The expressions of these benchmark functions are depicted in Table I.

TABLE I
BENCHMARK TEST FUNCTIONS

| Test Function | Domain Range | Acceptance | Global Optimal |
|---|----------------------------|------------|----------------|
| $f_1(x) = \sum_{i=1}^{30} x_i^2$ | $-100 \leq x_i \leq 100$ | 0.001 | 0 |
| $f_2(x) = \sum_{i=1}^{30} (\sum_{j=1}^i x_j)^2$ | $-100 \leq x_j \leq 100$ | 0.001 | 0 |
| $f_3(x) = \sum_{i=1}^{30} (100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2)$ | $-10 \leq x_i \leq 10$ | 50 | 0 |
| $f_4(x) = \sum_{i=1}^{30} ix_i^4 + random(0, 1)$ | $-1.28 \leq x_i \leq 1.28$ | 0.01 | 0 |
| $f_5(x) = -20 \exp(-0.2 \sqrt{\frac{1}{30} \sum_{i=1}^{30} x_i}) - \exp(\frac{1}{30} \sum_{i=1}^{30} \cos(2\pi x_i)) + 20 + e$ | $32 \leq x_i \leq 32$ | 0.001 | 0 |
| $f_6(x) = \frac{\pi}{30} \{10 \sin(\pi y_1) + \sum_{i=1}^{30-1} (y_i - 1)^2 [1 + 10 \sin^2(\pi y_{i+1})] + (y_{30} - 1)^2\}$ + $\sum_{i=1}^{30} u(x_i, 10, 100, 4)$ | $-50 \leq x_i \leq 50$ | 0.0001 | 0 |
| where, $y_i = 1 + \frac{1}{4}(x_i + 1)$, $u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m, \dots, x_i > a \\ 0, \dots, -a \leq x_i \leq a \\ k(-x_i - a)^m, x_i < -a \end{cases}$ | | | |

All these functions are tested 30 times and with 30 dimensions. Existing PSOs such as hybrid PSO with mutation (HPSOM) [10], hybrid PSO with genetic algorithm (HGAPSO) [11], hybrid PSO with Wavelet Mutation (HPSOWM) [13], Adaptive Comprehensive Learning Particle Swarm Optimization (A-CLPSO) [18], Adaptive Particle Swarm Optimization (APSO) [16], and Comprehensive Learning PSO (CLPSO), which aim at achieving better performance in multimodal functions optimization [17], are compared with the proposed ICPSO.

The value of “acceptance” in Table I is defined to judge whether a solution predefined found by the PSO would be acceptable or not in [16]. Set up parameters for ICPSO as follows: The inertia weight w in equation (1) is set to be $w \in$

$[0.90, 0.4]$ and decreases linearly [16], the acceleration coefficients $c_1=c_2$ are both set to be 1.49445[16]. In equation (3), β is set to be 0.8. b_l is setup to be 5. in equation (4) P_m is set to be 0.5, in equation (6)b is set to be 2. Fixed ($a_{max}=1000, a_{min}=10$) in equation (11). the mutation rates in clonal selection is set to be 0.8.

All algorithms use the same number of 3000 FEs for each test function, All the PSO algorithms using the same population size of 50 in subpopulation. All experiments are implemented on the same PC computer with AMD Athlon (tm) II X2 250 four processors. For reducing statistical errors, each function is independently simulated 30 times and the mean results and standard deviation are used for comparison.

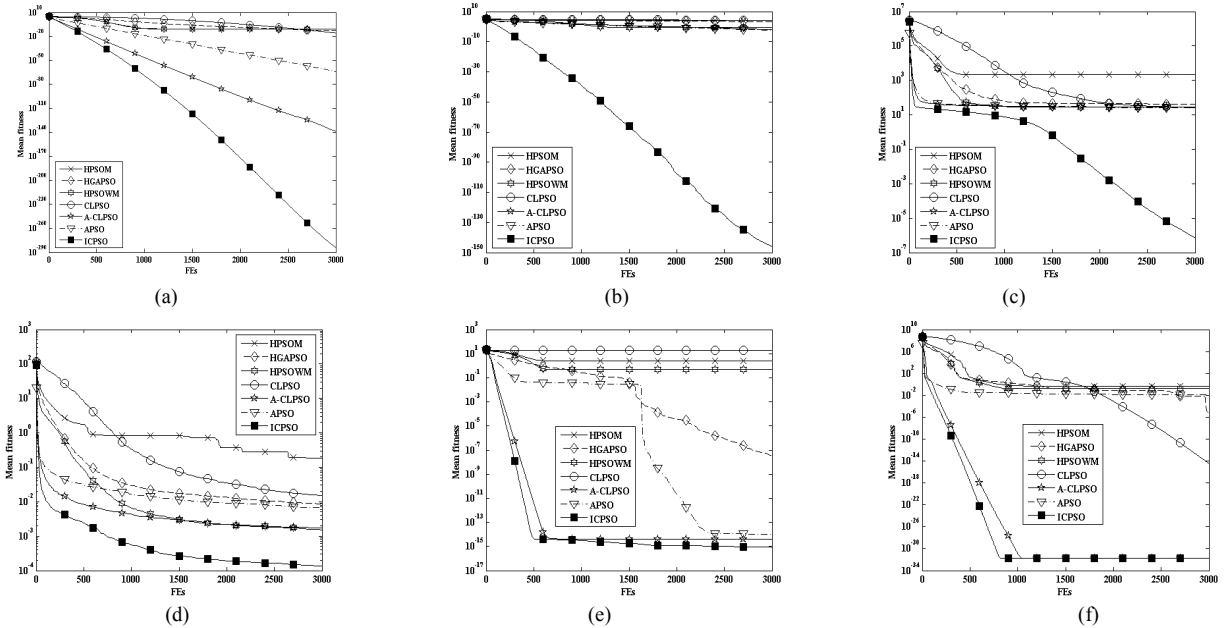


Fig. 7. Convergence performance of the seven different PSOs on the 6 test functions. (a) f_1 . (b) f_2 . (c) f_3 . (d) f_4 . (e) f_5 . (f) f_6 .

TABLE II
SEARCH RESULT COMPARISONS AMONG SEVENEIGHT PSOS ON 10 TEST FUNCTIONS

| Function | HPSOM | HGAPSO | HPSOWM | CLPSO | A-CLPSO | APSO | ICPSO |
|----------|---------|------------------------|------------------------|------------------------|------------------------|-------------------------|------------------------|
| f_1 | Mean | 9.05×10^{-12} | 2.43×10^{-13} | 8.50×10^{-12} | 1.63×10^{-16} | 2.58×10^{-139} | 1.15×10^{-64} |
| | Std.Dev | 2.70×10^{-11} | 1.10×10^{-12} | 3.63×10^{-11} | 8.33×10^{-17} | 1.12×10^{-138} | 4.48×10^{-64} |
| | t-test | 2.3701 | 1.5621 | 1.6558 | 13.8365 | 1.6289 | N/A |
| f_2 | Mean | 10149.8 | 1866.66 | 0.27 | 10447.6 | 1.13×10^{-2} | 4.15×10^{-3} |
| | Std.Dev | 6761.24 | 2874.01 | 0.31 | 1643.58 | 8.10×10^{-3} | 3.23×10^{-3} |
| | t-test | 10.6149 | 4.5926 | 6.1587 | 44.9480 | 9.8646 | 9.0851 |
| f_3 | Mean | 2072.59 | 40.14 | 27.67 | 26.70 | 26.88 | 24.81 |
| | Std.Dev | 3978.51 | 34.94 | 17.28 | 4.89 | 15.81 | 21.88 |
| | t-test | 3.6836 | 8.1234 | 11.3227 | 38.6089 | 12.0222 | 8.0180 |
| f_4 | Mean | 0.188 | 8.65×10^{-3} | 1.74×10^{-3} | 1.48×10^{-2} | 1.51×10^{-3} | 6.63×10^{-3} |
| | Std.Dev | 0.67 | 2.93×10^{-3} | 7.80×10^{-4} | 2.46×10^{-3} | 5.55×10^{-4} | 2.46×10^{-3} |
| | t-test | 1.9827 | 20.5453 | 14.4932 | 42.1369 | 17.3753 | 18.6637 |
| f_5 | Mean | 2.63 | 3.61×10^{-8} | 0.48 | 18.45 | 4.14×10^{-15} | 1.07×10^{-14} |
| | Std.Dev | 4.89 | 4.06×10^{-8} | 0.65 | 2.73 | 0 | 1.85×10^{-15} |
| | t-test | 3.8030 | 6.2873 | 5.2217 | 47.7880 | 26.4566 | 34.0416 |
| f_6 | Mean | 0.447 | 7.10×10^{-6} | 0.187 | 3.56×10^{-15} | 1.57×10^{-32} | 1.04×10^{-2} |
| | Std.Dev | 0.693 | 2.92×10^{-5} | 0.314 | 2.41×10^{-15} | 2.74×10^{-48} | 3.11×10^{-2} |
| | t-test | 4.5610 | 1.7193 | 4.2111 | 10.4452 | 0 | 2.3646 |

