

Solving IIR system identification by a variant of particle swarm optimization

De-Xuan Zou¹ · Suash Deb² · Gai-Ge Wang^{3,4,5,6}

Received: 9 March 2016 / Accepted: 3 May 2016 / Published online: 14 May 2016
© The Natural Computing Applications Forum 2016

Abstract A variant of particle swarm optimization (PSO) is represented to solve the infinitive impulse response (IIR) system identification problem. Called improved PSO (IPSO), it makes significant enhancement over PSO. To begin with, the population initialization step makes use of golden ratio to segment solution space so as to obtain high-quality solutions. It is followed by all particles using different inertia weights in velocity updating step, which is beneficial for preserving the balance between global search and local search. Subsequently, IPSO uses normal distribution to disturb the global best particle, which enhances its capacity of escaping from the local optimums. The above three operations cannot only guarantee high-quality solutions, strong global search capacity, and fast convergence rate, but also avoid low diversity, excessive local search, and premature stagnation. These properties of IPSO make it much better suited for IIR system identification problems. IPSO is applied on 12 examples. The experimental results amply demonstrate the capability of IPSO

toward obtaining the best objective function values in all the cases. Compared with the other four PSO approaches, IPSO has stronger convergence and higher stability which clearly points out its desirable performance in search accuracy and identifying efficiency.

Keywords Improved particle swarm optimization · IIR system identification · Golden ratio · Inertia weight · Normal distribution

1 Introduction

Many researchers and scholars are exhibiting increasingly greater interest in adaptive infinitive impulse response (IIR) filtering during the last few decades [1, 2]. This is getting enhanced by the fact that a variety of application areas, such as speech recognition, acoustics, and communications, strongly depend on adaptive signal processing. For some system identification problems, the adaptive IIR filter attempts to characterize the unknown system according to some function of the error between the output of the adaptive filter and the output of the plant. In order to obtain a satisfactory identification result, it is necessary to find suitable filter coefficients so as to produce minimal error between the output of the adaptive IIR filter and the output of the plant.

The task of choosing suitable parameters for the IIR model is indeed a formidable one. After all, the constructions of the adaptive IIR filter are itself complicated and the same gets compounded when the environment is disturbed. In order to address the IIR system identification problem, a variety of efficient approaches have been developed in recent years. The artificial bee colony (ABC) algorithm [3] is a simple, robust, and flexible optimization technique. Armed

✉ Gai-Ge Wang
gaigewang@163.com; gaigewang@gmail.com;
gaige@ualberta.ca

¹ School of Electrical Engineering and Automation, Jiangsu Normal University, Xuzhou, China

² IT & Educational Consultant, Ranchi, Jharkhand, India

³ School of Computer Science and Technology, Jiangsu Normal University, Xuzhou, China

⁴ Department of Electrical and Computer Engineering, University of Alberta, Edmonton AB T6R 2V4, Canada

⁵ Institute of Algorithm and Big Data Analysis, Northeast Normal University, Changchun 130117, China

⁶ School of Computer Science and Information Technology, Northeast Normal University, Changchun 130117, China

with these advantages, the ABC has been turned into a powerful tool for solving many global optimization problems. Recently, Karaboga [4] designed a new approach based on the ABC algorithm for low- and high-order digital IIR filters. The corresponding simulation results clearly demonstrate the effectiveness of this new methodology toward designing digital low- and high-order IIR filters. Luitel et al. [5] used a particle swarm optimization with quantum infusion (PSO-QI) [6] for IIR system identification. According to the identification results of benchmark IIR systems with full- and reduced-order models, the PSO-QI can obtain lower mean squared error and more consistent convergence than the other algorithms. Dai et al. [7] used a seeker optimization algorithm (SOA) toward designing digital IIR filter. SOA is based upon simulation of the behavior of the process of human searching where the search direction stems from the empirical gradient by evaluating the response to the position changes and the step length stems from uncertainty reasoning by using a simple fuzzy rule. The simulation results exhibit the efficiency of the SOA toward designing IIR filter. Panda et al. [8] formulated the IIR system identification task as an optimization problem and employed a cat swarm optimization (CSO) algorithm [9] in order to establish an adaptive learning rule for the model. Both actual and reduced-order identification of few benchmarked IIR plants are conducted by simulation study. Experimental results demonstrate the potential of the CSO toward identification of the IIR plant as compared to the other algorithms. Upadhyay et al. [10] proposed a craziness-based particle swarm optimization (CRPSO) algorithm for IIR system identification problem. The CRPSO utilizes some random variables to enhance the process of exploration and exploitation [65, 66] in multidimensional search space. Furthermore, craziness factor is introduced into the velocity updating of the particle swarm optimization (PSO) algorithm [11, 12]. As a result, the diversity of population is ensured and the convergence is improved. In addition to these, many adaptive system identification methods have been reported in the literatures [13–17].

The remainder of this paper is organized as follows. In Sect. 2, the adaptive IIR filter model and several system data are introduced, while Sect. 3 reports four PSO algorithms. In Sect. 4, the proposed version is presented. In Sect. 5, five PSOs are used to solve 12 IIR system identification problems. Finally, the paper ends after presenting the conclusions.

2 Adaptive IIR filter model

The goal of IIR system identification is to turn the coefficients of the adaptive IIR filter by adaptive algorithms. The purpose was to bring the filter's output closer to the output

of unknown system when the same input signal is applied simultaneously to both, unknown plant and adaptive filter. The block diagram of an adaptive IIR system identification is shown in Fig. 1.

This section studies the design method of adaptive IIR filter. The input–output relation can be described in terms of the following difference equation [10, 18]:

$$\sum_{l=0}^L a_l y(n-l) = \sum_{k=0}^K b_k x(n-k) \quad (1)$$

where $x(n)$ and $y(n)$ denote the filter's input and output, respectively, and $L(\geq K)$ is the filter's order. Suppose $a_0 = 1$, the transfer function of the adaptive IIR filter is given by

$$H(z) = \frac{\sum_{k=0}^K b_k z^{-k}}{1 + \sum_{l=1}^L a_l z^{-l}} \quad (2)$$

In the design method, the adaptive IIR filter $H_a(z)$ is utilized to identify the unknown plant of transfer function $H_u(z)$ so that the output of the unknown IIR system is well matched by the output of the adaptive IIR filter. Moreover, mean square error (MSE) of time samples is viewed as the objective function, and it is given by

$$\text{MSE} = J = \frac{1}{N_s} \sum_{n=1}^{N_s} e^2(n) \quad (3)$$

Additionally, the dB form of the mean square error (MSE) is given by

$$\text{MSE}(\text{dB}) = 10 \log_{10}(J) \quad (4)$$

where N_s is the number of time samples; $e(n) = d(n) - y(n)$ stands for the error signal; $y(n)$ stands for the response of the adaptive IIR filter; $d(n) = y_0(n) + v(n)$ stands for the overall response of the unknown IIR plant; $y_0(n)$ denotes the output of the unknown IIR plant; $v(n)$ denotes an additive white Gaussian noise. The task of adaptive algorithm is to minimize MSE by turning the coefficient vector Θ of the transfer function $H_a(z)$. Here,

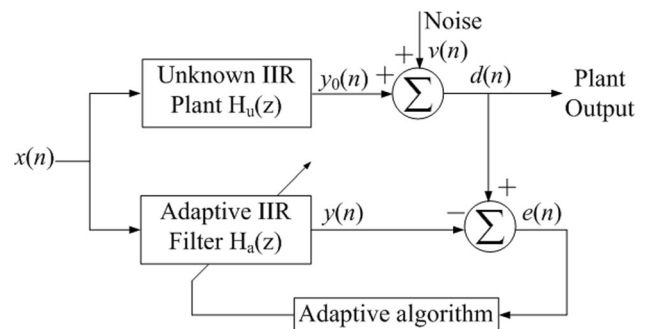


Fig. 1 Adaptive algorithm for IIR system identification

$$\Theta = (a_1, \dots, a_L, b_0, b_1, \dots, b_K)^T \tag{5}$$

3 Particle swarm optimization-based algorithms

Particle swarm optimization (PSO) algorithm [11, 12, 19–22] is one of the most popular evolutionary algorithms, and it is a very simple but efficient optimization technique for solving global optimization problems. Due to its simplicity and convenience, the PSO has been applied in many real-world applications, such as odor source localization [23], aggregate production planning [24], switching instant identification [25], defense against SYN flooding attacks [26], and designing fuzzy logic controllers [27].

3.1 PSO algorithm

In order to better understand the working principle of PSO, a simple flowchart of PSO is provided as follows (Fig. 2).

The PSO takes advantage of previous velocity vectors and position vectors to generate current velocity vectors and position vectors for the particles in the population. In particular, each particle makes use of its own experience

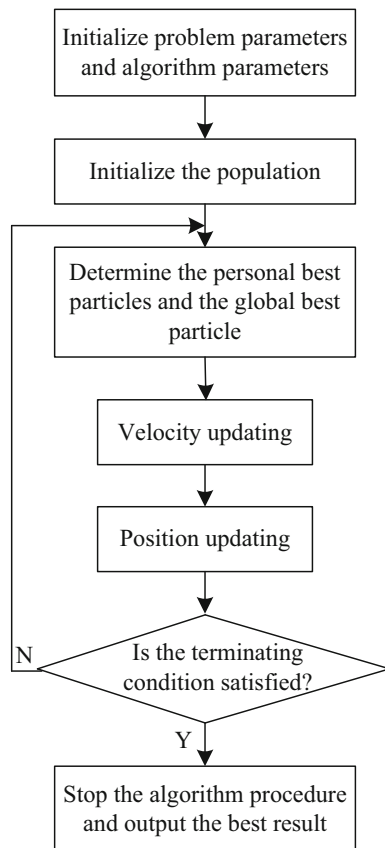


Fig. 2 Flowchart of particle swarm optimization

and the most successful particle’s experience to adjust its position. For each individual, its velocity and position can be updated as given by Eqs. (6), (7), respectively:

$$v_{ij}^{k+1} = \omega v_{ij}^k + c_1 r_1 (p_{ij} - x_{ij}^k) + c_2 r_2 (g_j - x_{ij}^k) \tag{6}$$

$$x_{ij}^{k+1} = x_{ij}^k + v_{ij}^{k+1} \tag{7}$$

Here, ω represents inertia weight, and it decreases linearly during the iteration process. v_{ij}^k and x_{ij}^k are, respectively, the j th velocity variable and the j th position variable of particle i at generation k . v_{ij}^{k+1} and x_{ij}^{k+1} are, respectively, the j th velocity variable and the j th position variable of particle i at generation $k + 1$. p_{ij} denotes the j th variable of the personal best position of particle i , while g_j signifies the j th variable of the global best position of the population. c_1 is defined as cognitive factor, while c_2 represents social factor. r_1 and r_2 denote the uniformly generated random numbers in the range $[0, 1]$.

3.2 Crazyiness-based particle swarm optimization algorithm

In [28–30], the authors modified Eq. (6) using several random numbers and a “craziness velocity” with a predefined probability is executed before the process of position updating. This PSO variant is described as crazyiness-based particle swarm optimization (CRPSO). The new velocity updating equation is given by

$$V_i^{k+1} = r_2 * \text{sign}(r_3) * V_i^k + (1 - r_2) * C_1 * r_1 * \{pbest_i^{(k)} - S_i^k\} + (1 - r_2) * C_2 * (1 - r_1) * \{gbest_i^{(k)} - S_i^k\} \tag{8}$$

where r_1, r_2 , and r_3 are the three different random numbers which are uniformly generated in the interval $[0,1]$, and $\text{sign}(r_3)$ is determined according to the following equation:

$$\text{sign}(r_3) = \begin{cases} -1, & r_3 \leq 0.05; \\ 1, & \text{Otherwise.} \end{cases} \tag{9}$$

In addition, the CRPSO introduces a craziness operator, intending to maintain the diversity of the particles, and it can be expressed as follows:

$$V_i^{k+1} = V_i^{k+1} + P(r_4) * \text{sign}(r_4) * v_i^{\text{craziness}} \tag{10}$$

where r_4 is a random number generated uniformly from the interval $[0,1]$; $v_i^{\text{craziness}}$ is a random number generated uniformly from the interval $[v_i^{\text{min}}, v_i^{\text{max}}]$; and $P(r_4)$ and $\text{sign}(r_4)$ are defined, respectively, as

$$P(r_4) = \begin{cases} 1, & \text{If } r_4 \leq P_{cr}; \\ 0, & \text{Otherwise.} \end{cases} \tag{11}$$

where P_{cr} is a predefined probability of craziness.

$$\text{sign}(r_4) = \begin{cases} -1, & \text{If } r_4 \geq 0.5; \\ 1, & \text{Otherwise.} \end{cases} \quad (12)$$

$v_i^{\text{craziness}}$ is set very small (=0.0001). $r_4 \geq 0.5$ (or <0.5) aims to use equal probability of direction reversal of $v_i^{\text{craziness}}$.

3.3 A global particle swarm optimization algorithm

In [31], the authors modified Eq. (6) by introducing a new inertia weight and adding small disturbance to the global optimal solution. This PSO variant is described as global particle swarm optimization (GPSO). The new velocity updating equation is given by:

$$v_{ij}^{t+1} = \omega(t) \times v_{ij}^t + c_1 r_1 (p_{ij} - x_{ij}^t) + c_2 r_2 [p_{gj}(1 \pm \delta \times U(0, 1)) - x_{ij}^t] \quad (13)$$

Here, $\omega(t)$ represents the inertia weight at generation t , and it is given by:

$$\omega(t) = a \times \exp(b \times t^2) \times R \quad (14)$$

where $b = \ln\{\omega_{\max}/\omega_{\min}\}/(T^2 - 1)$, and $a = \omega_{\max} \times \exp(-b)$. Additionally, $U(0,1)$ is a random number generated uniformly from the interval $[0,1]$; ω_{\max} (ω_{\min}) represents the maximal (minimal) inertia weight; δ is termed as disturbance factor. By using the above velocity updating equation, the GPSO can make a good balance between global search and local search.