B. Accuracy Comparison of Different PSOs

This section presents results gathered by allowing all of the methods tested to run for a fixed number of 3000 FEs. Table II shows the benchmark functions for comparison among the proposed ICPSO and six other PSOs such as HPSOM, HGAPSO, HPSOWM, CLPSO, A-CLPSO, and APSO. The comparison results are shown in Table II, which are in terms of mean fitness and standard deviation (Std. Dev) of the solutions obtained from the 30 independent runs of each algorithm. The Fig.7. graphically presents the comparison between these PSOs, which expressed in terms of convergence characteristics in solving 6 test functions.

In Table II and Fig.7, we can see that ICPSO has the best performance in solving most of these test functions, especially for unimodal problems (f_1 , f_2 , f_3 , and f_4). Furthermore, ICPSO also shows the highest accuracy in solving unimodal problems. In addition, Fig.7 and Table II also shows that ICPSO can be effective in optimizing complex multimodal functions such as f_5 . Overall, comparing with other hybrid PSOs, ICPSO has the best performance in solving the 6 test functions, especially for unimodal functions and multimodal functions.

The t -test [13] is a statistical method to evaluate the significant difference between two algorithms. The t -value will be positive if the first algorithm is better than the second, and it is negative if it is poorer. When the t -value is higher than 1.645 (the value of the degrees of freedom $\xi = 49$), there is a significant difference between the two algorithms with a 95% confidence level. The t -values between the ICPSO and other hybrid PSOs are shown in Table II. We see that most t -values in this Table are higher than 1.645. Therefore, the performance of the ICPSO is significantly better than that of other PSOs with a 95% confidence level.

The reasons that the proposed ICPSO consists of one memorized elite subpopulation and several normal subpopulations and have consider the co-evolution which can inspire the diversity of the whole population. A hybrid method, which creates new individuals by using three

different operators, can ensure the diversity of all the subpopulations. Furthermore, a simple adaptive wavelet Learning operator is utilized for accelerating the convergence speed of the $pbest$ particles. The improved immune clonal selection operator is employed for optimizing the elite subpopulation while the migration scheme is employed for the information exchange between elite subpopulation and normal subpopulations. So the ICPSO shows better performance in convergence, global search as well as solution stability.

C. Comparison in Convergence Speed

The convergence speed of an optimization algorithm is also an important feature to prove its superiority over other algorithms. Table III shows that ICPSO generally has a much higher speed in terms of either the mean number of FEs or the mean cost of CPU time for searching an acceptable solution (list in Table I). The spending of CPU time is an important feature to describe the computation cost of an algorithm, as many existing hybrid PSOs have added extra operations that cost computational time. In solving real-world problems, the ‘‘FEs’’ time overwhelms the algorithm overhead. Hence, the mean number of FEs needed to reach acceptable accuracy would be as important as the CPU time [15]. Thus, the mean FEs are also explicitly presented and compared in Table III. For example, tests on f_1 show that the average numbers of FEs of 1096.5, 2050.3, 1149.0, 3766.4, 205.8, and 283.2 are required for every particle in the HPSOM, HGAPSO, HPSOWM, CLPSO, and A-CLPSO algorithms, respectively, for reaching an acceptable solution. However, every particle within the ICPSO only uses 89.87 FEs on average whereas its CPU time of 0.056s is also the shortest among the seven hybrid PSOs algorithms. In summary, the ICPSO uses the least CPU time and the smallest number of FEs to reach acceptable solutions on 6 different test functions, the convergence is faster.

TABLE III
CONVERGENCE SPEED AND ALGORITHM RELIABILITY COMPARISONS

| Function | | HPSOM | HGAPSO | HPSOWM | CLPSO | A CLPSO | APSO | ICPSO |
|----------|-----------|--------|--------|--------|--------|---------|--------|--------|
| f_1 | Mean FEs | 1096.5 | 2050.3 | 1149.0 | 3766.4 | 205.8 | 283.2 | 89.87 |
| | Time(sec) | 0.228 | 0.406 | 0.244 | 0.520 | 0.077 | 0.139 | 0.056 |
| | Ratio(%) | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 |
| f_2 | Mean FEs | 2738.0 | 6304.3 | 2623.3 | - | 5351.6 | 3248.5 | 240.13 |
| | Time(sec) | 1.281 | 2.986 | 1.228 | - | 2.960 | 2.315 | 0.218 |
| | Ratio(%) | 33.3 | 60.0 | 100.0 | 0 | 100.0 | 100.0 | 100.0 |
| f_3 | Mean FEs | 1411.8 | 2049.6 | 1253.7 | 4382.6 | 563.8 | 661.6 | 90.67 |
| | Time(sec) | 0.324 | 0.458 | 0.285 | 0.635 | 0.231 | 0.334 | 0.131 |
| | Ratio(%) | 73.3 | 93.3 | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 |
| f_4 | Mean FEs | 2387.5 | 3559.2 | 1809.4 | 5535.4 | 386.80 | 1737.1 | 178.93 |
| | Time(sec) | 0.521 | 0.657 | 0.399 | 0.822 | 0.134 | 0.852 | 0.116 |
| | Ratio(%) | 96.7 | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 |
| f_5 | Mean FEs | 1265.3 | 2572.1 | 1285.0 | - | 246 | 438.9 | 113.97 |
| | Time(sec) | 0.470 | 0.726 | 0.472 | - | 0.110 | 0.250 | 0.085 |
| | Ratio(%) | 96.7 | 100.0 | 96.7 | 0 | 100.0 | 100.0 | 100.0 |
| f_6 | Mean FEs | 1721.4 | 2135.4 | 1507.0 | 4570.6 | 207.6 | 1129.7 | 227.70 |
| | Time(sec) | 1.338 | 1.222 | 1.128 | 1.998 | 0.132 | 0.858 | 0.546 |
| | Ratio(%) | 56.7 | 100.0 | 66.7 | 100.0 | 100.0 | 100.0 | 100.0 |

D. Comparison in Reliability

Table III also reveals that ICPSO offers a generally highest percentage of trials (reaching acceptable solutions) and the highest reliability averaged over all the test functions. The ICPSO reaches the acceptable solutions with a successful ratio of 100% on all the six test functions. An interesting result is that all the PSO algorithms have most reliably found the minimum of f_1 , this problem may relatively be easy to solve with a 100% success rate. ICPSO offers the highest reliability of 100%. followed by APSO, A-CLPSO, CLPSO, HPSOWM, HGAPSO, and HPSOM.

E. Sensitivity of parameters and operators for ICPSO

What impacts do the three operations of population size, adaptive wavelet learning and immune clonal selection operator, have on the performance of the ICPSO? This section aims to answer these questions by further testing the APSO by some representative functions (out of 6) from each

group were tested. Functions f_2 and f_3 are typical unimodal functions and functions f_4, f_5 and f_6 are multimodal functions with many local minima.

Sub-population size means the number of sub-population in Table IV, the numbers 3, 4, 5, and 6 mean that there are 3, 4, 5, and 6 sub-populations in each testing, respectively. It can see from the Table IV above, the performance of ICPSO method improved with increasing sub-population size, however, the cost of time increases slightly with increasing sub-population size. The performances of ICPSO with sub-population size equal to 4 nearly similar to the performance of ICPSO with sub-population size equal to 5. But the computational cost with sub-population size equal to 5 is lower than the cost of time with sub-population number equal to 5. From the above analysis, we can see when the size of the sub-population equal to 4 is better for the performance of ICPSO. So, based on the results, we recommend using subpopulation size equal to four in this paper.

TABLE IV
EFFECTS OF THE POPULATION SIZE ON GLOBAL SEARCH QUALITY

| Sub-Population Size | | f_1 | f_2 | f_3 | f_4 | f_5 | f_6 |
|---------------------|-----------|-------------------------|-------------------------|-----------------------|-----------------------|------------------------|------------------------|
| 3 | mean | $4.26e^{-280}$ | 1.26×10^{-132} | 9.19×10^{-2} | 2.11×10^{-4} | 1.42×10^{-15} | 1.57×10^{-32} |
| | Std.dev | 0 | 6.78×10^{-132} | 0.274 | 1.45×10^{-4} | 1.50×10^{-15} | 2.74×10^{-48} |
| | Time(sec) | 9.15 | 14.2547 | 9.203 | 10.352 | 10.890 | 17.087 |
| 4 | mean | 1.02×10^{-284} | 8.61×10^{-147} | 7.53×10^{-7} | 1.34×10^{-4} | 8.25×10^{-16} | 1.57×10^{-32} |
| | Std.dev | 0 | 4.61×10^{-146} | 8.19×10^{-7} | 7.45×10^{-5} | 8.86×10^{-16} | 2.74×10^{-48} |
| | Time(sec) | 13.31 | 17.716 | 12.620 | 13.780 | 13.745 | 22.833 |
| 5 | mean | $6.34e^{-286}$ | 5.53×10^{-149} | 1.04×10^{-7} | 1.17×10^{-4} | 9.44×10^{-16} | 1.57×10^{-32} |
| | Std.dev | 0 | 2.98×10^{-148} | 1.08×10^{-7} | 7.38×10^{-5} | 1.07×10^{-15} | 2.74×10^{-48} |
| | Time(sec) | 15.31 | 23.630 | 15.404 | 17.524 | 17.018 | 28.555 |
| 6 | mean | 1.01×10^{-287} | 7.36×10^{-153} | $1.48e^{-8}$ | 7.44×10^{-5} | 1.18×10^{-15} | 1.57×10^{-32} |
| | Std.dev | 0 | 3.95×10^{-152} | $1.71e^{-8}$ | 6.29×10^{-5} | 1.32×10^{-15} | 2.73×10^{-48} |
| | Time(sec) | 18.32 | 25.89 | 18.8 | 20.66 | 21.31 | 35.06 |

TABLE V
EFFECTS OF THE CLONAL SCALE COEFFICIENT β ON GLOBAL SEARCH QUALITY

| β | | f_1 | f_2 | f_3 | f_4 | f_5 | f_6 |
|---------|---------|------------------------|------------------------|-------|-----------------------|------------------------|-----------------------|
| 0.5 | mean | 1.8×10^{-97} | 2.81×10^{-12} | 8.43 | 3.2×10^{-4} | 4.14×10^{-15} | 4.36×10^{-4} |
| | Std.dev | 4.21×10^{-97} | 6.28×10^{-12} | 23.13 | 1.9×10^{-4} | 0 | 2.35×10^{-3} |
| 0.6 | mean | 3.22×10^{-97} | 3.13×10^{-12} | 6.81 | 3.2×10^{-4} | 4.14×10^{-15} | 6.5×10^{-4} |
| | Std.dev | 8.76×10^{-97} | 7.08×10^{-12} | 20.31 | 1.99×10^{-4} | 0 | 3.5×10^{-3} |
| 0.7 | mean | 1.66×10^{-98} | 4.52×10^{-12} | 3.10 | 3.59×10^{-4} | 4.26×10^{-15} | 1.31×10^{-3} |
| | Std.dev | 4.05×10^{-98} | 9.71×10^{-12} | 2.65 | 2.63×10^{-4} | 6.38×10^{-16} | 7.05×10^{-3} |
| 0.8 | mean | 1.07×10^{-97} | 5.87×10^{-12} | 2.68 | 3.89×10^{-4} | 4.49×10^{-15} | 2.18×10^{-4} |
| | Std.dev | 2.16×10^{-97} | 2.62×10^{-11} | 2.04 | 2.57×10^{-4} | 1.07×10^{-15} | 1.17×10^{-4} |
| 0.9 | mean | 1.38×10^{-96} | 3.28×10^{-12} | 3.19 | 4.09×10^{-4} | 4.26×10^{-15} | 1.75×10^{-3} |
| | Std.dev | 7.29×10^{-96} | 1.52×10^{-11} | 3.05 | 3.25×10^{-4} | 6.38×10^{-16} | 9.39×10^{-3} |

It can see from the Table V above, the performance of ICPSO improved with the clonal scale coefficient β in equation (3) equal to 0.8. However, the ICPSO nearly have the same performance when the β from 0.5 to 0.9. So, the

parameter β is robust. We recommend using β equal to 0.8 in this paper.

TABLE VI
EFFECTS OF THE NONLINEAR VARIATION COEFFICIENT b ON GLOBAL SEARCH QUALITY

| b | | f_1 | f_2 | f_3 | f_4 | f_5 | f_6 |
|-----|---------|------------------------|------------------------|--------|-----------------------|------------------------|-----------------------|
| 1 | mean | 1.74×10^{-97} | 5.47×10^{-12} | 26.12 | 3.86×10^{-4} | 4.26×10^{-15} | 8.73×10^{-4} |
| | Std.dev | 6.41×10^{-97} | 1.72×10^{-11} | 76.38 | 2.98×10^{-4} | 6.38×10^{-16} | 4.69×10^{-3} |
| 2 | mean | 1.07×10^{-97} | 5.87×10^{-12} | 2.68 | 3.89×10^{-4} | 4.49×10^{-15} | 2.18×10^{-4} |
| | Std.dev | 2.16×10^{-97} | 2.62×10^{-11} | 2.04 | 2.57×10^{-4} | 1.07×10^{-15} | 1.17×10^{-4} |
| 3 | mean | 6.89×10^{-97} | 1.06×10^{-12} | 44.61 | 3.96×10^{-4} | 4.14×10^{-15} | 3.15×10^{-4} |
| | Std.dev | 2.57×10^{-96} | 3.29×10^{-12} | 179.59 | 2.44×10^{-4} | 1.18×10^{-15} | 1.16×10^{-3} |

From the Table VI above, we can see, the performance of ICPSO is better when the nonlinear variation coefficient b in equation (6) equal to 2. At the same time, the performance of

ICPSO is stable when the b from one to three.

TABLE VII
EFFECTS OF THE a_{max} AND a_{min} ON GLOBAL SEARCH QUALITY

| a_{max}, a_{min} | | f_1 | f_2 | f_3 | f_4 | f_5 | f_6 |
|-----------------------------------|---------|------------------------|-----------------------|-------|-----------------------|-------|------------------------|
| $a_{max}=10000,$ $a_{min}=100$ | mean | 1.56×10^{-97} | 1.45×10^{-6} | 15.58 | 2.61×10^{-3} | 18.28 | 3.23×10^{-16} |
| | Std.dev | 3.86×10^{-97} | 2.12×10^{-6} | 5.74 | 9.92×10^{-4} | 5.04 | 3.04×10^{-16} |
| $a_{max}=5000,$ $a_{min}=50$ | mean | 2.52×10^{-97} | 5.98×10^{-6} | 19.82 | 2.17×10^{-3} | 15.26 | 1.06×10^{-16} |
| | Std.dev | 7.37×10^{-97} | 1.96×10^{-6} | 11.19 | 1.06×10^{-3} | 8.36 | 9.77×10^{-17} |
| $a_{max}=1000,$ $a_{min}=10$ | mean | 6.82×10^{-99} | 3.03×10^{-6} | 16.32 | 2.20×10^{-3} | 14.56 | 5.75×10^{-18} |
| | Std.dev | 1.10×10^{-98} | 7.99×10^{-6} | 2.58 | 7.31×10^{-4} | 8.78 | 5.38×10^{-18} |
| $a_{max}=500,$ $a_{min}=5$ | mean | 1.45×10^{-99} | 8.75×10^{-6} | 22.22 | 2.38×10^{-3} | 15.29 | 1.26×10^{-18} |
| | Std.dev | 2.34×10^{-99} | 1.95×10^{-5} | 17.58 | 7.68×10^{-4} | 8.43 | 1.56×10^{-18} |

Empirical study shows that($a_{max}=1000, a_{min}=10$ in equation (11)) result in good performance on most of the test functions in the Table VII. In a statistical sense, the performance of ICPSO is robust when the a_{max} from 10000 to 500 ,and the a_{min} from 100 to 5.