3.4 A particle swarm optimization with time-varying acceleration coefficients

In [32, 33], the authors modified Eq. (6) by introducing two dynamical acceleration coefficients. Especially, they adjust these coefficients in a way that the cognitive component is reduced and social component is increased as iteration proceeds. This PSO variant is described as particle swarm optimization with time-varying acceleration coefficients (TVAC-PSO). The novel velocity updating equation is stated as follows:

$$v_{in}^{iter+1} = \omega \times v_{in}^{iter} + \left(C_{1i} + \frac{C_{1f} - C_{1i}}{iter_{\max}} \right) \times r_1^n \times (p_{pbest_{in}}^{iter} - x_{in}^{iter}) + \left(C_{2i} + \frac{C_{2f} - C_{2i}}{iter_{\max}} \right) \times r_2^n \times (g_{best}^{iter} - x_{in}^{iter}) \quad (15)$$

where parameters C_{1i} and C_{1f} (C_{2i} and C_{2f}) represent initial and final values of cognitive (social) acceleration coefficients, respectively. Moreover, inertia weight ω is given by:

$$\omega = \omega_{\max} - \frac{\omega_{\max} - \omega_{\min}}{iter_{\max}} \times iter \quad (16)$$

ω_{\max} and ω_{\min} denote maximal and minimal inertia weights, respectively. $iter$ stands for the maximum iteration number. In addition, the position updating equation is slightly modified as:

$$x_{in}^{iter+1} = x_{in}^{iter} + C \times v_{in}^{iter+1} \quad (17)$$

Parameter C is the constriction factor and can be calculated by using the following equation:

$$C = \frac{2}{2 - \varphi - \sqrt{\varphi^2 - 4\varphi}} \quad (18)$$

where φ is set to 4.1. By using the above updating equation, the TVAC-PSO can improve solution quality and avoid premature convergence.

In addition to PSO, there are many other swarm intelligence-based algorithms that can be applied to IIR system identification problems, such as harmony search algorithm [34, 35], differential evolution [36], genetic algorithm [37], cuckoo search [38–40], earthworm optimization algorithm (EWA) [41], and elephant herding optimization (EHO) [42, 43], monarch butterfly optimization [44, 45], bat algorithm [46, 47], firefly algorithm [48], and krill herd algorithm [49–52]. Due to space limitations, we only concentrate on the application of several PSO algorithms to IIR system identification problems.

4 Improved particle swarm optimization

As already pointed out, this paper presents an improved particle swarm optimization algorithm (IPSO) for solving IIR system identification problems. In detail, IPSO and PSO differ in three aspects, which are described below:

4.1 Population initialization

Golden ratio is a very mysterious and interesting issue, and it has found many application areas, such as the human heart [53], n -body problem [54], computation of a face attractiveness index [55], and facial beauty [56]. Due to its simplicity and usefulness, golden ratio is employed to segment the solution space:

First, determine M values for the j th ($j = 1, 2, \dots, N$) dimension of all M particles, and the i th ($i = 1, 2, \dots, M$) value is calculated according to the following equation:

$$\alpha_{i,j} = U_j - 0.618^i \times (U_j - L_j) \quad (19)$$

where U_j and L_j represent the upper and lower bounds, respectively, for the j th dimension. After calculating M values, they are randomly assigned to the j th dimension of M particles. Let $x_{i,j}$ be the j th ($j = 1, 2, \dots, N$) dimension of the i th ($i = 1, 2, \dots, M$) particle, and then it can be determined by $x_{i,j} = \alpha_{i',j}$. It should be noted that the index i' is a

random integer in the range $[1, M]$. By using the above initialization method, a population P_1 is produced.

Second, calculate M values for the j th ($j = 1, 2, \dots, N$) dimension of all M particles, and the i th ($i = 1, 2, \dots, M$) value is obtained according to the following equation:

$$\beta_{i,j} = L_j + 0.618^i \times (U_j - L_j) \tag{20}$$

After calculating M values, they are randomly assigned to the j th dimension of the other M particles. Thus, $x_{i,j}$ is determined by $x_{i,j} = \beta_{i,j}$. By using the above initialization method, a population P_2 is produced.

Third, select M best particles from $P_1 \cup P_2$ for the initial population P . In population P , any solution vector $x_i = (x_{i,1}, x_{i,2}, \dots, x_{i,N})$ ($i = 1, 2, \dots, M$) stands for a candidate solution of the parameters [as Eq. (5)] used for implementing the IIR system identification. Besides, the length of solution vector x_i is equal to N , which is exactly the number of the parameters in coefficient vector Θ . By using this novel approach of population initialization, the quality of solutions and convergence of the IPSO can be improved simultaneously.

4.2 Global disturbance

In light of Eqs. (6), (7), the global best particle is a potential candidate solution and it facilitates rapid movement of all the particles toward the neighbors. However, this tendency may lead to trapping into the local optima. As a result, these particles will lose plenty of chances of searching very large spaces of candidate solutions. In order to help particles to easily get rid of the local optima, a global disturbance strategy is utilized to update the global best particle in each generation, and it can be expressed as follows:

$$g'_j = g_j \times (1 + \lambda \times (N(0, 1))) \tag{21}$$

Here, g_j represents the j th dimension of the global best particle \mathbf{g} ; parameter λ is defined as disturbance factor. We have tried different values (i.e., 0.01, 0.02, 0.05, 0.1, 0.15, and 0.2) for λ , but we did not notice obvious difference in experimental results. Thus, it is set to 0.1 in this paper. In addition, $N(0,1)$ denotes the normal distribution with mean 0 and variance 1. It should be emphasized that the global best particle \mathbf{g} will be replaced with \mathbf{g}' only when \mathbf{g}' is superior to \mathbf{g} . By using the above updating strategy, the capacity of escaping from the local optimums can be further enhanced for IPSO.

4.3 Randomly assigned inertia weights

Inertia weight ω is responsible for iteratively adjusting the velocities of all particles, and different values of ω will

have distinct effects on the scope of particles' search. To be more specific, a large value of ω contributes to the global search, while a small value is helpful to the local search. With regard to the traditional PSO algorithm, it has only one value of inertia weight at each generation, which can hardly give consideration to both global search and local search. To overcome this shortage, M different inertia weights are generated in terms of the following equation:

$$\omega_i = i \times \Delta\omega \tag{22}$$

Here, i ($i = 1, 2, \dots, M$) is the index of particle, $\Delta\omega$ is equal to $1/M$. After generating these M different inertia weights, they are randomly assigned to M particles at each generation. Based on this method, the velocity updating equation can be stated as follows:

$$v_{i,j}^{k+1} = \omega_i v_{i,j}^k + c_1 r_1 (p_{i,j} - x_{i,j}^k) + c_2 r_2 (g_j - x_{i,j}^k) \tag{23}$$

where $p_{i,j}$ stands for the j th dimension of personal best particle \mathbf{p}_i . ω_i denotes a randomly selected inertia weight from ω_i ($i = 1, 2, \dots, M$) for particle i , and each particle adopts a different inertia weight from the others. At each generation, the original PSO algorithm adopts only one inertia weight. Unlike original PSO, the IPSO uses M values of inertia weight. Furthermore, as mentioned earlier, the large values of ω_i are beneficial to global search, and the small values of ω_i are useful for carrying out local search. Clearly, the IPSO algorithm is able to keep better balance between the global search and the local search, as compared to that of the PSO algorithm.

In addition to these three improvements, as resulted by the application of the IPSO algorithm, we provide a more detailed IPSO steps which is presented in Table 1.

In computer science, the traditional PSO algorithm is a swarm intelligence-based approach which optimizes a problem by gradually trying to improve candidate solutions so as to meet the requirements of this problem. As a variant of PSO, IPSO optimizes the infinite impulse response system identification problem by iteratively updating a population of candidate solutions namely particles and disturbing them in the solution space according to the mathematical model of adaptive IIR filter. More specifically, the mean square error is an important criterion that gains confidence in, or reject the postulated mathematical model based on the final parameters optimized by the IPSO algorithm. If the mean square error reaches quite a small value, the outputs of the adaptive IIR filter will be very close to the outputs of the actual system outputs. Thus, it can be safely claimed that the postulated model is reasonable and accurate. On the other hand, if the mean square error reaches a large value, the outputs of the adaptive IIR filter will be far away from the outputs of the actual

Table 1 Procedure of IPSO

Line	Pseudocode of IPSO
1	Set the number of time samples $N_s = 100$; the feedback filter order L ; the feed forward filter order K
2	Set the number of variables $N = L + K + 1$; population size $M = 40$; the number of generations $N_G = 1000$
3	Set $\Delta\omega = 1/M$; $\lambda = 0.1$; $x_{Lj} = -1000$, $x_{Uj} = 1000$, $v_{Lj} = -1000$, $v_{Uj} = 1000$, $j = 1, \dots, N$
4	Generate two swarms P_1 and P_2 according to golden ratio
5	Select M best particles from $P_1 \cup P_2$ for the initial population P
6	Produce M different inertia weights
7	For $t = 1$ to N_G
8	Calculate the objective function values of all particles according to the mean square error
9	Disturb the global best particle according to normal distribution, and update g if g' is better than g
10	Randomly assign M different inertia weights to M particles.
11	For $i = 1$ to M
12	If $f(x_i) < f(p_i)$
13	$p_i = x_i$. % Update the personal best particle
14	If $f(p_i) < f(g)$
15	$g = p_i$. % Update the global best particle
16	For $j = 1$ to N
17	Update the j th velocity variable of particle i using velocity updating equation based on modified inertia weights
18	Update the j th position variable of particle i using traditional position updating equation
19	Output the best results

N_G denotes the total number of generations for adjusting the parameters associated with the IIR system identification. In each generation, any new updated candidate solution will be accepted if it is superior to the solution before updating

Table 2 Twelve IIR system identification problems

Problems	The transfer function of IIR plant $H_u(z)$	The transfer function of the adaptive IIR filter model $H_a(z)$
Example I (Case 1) [2, 8, 10]	$\frac{0.1084+0.5419z^{-1}+1.0837z^{-2}+1.0837z^{-3}+0.5419z^{-4}+0.1084z^{-5}}{1+0.9853z^{-1}+0.9738z^{-2}+0.3864z^{-3}+0.1112z^{-4}+0.0133z^{-5}}$	$\frac{b_0+b_1z^{-1}+b_2z^{-2}+b_3z^{-3}+b_4z^{-4}+b_5z^{-5}}{1-a_1z^{-1}-a_2z^{-2}-a_3z^{-3}-a_4z^{-4}-a_5z^{-5}}$
Example I (Case 2) [8, 10]	$\frac{0.1084+0.5419z^{-1}+1.0837z^{-2}+1.0837z^{-3}+0.5419z^{-4}+0.1084z^{-5}}{1+0.9853z^{-1}+0.9738z^{-2}+0.3864z^{-3}+0.1112z^{-4}+0.0133z^{-5}}$	$\frac{b'_0+b'_1z^{-1}+b'_2z^{-2}+b'_3z^{-3}+b'_4z^{-4}}{1-a'_1z^{-1}-a'_2z^{-2}-a'_3z^{-3}-a'_4z^{-4}}$
Example II (Case 1) [8, 10, 57]	$\frac{1-0.9z^{-1}+0.81z^{-2}-0.729z^{-3}}{1+0.04z^{-1}+0.2775z^{-2}-0.2101z^{-3}+0.14z^{-4}}$	$\frac{b_0+b_1z^{-1}+b_2z^{-2}+b_3z^{-3}}{1-a_1z^{-1}-a_2z^{-2}-a_3z^{-3}-a_4z^{-4}}$
Example II (Case 2) [8, 10]	$\frac{1-0.9z^{-1}+0.81z^{-2}-0.729z^{-3}}{1+0.04z^{-1}+0.2775z^{-2}-0.2101z^{-3}+0.14z^{-4}}$	$\frac{b'_0+b'_1z^{-1}+b'_2z^{-2}}{1-a'_1z^{-1}-a'_2z^{-2}-a'_3z^{-3}}$
Example III (Case 1) [4, 7, 8, 10]	$\frac{0.05-0.4z^{-1}}{1-1.131z^{-1}+0.25z^{-2}}$	$\frac{b_0+b_1z^{-1}}{1-a_1z^{-1}-a_2z^{-2}}$
Example III (Case 2) [4, 7, 8, 10]	$\frac{0.05-0.4z^{-1}}{1-1.131z^{-1}+0.25z^{-2}}$	$\frac{b'_0}{1-a'_1z^{-1}}$
Example IV (Case 1) [5, 8, 10]	$\frac{-0.2-0.4z^{-1}+0.5z^{-2}}{1-0.6z^{-1}+0.25z^{-2}-0.2z^{-3}}$	$\frac{b_0+b_1z^{-1}+b_2z^{-2}}{1-a_1z^{-1}-a_2z^{-2}-a_3z^{-3}}$
Example IV (Case 2) [5, 8, 10]	$\frac{-0.2-0.4z^{-1}+0.5z^{-2}}{1-0.6z^{-1}+0.25z^{-2}-0.2z^{-3}}$	$\frac{b'_0+b'_1z^{-1}}{1-a'_1z^{-1}-a'_2z^{-2}}$
Example V [4, 5, 10]	$\frac{1-0.4z^{-2}-0.65z^{-4}+0.26z^{-6}}{1-0.77z^{-2}-0.8498z^{-4}+0.6486z^{-6}}$	$\frac{b_0+b_2z^{-2}+b_4z^{-4}+b_6z^{-6}}{1-a_2z^{-2}-a_4z^{-4}-a_6z^{-6}}$
Example VI [4, 10, 58]	$\frac{1}{1-1.2z^{-1}+0.6z^{-2}}$	$\frac{b_0}{1-a_1z^{-1}-a_2z^{-2}}$
Example VII [10, 59]	$\frac{1}{(1-0.5z^{-1})^3}$	$\frac{1}{1-a_1z^{-1}-a_2z^{-2}}$
Example VIII [2, 6, 10, 59]	$\frac{1.25z^{-1}-0.25z^{-2}}{1-0.3z^{-1}+0.4z^{-2}}$	$\frac{b_1z^{-1}+b_2z^{-2}}{1-a_1z^{-1}-a_2z^{-2}}$

system outputs, indicating that the postulated model is unreasonable and inaccurate. This paper aims to utilize the IPSO algorithm to optimize the parameters of the adaptive IIR filter in order to validate the postulated mathematical model.