From the Table VII above, we can see, the increasing a provides a higher learning rate in the early stage for $Pbest$ to jump out of a possible local optimum, whereas a smaller learning rate in the latter stage guides the $Pbest$ to refine the solution.

TABLE VIII
EFFECTS OF SOME OPERATORS ON GLOBAL SEARCH QUALITY

| | | f_1 | f_2 | f_3 | f_4 | f_5 | f_6 |
|---------------------------|---------|---|---|---|---|--|--|
| ICPSO with only wavelet | mean | 6.82×10^{-99} | 3.03×10^{-6} | 16.32 | 2.20×10^{-3} | 14.56 | 5.75×10^{-18} |
| | Std.dev | 1.10×10^{-98} | 7.99×10^{-6} | 2.58 | 7.31×10^{-4} | 8.78 | 5.38×10^{-18} |
| ICPSO with only AIS | mean | 1.07×10^{-97} | 5.87×10^{-12} | 2.68 | 3.89×10^{-4} | 4.49×10^{-15} | 2.18×10^{-4} |
| | Std.dev | 2.16×10^{-97} | 2.62×10^{-11} | 2.04 | 2.57×10^{-4} | 1.07×10^{-15} | 1.17×10^{-4} |
| ICPSO with only migration | mean | 1.69×10^{-97} | 2.04×10^{-6} | 18.06 | 3.48×10^{-3} | 4.14×10^{-15} | 2.98×10^{-15} |
| | Std.dev | 5.25×10^{-97} | 4.41×10^{-6} | 8.51 | 1.47×10^{-3} | 0 | 3.66×10^{-15} |
| ICPSO with only AIN-PSO | mean | 2.45×10^{-92} | 4.74×10^{-4} | 21.32 | 4.15×10^{-3} | 4.14×10^{-15} | 7.39×10^{-13} |
| | Std.dev | 8.33×10^{-92} | 7.95×10^{-4} | 18.54 | 1.42×10^{-3} | 0 | 7.68×10^{-13} |
| ICPSO with each | mean | 1.02×10^{-284} | 8.61×10^{-147} | 7.53×10^{-7} | 1.34×10^{-4} | 8.25×10^{-16} | 1.57×10^{-32} |
| | Std.dev | 0 | 4.61×10^{-146} | 8.19×10^{-7} | 7.45×10^{-5} | 8.86×10^{-16} | 2.74×10^{-48} |
| ICPSO without Either | mean | 9.35×10^{-17} | 8523.19 | 23.89 | 1.13×10^{-2} | 15.51 | 7.84×10^{-3} |
| | Std.dev | 2.92×10^{-17} | 1481.34 | 0.876 | 2.67×10^{-3} | 3.02 | 4.18×10^{-5} |

It is clear from Table VIII the results that with adaptive wavelet learning, ICPSO has good solutions on unimodal and multimodal functions. The adaptive wavelet learning offer good performance for ICPSO. The wavelet learning can speed up the convergence of the algorithm. On the other hand, the ICPSO without adaptive wavelet learning can hardly jump out of the local optima. The reasons are the adaptive wavelet learning can help $pbest$ jump out of the local optimal region. It is very clear from Table VIII. The ICPSO suffers from lower accuracy in solutions without AIS, since algorithms can easily get trapped in the global optimal region. AI contributes more to helping the elite particles in the memory move away from its existing position so as to jump out of the local optima. ICPSO with AIS performs better than ICPSO without AIS. At the same time, we also find ICPSO with wavelet, AIS, migration and AIN-PSO collaborative operation having the best optimization performance than ICPSO without either operation. From the Table X, we can see the wavelet, AIS, migration and AIN-PSO each operator contributes more to the ICPSO.

F. Compared with other PSO using AIS

PSO-AIS (the combination between AIS and PSO) was proposed in [33]. For a fair comparison between the PSO-AIS and the ICPSO method, the algorithms are tested using the

same setting parameter such as 500 iterations, test functions 10 dimensions, and the population size is 20, these values are adopted in [33]. Comparative tests have been performed using functions $f_1, f_2, f_8,$ and f_9 are listed in [33]. The results are shown in Table IX in terms of the mean fitness and standard deviation (Std. Dev) of the solutions obtained in the 10 independent runs on each problem. The results of the PSO-AIS comes from [33]. We can see, the ICPSO gives the best performance for all five optimization problems on the mean fitness and standard deviation (Std. Dev) of the solutions, outperforms the PSO-AIS.

TABLE IX
COMPARISONS BETWEEN PSO-AIS AND ICPSO

| | | f_1 | f_2 | f_8 | f_9 |
|---------|---------|-----------------------|------------------------|------------------------|------------------------|
| PSO-AIS | mean | 4.89 | 8.2×10^{-7} | 2.6×10^{-7} | 1.47 |
| | Std.dev | 1.094 | 6.3×10^{-7} | 2.1×10^{-7} | 0.856 |
| ICPSO | mean | 2.04×10^{-3} | 7.37×10^{-33} | 1.67×10^{-11} | 1.219×10^{-8} |
| | Std.dev | 4.5×10^{-4} | 3.05×10^{-33} | 3.22×10^{-11} | 1.47×10^{-8} |

G. Shifted Functions Test

In addition, four shifted functions [34]-[35] are used to evaluate the global search ability of the ICPSO algorithm. The problems existing in some benchmark that have the same values among all independent variables at the global optima can be avoided through the shifted functions. Since the global

optimums of the shifted functions have different parameter values for different dimensions and there is no linking among these variables. The details are described in [34]-[35]. The expressions of the shifted functions are depicted in Table X.

All these functions are tested 30 times and with 30 dimensions. The basic experimental setup is the same as that mentioned in subsection c of section III.

TABLE X
SHIFTED FUNCTIONS

| Test Function | Domain Range | Z | Global Optimal |
|--|--------------------------|-------------|----------------|
| $F_{1-shifted}(x) = \max\{ z_i , 1 \leq i \leq D\} - 450$ | $-100 \leq X_i \leq 100$ | $z = X - o$ | -450 |
| $F_{2-shifted}(x) = \sum_{i=1}^D \frac{z_i^2}{4000} - \prod_{i=1}^D \cos(\frac{z_i}{\sqrt{i}}) - 180$ | $-600 \leq X_j \leq 600$ | $z = X - o$ | -180 |
| $F_{3-shifted}(x) = \sum_{i=1}^{30} z_i^2 - 450$ | $-100 \leq X_i \leq 100$ | $z = X - o$ | -450 |
| $F_{4-shifted}(x) = -20 \exp(-0.2 \sqrt{\frac{1}{D} \sum_{i=1}^D z_i}) - \exp(\frac{1}{D} \sum_{i=1}^D \cos(2\pi z_i)) + 20 + e - 140$ | $-32 \leq X_i \leq 32$ | $z = X - o$ | -140 |

TABLE XI
COMPARISON BETWEEN DIFFERENT PSO METHODS FOR SHIFTED FUNCTIONS

| Shifted Function | HPSOM | HGAPSO | HPSOWM | CLPSO | A-CLPSO | APSO | ICPSO | |
|------------------|---------|---------|---------|------------------------|------------------------|-----------------------|------------------------|-----------------------|
| $F_{1-shift}$ | Mean | -400.94 | -446.89 | -449.34 | -446.59 | -449.935 | -449.925 | -450 |
| | Std.Dev | 9.25 | 2.967 | 0.385 | 0.480 | 0.115 | 5.20×10^{-2} | 2.02×10^{-5} |
| $F_{2-shift}$ | Mean | -88.737 | -179.70 | -179.956 | -179.839 | -179.999 | -179.989 | -179.993 |
| | Std.Dev | 49.6824 | 0.247 | 8.61×10^{-2} | 0.532 | 2.49×10^{-3} | 1.02×10^{-2} | 9.89×10^{-3} |
| $F_{3-shift}$ | Mean | 6915.91 | -449.94 | -450 | -450 | -450 | -450 | -450 |
| | Std.Dev | 5669.53 | 0.165 | 6.89×10^{-10} | 5.59×10^{-10} | 0 | 1.80×10^{-14} | 0 |
| $F_{4-shift}$ | Mean | -126.02 | -132.58 | -139.329 | -140 | -140 | -139.778 | -140 |
| | Std.Dev | 2.827 | 5.557 | 1.110 | 7.71×10^{-8} | 0 | 0.449 | 0 |

From the Table XI, we can see that the ICPSO show best performance in terms of the mean fitness and the standard deviation than other PSOs on $F_{1-shift}$, we can also see that the ICPSO show best performance in terms of the mean fitness and the standard deviation than other PSOs but A-CLPSO on $F_{2-shift}$, $F_{3-shift}$ and $F_{4-shift}$. However, the ICPSO is comparable to the A-CLPSO in terms of the mean fitness and the standard

deviation on $F_{2-shift}$, $F_{3-shift}$ and $F_{4-shift}$. Based on this observation, it is very clear that the ICPSO can solve the optimization problems with the global optimum points shifted and rotated almost perfectly.

H. High dimension function test

TABLE XII
COMPARISON BETWEEN DIFFERENT PSO METHODS FOR HIGH DIMENSION FUNCTION WITH 200 DIMENSIONS

| Test Function with 200 Dimensions | HPSOM | HGAPSO | HPSOWM | CLPSO | A-CLPSO | APSO | ICPSO | |
|-----------------------------------|---------|-----------------------|------------------------|---------|-----------------------|------------------------|------------------------|-------------------------|
| f_1 | Mean | 130535 | 8.57×10^{-4} | 4.508 | 446.664 | 2.08×10^{-25} | 4.17×10^{-73} | 5.99×10^{-85} |
| | Std.Dev | 30998.7 | 3.90×10^{-4} | 1.95 | 62.7369 | 1.02×10^{-25} | 1.46×10^{-72} | 1.26×10^{-84} |
| f_2 | Mean | 607269 | 295888 | 328963 | $1.08 \times 10^{+6}$ | 262875 | 0.6519 | 8.82×10^{-46} |
| | Std.Dev | 106598 | 67083.8 | 50966.9 | 81500.1 | 32501 | 0.5705 | 2.62×10^{-45} |
| f_3 | Mean | $1.36 \times 10^{+6}$ | 541.164 | 363.609 | 39443.4 | 257.433 | 178.194 | 179.137 |
| | Std.Dev | $1.21 \times 10^{+6}$ | 94.1255 | 72.0284 | 6837.13 | 51.8879 | 4.39218 | 0.825 |
| f_4 | Mean | 604.181 | 0.483 | 0.0809 | 5.21435 | 0.0699 | 5.50×10^{-3} | 1.46×10^{-4} |
| | Std.Dev | 325.347 | 0.067 | 0.0115 | 0.742 | 0.01325 | 2.10×10^{-3} | 1.22×10^{-4} |
| f_5 | Mean | 19.3779 | 9.675×10^{-2} | 5.6596 | 20.191 | 4.88×10^{-14} | 1.52×10^{-14} | 1.373×10^{-14} |
| | Std.Dev | 0.2676 | 0.373 | 0.5956 | 0.0346 | 8.93×10^{-15} | 4.14×10^{-15} | 2.774×10^{-14} |
| f_6 | Mean | 8.12×10^{-7} | 11.845 | 8.878 | $2.82 \times 10^{+6}$ | 5.19×10^{-3} | 0.996 | 2.546×10^{-7} |
| | Std.Dev | 1.37×10^{-8} | 2.467 | 2.026 | 874605 | 1.77×10^{-2} | 3.33×10^{-16} | 3.077×10^{-7} |

A suite of high dimensional function 200 dimensions are carried out in this subsection to validate the proposed ICPSO has the ability of solve the high dimensional problem. Using

functions f_1 , f_2 , f_3 , f_4 , f_5 , and f_6 listed in Table I with 200 dimensions. All these functions are tested 30 times in terms of the mean fitness and the standard deviation. Further, use

the same number of 4000FEs for each problem on the all algorithms. Other parameter setting is the same as the subsection c of section III.

From Table XII, it shows that ICPSO has the best performance in solving most of these high dimensional functions than other PSOs but A-CLPSO on f_3 . However, the ICPSO is comparable to the A-CLPSO in terms of the mean fitness on f_3 , but the standard deviation of the ICPSO is better than APSO. Furthermore, ICPSO also shows the highest accuracy in solving these high dimensional problems with 200 dimensions.

IV. EXPERIMENTAL VERIFICATION

In this section, the proposed ICPSO is applied for the multi-parameter estimation of permanent magnet synchronous machines (PMSMs). PMSM has been widely used in servo control and wind power generation [36]-[44] thanks to its fast torque response, high power density and high efficiency etc. However, in real application, it is necessary to accurately obtain the PMSM parameter values prior to the design of related control systems. Thus, technologies for multi-parameter estimation of PMSMs have been widely reported in existing papers [40]-[42] and become one of the most popular research topics in machine control.

Existing literature references mainly focus on online estimation and algorithms such as extended Kalman filter (EKF), model reference adaptive system (MRAS), recursive least-square (RLS) methods, neural network (NN) [39]-[41] and evolutionary algorithms[42][43]are widely employed. In order to improve the quality of multi-parameter estimation in PMSM, thus, in this paper, the proposed ICPSO will be employed for estimating the dq -axis inductances, stator winding resistance and rotor flux linkage of a prototype PMSM and its performance will be compared with other newly published PSOs.

A. PMSM Model

Assuming that the PMSM has ideal physical performances and mechanical structures, the dq -axis equations of the PMSM are given by:

$$\begin{cases} \frac{di_d}{dt} = -\frac{R}{L}i_d + \omega i_q + \frac{u_d}{L} \\ \frac{di_q}{dt} = -\frac{R}{L}i_q - \omega i_d + \frac{u_q}{L} - \frac{\psi}{L}\omega \end{cases} \quad (13)$$

Where i_d , i_q , u_d and u_q are dq axis stator current and voltage, ω is the electrical angle speed respectively, R , L_d , L_q and ψ_m are the motor winding resistance, dq -axis inductances and magnet flux ($L_d \approx L_q \approx L$). After low-pass filtering, the parameter estimation can be based on the steady-state machine model [44], which can be expressed in discrete form.

$$\begin{cases} u_d(k) = Ri_d(k) - L_q\omega(k)i_q(k) \\ u_q(k) = Ri_q(k) + L_d\omega(k)i_d(k) + \psi_m\omega(k) \end{cases} \quad (14)$$

Generally, i_d is set to be 0 for decoupling the flux and

torque control. Thus, (14) can be simplified to be (15).

$$\begin{cases} u_d(k) = -L_q\omega(k)i_q(k) \\ u_q(k) = Ri_q(k) + \psi_m\omega(k) \end{cases} \quad (15)$$

In real application, R , L_d , L_q , ψ_m are unknown parameters to be identified. From (14) and (15), it is evident that their rank numbers are both two while there are four parameters to be identified [42]. Thus, it is detailed in [43] that it is impossible to identify these parameters simultaneously and a short time of $i_d < 0$ should be injected for obtaining a full rank reference model [42], [43], which is shown as follows.

$$\begin{cases} u_{d0}(k_0) = -L_{q0}\omega(k_0)i_{q0}(k_0) \\ u_{q0}(k_0) = Ri_{q0}(k_0) + \psi_{m0}\omega(k_0) \\ u_d(k_1) = Ri_d(k_1) - L_q\omega(k_1)i_q(k_1) \\ u_q(k_1) = Ri_q(k_1) + L_d\omega(k_1)i_d(k_1) + \psi_m\omega(k_1) \end{cases} \quad (16)$$

In the following investigation, (16) is the full rank model used for the multi-parameter estimation of PMSM and will be used for the design of penalty function.