5 Experimental results and analysis

This section aims to investigate the improvement made by the proposed population initialization, global disturbance, and dynamical inertia weight on solving the IIR system

Table 3 Parameters of PSO, CRPSO, GPSO, TVAC-PSO, and IPSO

Algorithm	P_S	N_G	Inertia weight	Acceleration coefficients	Velocity	Other parameters
PSO	40	1000	$\omega_{\max} = 0.9, \omega_{\min} = 0.4$	$c_1 = 2, c_2 = 2$	$v_{\max} = 1000$	–
CRPSO	40	1000	–	$c_1 = 2.05, c_2 = 2.05$	$v_{\max} = 1000$	$P_{cr} = 0.3$
GPSO	40	1000	$\omega_{\max} = 0.9, \omega_{\min} = 0.4$	$c_1 = 2, c_2 = 2$	$v_{\max} = 1000$	$\delta = 0.01$
TVAC-PSO	40	1000	$\omega_{\max} = 0.9, \omega_{\min} = 0.4$	$c_{1i} = 2.5, c_{1f} = 0.5, c_{2i} = 0.5, c_{2f} = 2.5,$	$v_{\max} = 0.4, x_{Uj} = 400$	$\phi = 4.1$
IPSO	40	1000	$\omega_{\max} = 0.9, \omega_{\min} = 0.4$	$c_1 = 2, c_2 = 2$	$v_{\max} = 1000$	$\lambda = 0.1$

identification problems. For this purpose, we compared the performance of the following five PSO algorithms: the original PSO [11, 12], the CRPSO [28–30], the GPSO [31], the TVAC-PSO [32, 33], and our proposed IPSO. Twelve problems are recorded in Table 2. In order to understand the difference in the problems for algorithm validation, we will describe and discuss these problems as follows: Example I (Case 1) and Example I (Case 2) share the same unknown plant that is inherently a fifth-order low-pass Butterworth filter, but they employ different adaptive filters to identify this plant. To be more specific, the first adaptive filter has the same order as that of the unknown plant, and the order of the second adaptive filter is four which is slightly lower than that of the unknown plant. Obviously, the first filter is more accurate than the second one, but it introduces more parameters needed for solving IIR system identification, hence it increases the computational cost. However, the order of the adaptive filter should not be far from that of the unknown plant; otherwise, it may lead to serious mean square error. The same observations apply to the other cases. In addition, the problem parameters are set as follows: the number of time samples $N_s = 100$; the variables’ upper bound $x_{Uj} = 1000$ and lower bound $x_{Lj} = -1000$; the input common to both the unknown plant and the identifying IIR filter is the randomly generated Gaussian white noise signal with mean 0 and standard deviation 1; the additive noise is a Gaussian white signal with mean 0 and standard deviation 10^{-3} .

The algorithm parameters are displayed in Table 3. Here, “ P_S ” represents population size, N_G represents the number of generations. MATLAB 7.0 was used to execute the above design steps under the environment of Intel(R) Core(TM) i5-2410M CPU @ 2.30 GHz. Thirty independent runs were performed for each problem, and the optimization results of each of these are listed in Table 4.

Table 4 compares the five PSO approaches on the IIR system identification problems with different feedback filter orders and feed forward filter orders. Here, the term “AET” stands for average execution time, and the term “Std” stands for standard deviation. Additionally, the best results are highlighted in bold. According to Table 4, the IPSO algorithm outperforms the other four PSO

approaches on most IIR system identification problems. Especially for Example I (Case 1) and Example II (Case 1), IPSO can obtain the best results according to the five criteria “Best,” “Worst,” “Mean,” “Median,” and “Std.” For Example III (Case 1), Example III (Case 2), Example IV (Case 1), Example IV (Case 2), Example VI, Example VII, and Example VIII, IPSO finds their global optimal solutions in each of the 30 runs. This indicates that the new global disturbance strategy can offer much more accurate solutions for IIR system identification problems, which mainly benefits from its strong exploitation ability. With regard to Example V, IPSO finds the second best solution which is slightly worse than that of CRPSO, but better than those attained by the other three PSO approaches. Moreover, IPSO can obtain the best results in terms of the other four criteria involving “Worst,” “Mean,” “Median,” and “Std.” For Example I (Case 2), although the best solution obtained by IPSO is not as good as that of TVAC-PSO, the solutions it generates are all acceptable. For Example II (Case 2), three PSO approaches, viz., CRPSO, TVAC-PSO, and IPSO, can find the best solution which is slightly better than those of the other two approaches. A careful observation from Table 4 reveals that IPSO has the best convergence among the five approaches. The reason lies in its capability of attaining the smallest values of “Mean” for all the 12 problems. Furthermore, it can attain the smallest values of “Std” for ten out of 12 problems, which signifies its high reliability. Additional executing steps lead to increased computing time for IPSO, and it uses more average executing times as compared with the other four PSO approaches for six problems according to the term “AET”. Nevertheless, there is no significant difference regarding the average executing times of five PSO approaches. By and large, the comparison among five PSOs is fair and acceptable.

To learn more about the evolution processes of different PSO approaches, Fig. 3 plots the average optimization curves of five PSO approaches in optimizing 12 IIR system identification problems.

As can be seen from Fig. 3, the IPSO algorithm generates better solutions for most IIR system identification problems in initial evolution progress. This indicates that

Table 4 Comparison of PSO, CRPSO, GPSO, TVAC-PSO, and IPSO on 12 IIR system identification problems