B. Parameter Identification by using ICPSO

Based on (16), the penalty function for estimating the dq -axis inductances, winding resistance and rotor flux linkage is shown as follows.

$$f(\hat{p}) = \sum_{k=1}^n w_1 |u_{d0}(k) - \hat{u}_{d0}(k)| + w_2 |u_{q0}(k) - \hat{u}_{q0}(k)| + w_3 |u_d(k) - \hat{u}_d(k)| + w_4 |u_q(k) - \hat{u}_q(k)| \quad (17)$$

Where w_1 , w_2 , w_3 , w_4 are the weight coefficients and the variables with '^' mean that they are computed voltages by the estimated parameters and measured currents. The actual machine parameter values can be obtained if (17) is minimized by the proposed ICPSO. Thus, the optimization of (17) can be regarded as a multi-dimensional function optimization problem.

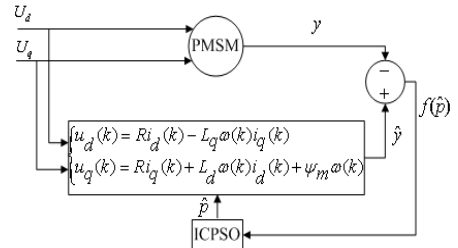


Fig. 8. Parameter identification model based on ICPSO.

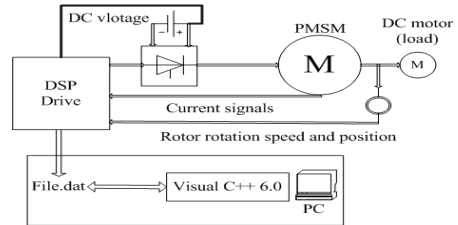


Fig. 9. Schematic diagram of Identification system.

C. Hardware Platform for Experiments

The proposed estimator is verified by experiments in this section. The offline estimation model shown in Fig.8.

TABLE XIII
DESIGN PARAMETERS AND SPECIFICATION OF PMSM

| | |
|---|------------------------|
| Rated current | 4A |
| Rated speed | 400rpm |
| DC link voltage | 36v |
| Nominal phase resistance ($T=25\text{ }^{\circ}\text{C}$) | 0.330 Ω |
| Nominal terminal wire resistance | 0.043 |
| Nominal self inductance | 2.91mh |
| Nominal mutual inductance | -0.330mh |
| Nominal d -axis inductance | 3.24mh |
| Nominal q -axis inductance | 3.24mh |
| Nominal amplitude of flux induced by magnets | 77.6 mWb |
| Inertia | 0.8e-5kgm ² |
| Number of pole pairs | 5 |

The DSP based vector control hardware platform and the schematic diagram of testing process are shown in Fig.9. The design parameters of prototype machine are shown in Table XIII. For comparison, the PMSM parameter values are also identified by other PSOs such as HPSOM, HGAPSO,

TABLE XIV
RESULT COMPARISONS AMONG SEVEN PSOS ON PMSM PARAMETER IDENTIFICATION WITH NORMAL TEMPERATURE

| | HPSOM | HGAPSO | HPSOWM | CLPSO | A-CLPSO | APSO | ICPSO | |
|----------------------|---------|---------|---------|---------|---------|---------|---------|-------|
| T=300 | | | | | | | | |
| Identification value | | | | | | | | |
| R (Ω) | 0.338 | 0.332 | 0.363 | 0.367 | 0.405 | 0.372 | 0.3734 | |
| ψ_m (wb) | 0.0790 | 0.0792 | 0.0783 | 0.0783 | 0.0770 | 0.0783 | 0.0778 | |
| $Ld(h)$ | 0.00344 | 0.00275 | 0.00376 | 0.00382 | 0.00380 | 0.00336 | 0.00330 | |
| $Lq(h)$ | 0.00395 | 0.00398 | 0.00387 | 0.00388 | 0.00371 | 0.00398 | 0.00396 | |
| fitness | mean | 0.247 | 1.888 | 0.844 | 2.375 | 2.726 | 4.163 | 0.149 |
| | Std.dev | 0.182 | 0.943 | 0.457 | 0.890 | 0.912 | 2.276 | 0.028 |
| | t-value | 3.7632 | 13.0341 | 10.7335 | 17.6769 | 19.9710 | 12.4697 | N/A |

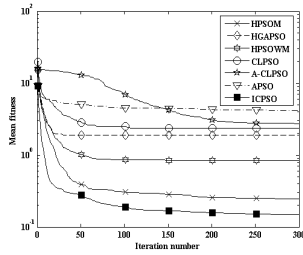


Fig. 10. The fitness convergence curve of seven psos on PMSM parameter identification with normal temperature.

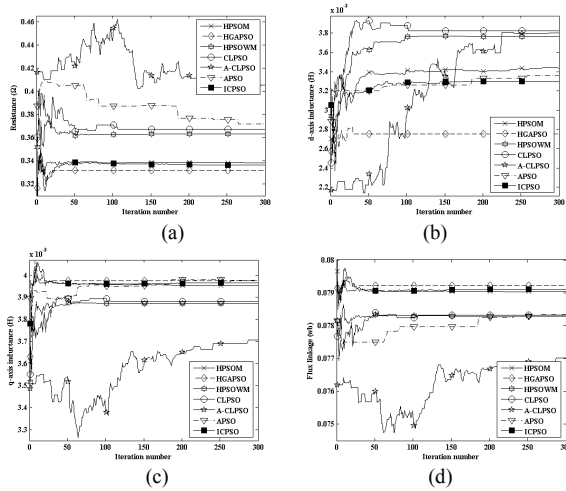


Fig. 11. Identified parameters with Normal temperature (a) Estimated winding resistance. (b) Estimated d -axis inductance. (c) Estimated q -axis inductance. (d) Estimated rotor flux linkage.

The statistical results in terms of mean value of time cost,

HPSOWM, CLPSO, A-CLPSO, and APSO. The basic settings of these PSOs are the same as those in Section III while the maximum generation is set to be 300 and the average results of evolution computation of 30 times are memorized.

i. Experiments under Normal Temperature

The convergence rates of different PSOs are shown in Fig.10 and the experimental results are shown in Table X IV. From Table X IV and Fig.10, it is obvious that the ICPSO shows the best performances in terms of mean, standard deviations and t -values. All t -values are higher than 2.06, implying that 0 the ICPSO is significantly better, with a 98% confidence level, than other hybrid PSOs. From Fig.10, the convergence speed of ICPSO is faster than other hybrid PSOs.

standard deviation and t -test value, are shown in Table X IV. It is evident that the stability of the optimization is improved, thanks to the multi-population, and the smallest standard deviation is achieved by ICPSO. As can be seen from the Table X IV, the estimated winding resistance (0.3734 Ω) is almost the same as the measured resistance (0.373 Ω), under normal temperature, it is shown in Fig.11 that the results of identified PMSM parameters by using ICPSO are of high accuracy and the estimated parameters such as motor resistance, dq -axis inductances and the rotor flux converge to their right points rapidly. The four parameters estimated results with seven different PSOs are shown in Fig.11.

ii. Experiments under Temperature Variation

A heater is used to heat the prototype PMSM for 20 minutes. The identified training results are listed in Table X V, and the comparison with different PSOs is shown in Fig.12 and 13, respectively.

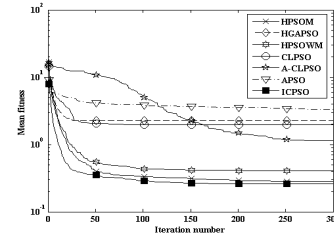


Fig. 12. The fitness Convergence curve of seven psos on PMSM parameter identification with temperature variation.

The convergence rates of different PSOs are shown in Fig.12. From Table X V, it is evident that ICPSO shows best performances in terms of mean, standard deviations and

t -values. All t -values are higher than 2.06, implying that the ICPSO is significantly better, with a 98% confidence level, than other hybrid PSOs. It also shows that the convergence

speed of ICPSO is faster than other hybrid PSOs. The four parameters estimated results with seven different PSOs are shown in Fig.13.