Problem	Algorithm	AET(s)	Best	Worst	Mean	Median	Std
Example I (Case 1)	PSO	1.7031	3.1087e + 001	1.1683e + 003	2.2689e + 002	6.2253e + 001	3.8166e + 002
	CRPSO	1.6983	-2.4906e + 001	1.0299e + 003	2.5852e + 002	5.9165e + 001	3.4514e + 002
	GPSO	1.7021	-3.0789e + 001	5.6210e + 002	1.5058e + 002	4.7680e + 001	2.0751e + 002
	TVAC-PSO	1.6769	-3.3954e + 001	4.4930e + 002	9.3305e + 001	4.4749e + 001	1.3248e + 002
	IPSO	1.7364	-5.3741e + 001	-3.8482e + 001	-4.4715e + 001	-4.4038e + 001	3.7665e + 000
Example I (Case 2)	PSO	1.6257	4.8972e + 001	6.4502e + 001	5.8651e + 001	5.9681e + 001	3.7262e + 000
	CRPSO	1.6150	-4.0506e + 001	1.0551e + 003	8.6110e + 001	4.4333e + 001	2.1715e + 002
	GPSO	1.6083	-3.7016e + 001	6.4256e + 001	2.2467e + 000	-2.7375e + 001	3.8644e + 001
	TVAC-PSO	1.6099	-4.6217e + 001	3.2544e + 002	1.6389e + 001	-1.0256e + 001	7.1473e + 001
	IPSO	1.6480	-3.5592e + 001	-3.0224e + 001	-3.3560e + 001	-3.3617e + 001	1.0846e + 000
Example II (Case 1)	PSO	2.3832	-3.0109e + 001	1.4393e + 003	1.0187e + 002	5.9511e + 001	2.5312e + 002
	CRPSO	2.4016	-5.9877e + 001	7.0831e + 002	6.6121e + 001	1.5800e + 001	2.0150e + 002
	GPSO	2.3957	-3.7538e + 001	1.8548e + 002	-4.8401e + 000	-2.6268e + 001	4.7315e + 001
	TVAC-PSO	2.3559	-6.0049e + 001	4.7228e + 001	-1.7785e + 001	-1.1880e + 001	2.8503e + 001
	IPSO	2.4474	-6.0164e + 001	-2.7259e + 001	-5.1965e + 001	-5.2910e + 001	6.9944e + 000
Example II (Case 2)	PSO	2.2957	-1.1309e + 001	6.4487e + 001	3.5951e + 001	5.4629e + 001	3.0851e + 001
	CRPSO	2.2742	-1.1310e + 001	5.9634e + 001	-7.6463e - 001	-1.1310e + 001	2.3674e + 001
	GPSO	2.2729	-1.1303e + 001	-6.6007e + 000	-1.0100e + 001	-1.1173e + 001	1.9645e + 000
	TVAC-PSO	2.2731	-1.1310e + 001	2.2186e + 000	-9.0110e + 000	-1.1310e + 001	3.7050e + 000
	IPSO	2.3299	-1.1310e + 001	-6.6099e + 000	-1.0286e + 001	-1.1301e + 001	1.8814e + 000
Example III (Case 1)	PSO	2.2559	-5.9840e + 001	-2.0306e + 000	-1.0464e + 001	-2.0306e + 000	1.9924e + 001
	CRPSO	2.1457	-5.9839e + 001	-2.0306e + 000	-5.0203e + 001	-5.9838e + 001	2.1912e + 001
	GPSO	2.1481	-5.8312e + 001	-2.0306e + 000	-4.9639e + 001	-5.4565e + 001	1.6223e + 001
	TVAC-PSO	2.1181	-5.9840e + 001	-5.9840e + 001	-5.9840e + 001	-5.9840e + 001	6.0097e - 013
	IPSO	2.1844	-5.9840e + 001	-5.9840e + 001	-5.9840e + 001	-5.9840e + 001	2.1276e - 010
Example III (Case 2)	PSO	0.3684	-6.4540e + 000	-6.4540e + 000	-6.4540e + 000	-6.4540e + 000	2.7892e - 015
	CRPSO	0.2766	-6.4540e + 000	-6.4540e + 000	-6.4540e + 000	-6.4540e + 000	4.6228e - 011
	GPSO	0.2762	-6.4540e + 000	-6.4540e + 000	-6.4540e + 000	-6.4540e + 000	1.8177e - 006
	TVAC-PSO	0.2515	-6.4540e + 000	-6.4540e + 000	-6.4540e + 000	-6.4540e + 000	2.4740e - 015
	IPSO	0.2741	-6.4540e + 000	-6.4540e + 000	-6.4540e + 000	-6.4540e + 000	2.6645e - 015
Example IV (Case 1)	PSO	2.1870	-6.0038e + 001	5.8715e + 001	1.6615e + 001	5.2618e + 001	5.2232e + 001
	CRPSO	2.1716	-6.0037e + 001	5.6253e + 001	-5.2403e + 001	-6.0036e + 001	2.9048e + 001
	GPSO	2.1680	-5.7798e + 001	-2.2609e + 001	-4.9594e + 001	-5.3728e + 001	9.3060e + 000
	TVAC-PSO	2.1501	-6.0038e + 001	4.3409e + 001	-4.9256e + 001	-6.0038e + 001	2.5793e + 001
	IPSO	2.2262	-6.0038e + 001	-6.0038e + 001	-6.0038e + 001	-6.0038e + 001	2.5650e - 005
Example IV (Case 2)	PSO	2.2846	-2.0943e + 001	-4.6518e + 000	-5.7378e + 000	-4.6518e + 000	4.1331e + 000
	CRPSO	2.1711	-2.0943e + 001	-4.6518e + 000	-2.0400e + 001	-2.0943e + 001	2.9743e + 000
	GPSO	2.1791	-2.0943e + 001	-4.6518e + 000	-1.8197e + 001	-2.0942e + 001	6.1633e + 000
	TVAC-PSO	2.1498	-2.0943e + 001	-2.0943e + 001	-2.0943e + 001	-2.0943e + 001	3.7898e - 015
	IPSO	2.1982	-2.0943e + 001	-2.0943e + 001	-2.0943e + 001	-2.0943e + 001	3.6134e - 015
Example V	PSO	1.8942	-2.8143e + 001	6.4170e + 001	4.7024e + 001	5.7238e + 001	2.6663e + 001
	CRPSO	1.9277	-3.2180e + 001	5.7611e + 001	8.1697e + 000	-1.5615e + 001	3.7705e + 001
	GPSO	1.9220	-3.1153e + 001	4.9858e + 001	-6.9342e + 000	-1.6705e + 001	2.7055e + 001
	TVAC-PSO	1.9037	-3.1918e + 001	4.8341e + 001	-6.5120e - 001	-1.6000e + 001	2.8355e + 001
	IPSO	1.9653	-3.2050e + 001	1.3086e + 001	-1.9679e + 001	-1.8034e + 001	9.6962e + 000