TABLE X V
RESULT COMPARISONS AMONG SEVEN PSOS ON PMSM PARAMETER IDENTIFICATION WITH TEMPERATURE VARIATION

| $T=300$ | | HPSOM | HGAPSO | HPSOWM | CLPSO | A-CLPSO | APSO | ICPSO |
|---------------|---------|----------------------|---------|---------|---------|---------|---------|---------|
| | | Identification value | | | | | | |
| $R (\Omega)$ | | 0.453 | 0.459 | 0.458 | 0.473 | 0.479 | 0.455 | 0.453 |
| $\psi_m (wb)$ | | 0.0769 | 0.0770 | 0.0768 | 0.0764 | 0.0762 | 0.0772 | 0.0770 |
| $Ld(h)$ | | 0.00347 | 0.00341 | 0.00411 | 0.00414 | 0.00388 | 0.00370 | 0.00364 |
| $Lq(h)$ | | 0.00376 | 0.00378 | 0.00376 | 0.00370 | 0.00369 | 0.00386 | 0.00376 |
| fitness | mean | 0.283 | 2.246 | 0.410 | 1.968 | 1.135 | 3.300 | 0.261 |
| | Std.dev | 0.121 | 1.121 | 0.215 | 0.841 | 0.483 | 3.866 | 0.147 |
| | t-value | 0.8171 | 12.4147 | 4.0453 | 14.1380 | 12.2409 | 5.5544 | 0 |

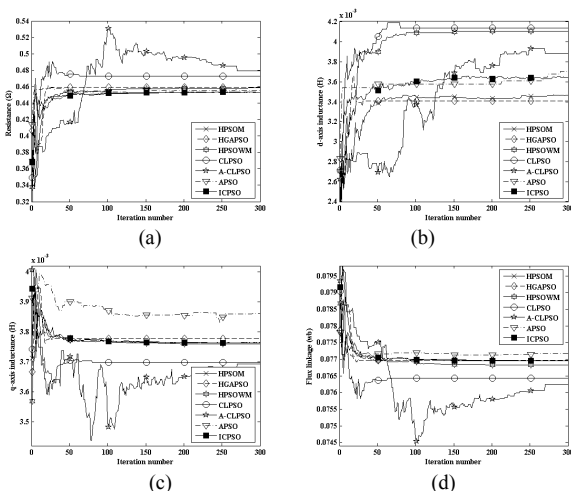


Fig. 13. Identified parameters with temperature variation.
(a) Estimated winding resistance. (b) Estimated d -axis inductance.
(c) Estimated q -axis inductance. (d) Estimated rotor flux linkage.

V. CONCLUSION

This paper proposes a new PSO named ICPSO, which incorporates artificial immune system with PSO. The algorithm based on multi-population and memory scheme. Each Sub-Population's individuals in a new generation are created by three hybrid methods: elitist reservations scheme, immune network and Cauchy mutation, Reinitialize, etal. Particle's $pbest$ positions within subpopulations were learned by Wavelet learning strategy for accelerating convergence speed. Further, a Wavelet perturbation-based learning strategy is developed to lead the $pbest$ particles to jump out of any possible local optima and also to refine converging solutions. The best solutions are stored in the memory and optimized by using improved immune clonal selection algorithm. We also investigated the interactions between the memory and several normal sub-populations; the investigated memory scheme is efficient for improving the ICPSO. Then, ICPSO is applied to solve the parameter identification of PMSM problem; the experiment results demonstrate that the ICPSO is superior to other hybrid PSOs. The experimental on the PMSM platform results show that the proposed strategy has good convergence in simultaneously estimating winding resistance, dq axis inductances and rotor flux linkage.

REFERENCES

- [1] R. C. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," in *Proc. 6th Int. Symp. Micromachine Human Sci.*, Nagoya, Japan, 1995, pp. 39–43.
- [2] J. Kennedy and R. C. Eberhart, "Particle swarm optimization," in *Proc. IEEE Int. Conf. Neural Netw.*, Perth, Australia, 1995, vol. 4, pp. 1942–1948.
- [3] C. F. Juang, "A hybrid of genetic algorithm and particle swarm optimization for recurrent network design," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 34, no. 2, pp. 997–1006, Apr. 2004.
- [4] F.J. Lin, L.T. Teng, J.W. Lin, S.Y. Chen, "Recurrent Functional-Link-Based Fuzzy-Neural-Network-Controlled Induction-Generator System Using Improved Particle Swarm Optimization," *IEEE Trans. Ind. Electron.*, vol. 56, no. 5, pp. 1557 - 1577, May, 2009.
- [5] C.H. Liu and Y.Y. Hsu, "Design of a Self-Tuning PI Controller for a STATCOM Using Particle Swarm Optimization," *IEEE Trans. Ind. Electron.*, vol. 57, no. 2, pp. 702 – 715, Feb. 2010.
- [6] R.J. Wai, K.L. Chuang and J.D. Lee, "On-Line Supervisory Control Design for Maglev Transportation System via Total Sliding-Mode Approach and Particle Swarm Optimization" *IEEE Trans. Auto. Contr.*, vol. 55, no. 7, pp. 1544 – 1559, Jul. 2010.
- [7] H.L. Wei, S.A. Billings, Y.F. Zhao and L.Z. Guo, "Lattice Dynamical Wavelet Neural Networks Implemented Using Particle Swarm Optimization for Spatio-Temporal System Identification," *IEEE Trans. Neural Networks*, vol. 20, no. 1, pp. 181– 185, Jan. 2009.
- [8] L. Liu, W.X. Liu and D. A. Cartes, "Permanent Magnet Synchronous Motor Parameter Identification using Particle Swarm Optimization" *International Journal of Computational Intelligence Research*. vol. 4, no. 2, pp. 211–218, Oct. 2008.
- [9] R. Marion, R. Scorretti, N. Siauve, M.A. Raulet and L. Krahenbihl, "Identification of Jiles-Atherton Model Parameters Using Particle Swarm Optimization," *IEEE Trans. Magnetics*, vol. 44, no. 4, pp. 894–897, Jun. 2008.
- [10] A. A. E. Ahmed, L. T. Germano, and Z. C. Antonio, "A hybrid particle swarm optimization applied to loss power minimization," *IEEE Trans. Power Syst.*, vol. 20, no. 2, pp. 859–866, May, 2005.
- [11] C. F. Juang, "A hybrid of genetic algorithm and particle swarm optimization for recurrent network design," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 34, no. 2, pp. 997–1006, Apr. 2004.
- [12] P. J. Angeline, "Using Selection to Improve Particle Swarm Optimization", *Proceedings of IEEE International Conference on Evolutionary Computation*, Anchorage, pp. 84–89, May 1998.
- [13] S.H. Ling, H.H.C. Ju; K.Y. Chan, H.K. Lam, B.C.W. Yeung and F.H. Leung, "Hybrid Particle Swarm Optimization With Wavelet Mutation and Its Industrial Applications," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 38, no. 3, pp. 743 – 763, Jun. 2008.
- [14] Y. Shi and R. C. Eberhart, "A modified particle swarm optimizer," in *Proc. IEEE World Congr. Comput. Intell.*, 1998, pp. 69–73.
- [15] Y. Shi and R. C. Eberhart, "Particle swarm optimization with fuzzy adaptive inertia weight," in *Proc. Workshop Particle Swarm Optimization*, Indianapolis, IN, 2001, pp. 101–106.
- [16] Z.H. Zhan, J. Zhang, Y. Li, and H. S.H. Chung, "Adaptive Particle Swarm Optimization," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 39, no. 6, pp. 1362–1381, Dec. 2009.

[17] J. J. Liang, A. K. Qin, P. N. Suganthan, and S. Baskar, "Comprehensive learning particle swarm optimizer for global optimization of multimodal functions," *IEEE Trans. Evol. Comput.*, vol. 10, no. 3, pp. 281–295, Jun. 2006.

[18] H. Wu, J. P. Geng, R. H. Jin, J. Z. Qiu, W. Liu, J. Chen, S. Liu, "An Improved Comprehensive Learning Particle Swarm Optimization and Its Application to the Semiautomatic Design of Antennas," *IEEE Trans. Antennas and Propagation*, vol. 57, no. 10, pp. 3018–3028, Oct. 2009.

[19] J. Kennedy and R. Mendes, "Population structure and particle swarm performance," in *Proc. IEEE Congr. Evol. Comput.*, Honolulu, HI, 2002, pp. 1671–1676.