Table 4 continued

Problem	Algorithm	AET(s)	Best	Worst	Mean	Median	Std
Example VI	PSO	0.4891	-6.0498e + 001	5.1357e + 000	-3.8834e + 001	-6.0498e + 001	3.0050e + 001
	CRPSO	0.4403	-6.0498e + 001	5.1357e + 000	-2.8720e + 001	-4.3141e + 001	3.2779e + 001
	GPSO	0.4339	-5.9602e + 001	5.1357e + 000	-2.0348e + 001	-7.5901e + 000	2.5080e + 001
	TVAC-PSO	0.3279	-6.0498e + 001	-6.0498e + 001	-6.0498e + 001	-6.0498e + 001	1.1285e - 013
	IPSO	0.3523	-6.0498e + 001	-6.0498e + 001	-6.0498e + 001	-6.0498e + 001	1.0484e - 013
Example VII	PSO	0.3462	-1.5350e + 001	-1.5350e + 001	-1.5350e + 001	-1.5350e + 001	6.0734e - 015
	CRPSO	0.3146	-1.5350e + 001	-1.5350e + 001	-1.5350e + 001	-1.5350e + 001	1.5473e - 009
	GPSO	0.3261	-1.5350e + 001	-1.5350e + 001	-1.5350e + 001	-1.5350e + 001	1.6202e - 004
	TVAC-PSO	0.3214	-1.5350e + 001	-1.5350e + 001	-1.5350e + 001	-1.5350e + 001	8.3579e - 015
	IPSO	0.3421	-1.5350e + 001	-1.5350e + 001	-1.5350e + 001	-1.5350e + 001	5.2881e - 015
Example VIII	PSO	1.4983	-6.0614e + 001	2.6652e + 000	-9.4205e + 000	2.6652e + 000	2.4755e + 001
	CRPSO	1.3980	-6.0614e + 001	2.6652e + 000	-5.6395e + 001	-6.0613e + 001	1.6054e + 001
	GPSO	1.3884	-5.8887e + 001	2.6652e + 000	-5.3104e + 001	-5.6892e + 001	1.5200e + 001
	TVAC-PSO	1.3614	-6.0614e + 001	-6.0614e + 001	-6.0614e + 001	-6.0614e + 001	8.1411e - 014
	IPSO	1.4146	-6.0614e + 001	-6.0614e + 001	-6.0614e + 001	-6.0614e + 001	4.9982e - 014

the novel population initialization strategy can provide high-quality solutions for IIR system identification problems. In addition, the curves of IPSO descend much faster than those of the other four PSO approaches, e.g., Example I (Case 1), Example I (Case 2), Example II (Case 1), Example IV (Case 1), and Example V, suggesting its better performance in search speed. Moreover, IPSO converges to the lowest levels for all the 12 problems, demonstrating its best search ability among the five PSO approaches. So far as Example III (Case 2) and Example VII are concerned, PSO can converge to the same levels as those of IPSO. Regarding Example III (Case 2), Example IV (Case 2), and Example VII, CRPSO can converge to the same levels as those of IPSO. With respect to Example II (Case 2), Example III (Case 2), and Example VII, GPSO can achieve the same levels as those of IPSO. With respect to Example III (Case 1), Example III (Case 2), Example IV (Case 2), Example VI, and Example VII, TVAC-PSO has the capability to achieve the same levels as those of IPSO. From the above observation and analysis, the IPSO algorithm has clearly demonstrated faster convergence rate and stronger stability than the other four PSO approaches on solving various IIR system identification problems.

The identifying results are visualized in order to testify the identifying performance of the IIR model based on IPSO, and Fig. 4 describes the comparison of actual plant outputs and IIR model outputs for 12 IIR system identification problems.

It is clear from Fig. 4 that IIR model outputs are very close to actual plant outputs in most cases, which indicates the high accuracy of IPSO. For Example II (Case 2), Example IV (Case 2), and Example VII, there are minor

errors between actual plant outputs and IIR model outputs. Regarding Example III (Case 2), the rough shape of IIR model outputs is identical to that of IIR model outputs, but there exist certain errors between these two kinds of outputs. In fact, the above four problems are all based on reduced-order IIR filter models. Although this kind of approximation model can reduce problem parameters and simplify calculation of complexity, it may result in non-negligible errors. To summarize, IPSO shows desirable performance for most IIR system identification problems. By combining suitable IIR model, it will play an important role in IIR system identification.

IIR system identification is a process of identifying or measuring the mathematical model of an infinitive impulse response system based on the measurements of the system inputs and outputs. In this paper, there are mainly four steps to be carried out for IIR system identification: (1) gather the outputs $d(n)$ of unknown system; (2) calculate the outputs $y(n)$ of the adaptive IIR filter according to the postulated model; (3) use the IPSO algorithm to adjust the parameters of the adaptive IIR filter $\Theta = (a_1, \dots, a_L, b_0, b_1, \dots, b_K)^T$; step (4) model validation. Step (1) is known as a necessary and basic part in the process of IIR system identification, and the outputs $d(n)$ are composed of the useful signals and noise. Step (2) is used to calculate the outputs $y(n)$ based on the current parameters of the adaptive IIR filter. Also, the more accurate these parameters are, the smaller mean square error (MSE) is. Step (3) aims to find the smallest mean square error so as to find the most suitable parameters of the adaptive IIR filter. After a number of generations, all these three steps are stopped, and the best results are given.