[20] J. Kennedy and R. Mendes, "Neighborhood topologies in fully informed and best-of-neighborhood particle swarms," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 36, no. 4, pp. 515–519, Jul. 2006.

[21] F. Van den Bergh and A. P. Engelbrecht, "A cooperative approach to particle swarm optimization," *IEEE Trans. Evol. Comput.*, vol. 8, no. 3, pp. 225–239, Jun. 2004.

[22] R. A. Krohling and L. S. Coelho, "Coevolutionary particle swarm optimization using Gaussian distribution for solving constrained optimization problems," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 36, no. 6, pp. 1407–1416, Dec. 2006.

[23] G. G. Yen, and W. F. Leong, "Dynamic Multiple Swarms in Multiobjective Particle Swarm Optimization". *IEEE Trans. Syst. Man Cybern. Part A-Syst. Hum.*, vol. 39, no. 4, pp. 890–911, Jul. 2009.

[24] B. Niu, Y. Zhu, and X. He, "Multi-population cooperative particle swarm optimization," in *Proc. Eur. Conf. Artif. Life, Canterbury, U.K.*, 2005, pp. 874–883.

[25] L. N. de Castro and F. J. von Zuben, "Learning and optimization using the clonal selection principle," *IEEE Trans. Evol. Comput.*, vol. 6, no. 1, pp. 239 – 251, Jun. 2002.

[26] A. M. Whitbrook, U. Aickelin, J. M. Garibaldi, "Idiotypic Immune Networks in Mobile-Robot Control," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 37, no. 6, pp. 1581 – 1598, Dec. 2007.

[27] L. dos Santos Coelho, P. Alotto, "Loney's Solenoid Design Using an Artificial Immune Network With Local Search Based on the Simplex Method," *IEEE Trans. Magnetics*, vol. 44, no. 6, pp. 1070–1073, Jun. 2008.

[28] C. Yan, G. K. Venayagamoorthy, K. Corzine, "Hardware Implementation of an AIS-Based Optimal Excitation Controller for an Electric Ship," *IEEE Trans. Ind. Appl.*, vol. 44, no. 2, pp. 1060 – 1070, Mar. 2011.

[29] Woldemariam, K. M.; Yen, G. G., "Vaccine-Enhanced Artificial Immune System for Multimodal Function Optimization". *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 40, no. 1, pp. 218 – 228, Feb. 2010.

[30] M. Potter, K. De Jong, "A cooperative coevolutionary approach to function optimization", in *Proc. 3rd. Parallel Problem Solving from Nature*. Berlin, Germany, 1994, pp. 249 – 257.

[31] I. Daubechies, *Ten Lectures on Wavelets*. Philadelphia, PA: SIAM, 1992.

[32] X. Yao, Y. Liu and G. M. Lin, "Evolutionary Programming Made Faster," *IEEE Trans. Evol. Comput.*, vol. 3, no. 2, pp. 82–102, Jul. 1999.

[33] David F. W. Yap, S. P. Koh, S. K. Tiong, "Mathematical Function Optimization using AIS Antibody Remainder method", *International Journal of Machine Learning and Computing*, Vol. 1, No. 1, pp. 13–19, April, 2011.

[34] J. J. Liang, P. N. Suganthan, K. Deb. Novel composition test functions for numerical global optimization, in *Proc. IEEE Swarm Intell. Symp.*, Pasadena, CA, pp. 68–75, Jun. 2005.

[35] K. Tang, X. Yao, P. N. Suganthan, C. MacNish, Y. P. Chen, C. M. Chen, and Z. Yang, "Benchmark functions for the CEC'2008 special session and competition on large scale global optimization," *Nature Inspired Comput. Applicat. Lab., USTC, China, Nanyang Technol. Univ.*, Singapore, Tech. Rep., 2007.

[36] M. C. Chou, C. M. Liaw, S. B. Chien, F. H. Shieh, J. R. Tsai and H. C. Chang, "Robust Current and Torque Controls for PMSM Driven Satellite Reaction Wheel," *IEEE Trans. Aero sp. Electron. Syst.*, vol. 47, no. 1, pp. 58–74, Jan. 2011.

[37] P. Zheng, J. Zhao, R. Liu, C. D. Tong and Q. Wu, "Magnetic Characteristics Investigation of an Axial-Axial Flux Compound-Structure PMSM Used for HEVs," *IEEE Trans. Mag.*, vol. 46, no. 6, pp. 2191 – 2194, Jun. 2010.

[38] Y. A.-R. I. Mohamed, E. F. El-Saadany, "A Current Control Scheme With an Adaptive Internal Model for Torque Ripple Minimization and Robust

Current Regulation in PMSM Drive Systems," *IEEE Trans. Ener. Con.*, vol. 23, no. 1, pp. 92 – 100, Mar. 2008.

[39] B. N. Mobarakeh, F. Meibody-Tabar, and F. M. Sargos, "Mechanical sensorless control of PMSM with online estimation of stator resistance," *IEEE Trans. Ind. Applicat.*, vol. 40, no. 2, pp. 457–471, Mar.-Apr., 2004.

[40] S. J. Underwood and I. Husain, "Online Parameter Estimation and Adaptive Control of Permanent-Magnet Synchronous Machines," *IEEE Trans. Ind. Electron.*, vol. 57, no. 7, pp. 2435 – 2443, Jul. 2010.

[41] F. F. M. El-Sousy, "Robust wavelet-neural-network sliding-mode control system for permanent magnet synchronous motor drive," *IET Electr Power App.*, vol. 5, no. 1, pp. 113 – 132, Jan. 2011.

[42] K. Liu, Q. Zhang, J. T. Chen, Z. Q. Zhu, J. Zhang, and Shen A. W., "Online multiparameter estimation of nonsalient-pole PM synchronous machines with temperature variation tracking," *IEEE Trans. Ind. Electron.*, vol. 58, no. 5, pp. 1776 – 1788, May. 2011.

[43] K. Liu, Z. Q. Zhu, and J. Zhang, "Multi-parameter estimation of non-salient pole permanent magnet synchronous machines by using evolutionary algorithms," *IEEE Int. Conf. BIC-TA, Changsha, China*, 2010, pp. 766 – 774.

[44] A. M. N. Lima, C. B. Jacobina, and E. B. D. S. Filho, "Nonlinear parameter estimation of steady-state induction machine models," *IEEE Trans. Ind. Electron.*, vol. 44, no. 3, pp. 390–397, Jun. 1997.



Zhao-Hua Liu He received M.Sc. degrees in computer science and technology and the PhD. degree in electrical engineering from the Hunan University, China, 2010 and 2012, respectively. Currently, he was with the School of Information and Electrical Engineering, Hunan University of Science and Technology, Xiangtan, China. His current research interests include evolutionary computation Techniques, Intelligent control, Computer control technology, parallel computing, and the parameter identification of the permanent magnet synchronous motor.



Jing Zhang received his B.Eng., M.Sc. and PHD degrees from Hunan University, China, in 1982, 1984 and 1997, respectively. His research interests are in optimal control, fuzzy control and intelligent control of rotary kiln. He has published more than 100 papers in journals and conferences and was awarded China national second prize of scientific and technological progress.



Shao-Wu Zhou He received M.Sc. degrees and the PhD. degree in electrical engineering from the Central south university and Hunan University, China, 1990 and 2005, respectively. He is currently a Full Professor with the School of Information and Electrical Engineering, Hunan University of Science and Technology, Xiangtan, China. His research interests are in intelligent control and robot control.



Xiao-Hua Li received the B. Eng. and M.Sc. degrees in computer science and technology from Hunan University of Science and Engineering, Hunan university, China, in 2007 and 2010, respectively. Currently, she was with the School of Information and Electrical Engineering, Hunan University of Science and Technology, Xiangtan, China. Her interests in evolutionary computation Techniques, Computer control technology.



Kan Liu received his B.Eng. PhD. degree in automation from the Hunan University, China, in 2005, 2011 respectively. Currently, he is working in the Department of Electronics and Electrical Engineering at the University of Sheffield. His research interest focuses on brushless AC motor parameters estimation by control theory and nonlinearity compensation for VSI output voltage.