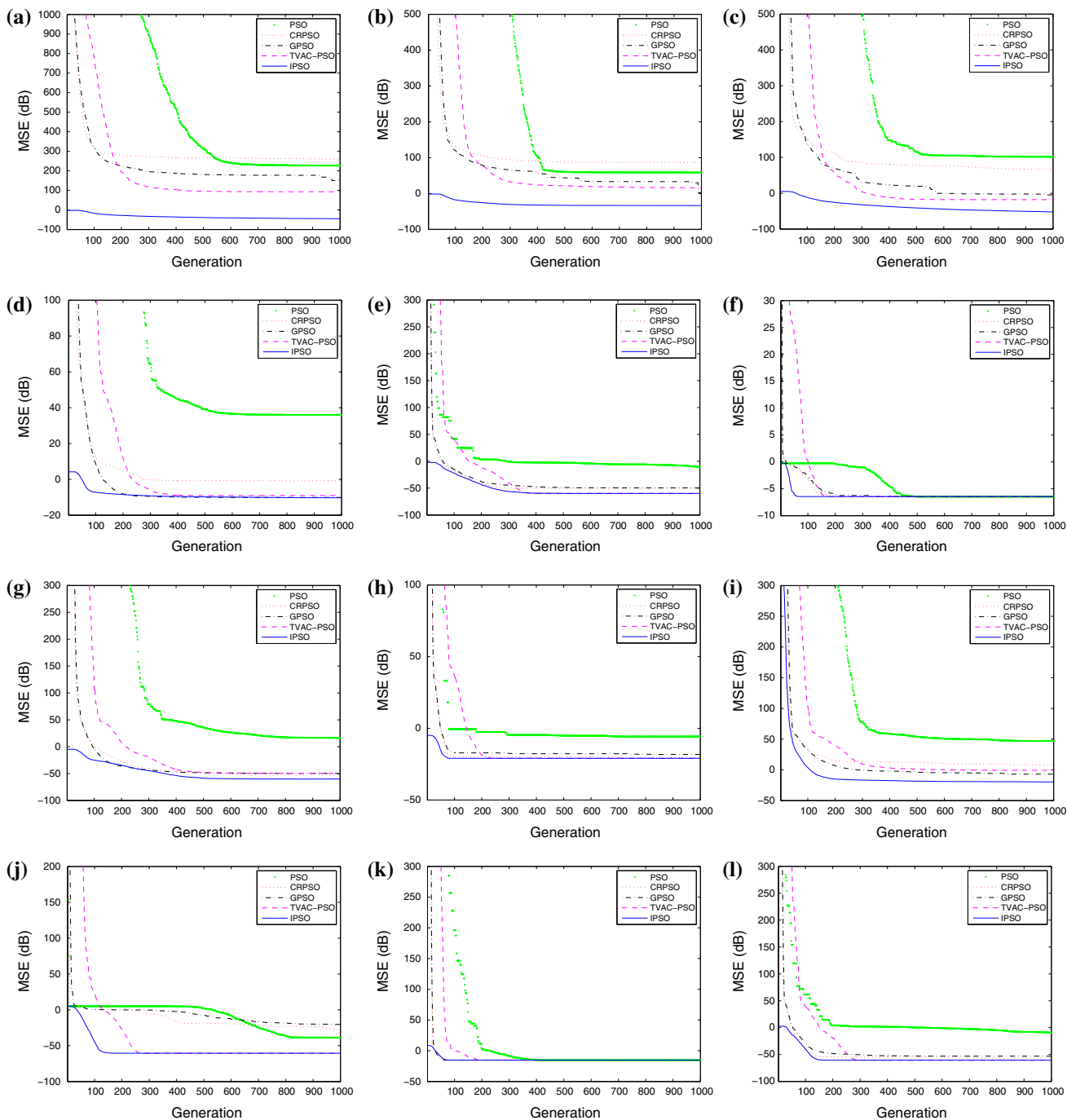


Fig. 3 Average optimization curves of five PSO approaches on solving 12 IIR system identification problems. **a** Example I (Case 1), **b** Example I (Case 2), **c** Example II (Case 1), **d** Example II (Case 2), **e** Example III (Case 1), **f** Example III (Case 2), **g** Example IV (Case 1), **h** Example IV (Case 2), **i** Example V, **j** Example VI, **k** Example VII, **l** Example VIII

Furthermore, the goal of step (4) is to gain confidence in, or reject the postulated mathematical model based on the final parameters optimized by the IPSO algorithm. Concretely, if there is an adequate correspondence between the outputs of the postulated model and the actual system outputs, the postulated model is verified and vice versa.

In addition to the ABC, CSO, and PSO algorithms, our future work will concentrate on the application of krill herd algorithm [60–64] and other metaheuristic algorithms [65–70] to IIR system identification, which has been proved to be effective and efficient on solving some other complex optimization problems.

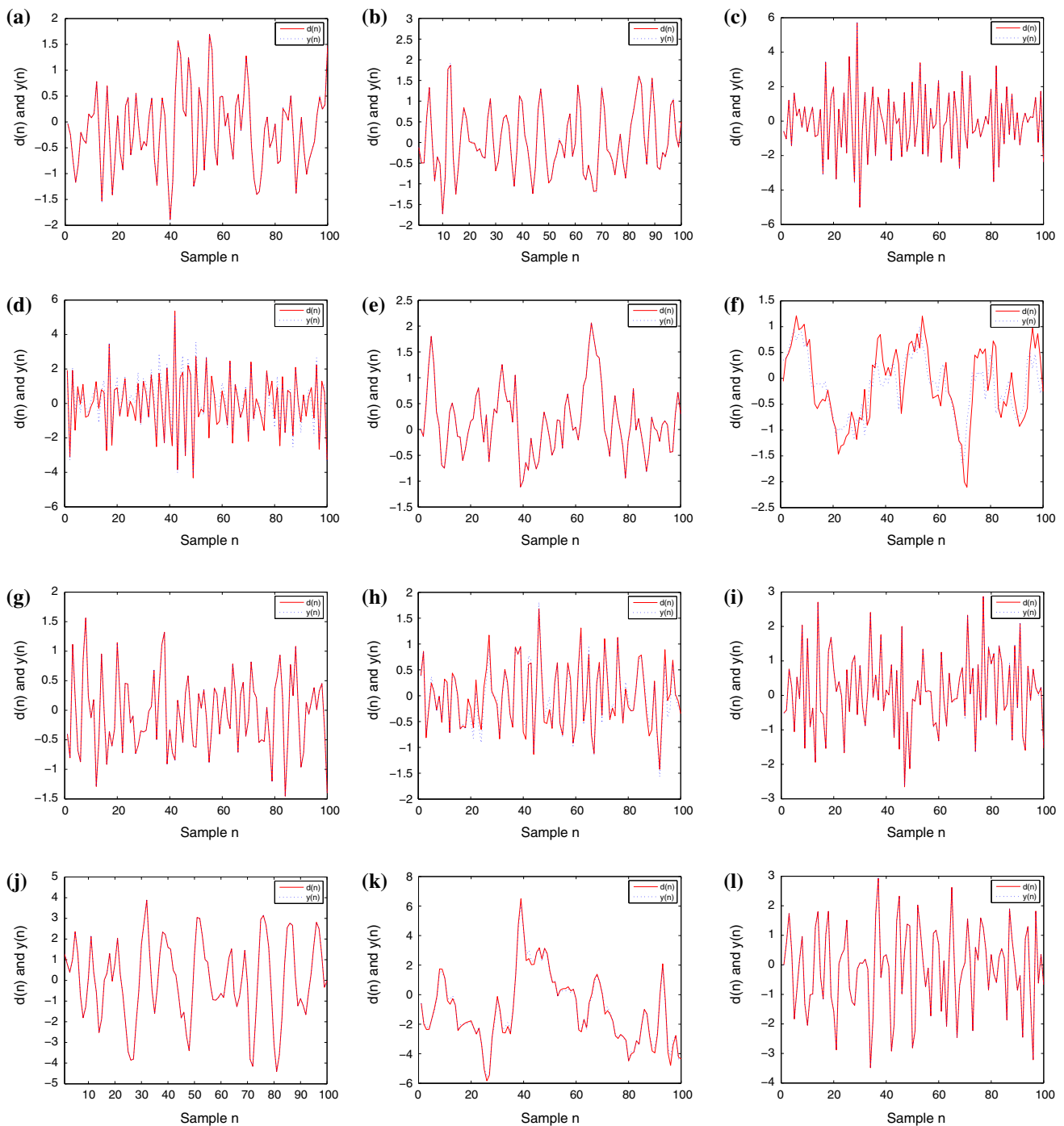


Fig. 4 Comparison of actual plant output and IIR model output using IPSO for 12 IIR system identification problems. **a** Example I (Case 1), **b** Example I (Case 2), **c** Example II (Case 1), **d** Example II (Case 2),

e Example III (Case 1), **f** Example III (Case 2), **g** Example IV (Case 1), **h** Example IV (Case 2), **i** Example V, **j** Example VI, **k** Example VII, **l** Example VIII

6 Conclusions

Identification of the actual unknown IIR plant based on IIR model design is often a complex process. This research presents an effective alternative approach namely IPSO for optimizing the mean square error (MSE) associated with

IIR system identification problem. Specifically, IPSO is utilized to determine suitable coefficients of IIR model such that the objective function MSE is minimized. IPSO improves PSO in three aspects as follows: To begin with, it segments solution space using golden ratio at the beginning of evolutionary process; next, it dynamically adjusts inertia

weights according to a random assigning strategy; finally, it introduces a global disturbance step by using normal distribution. The above three improvements are easily implemented by MATLAB 7.0 in the environment of Intel(R) Core(TM) i5-2410M CPU @ 2.30 GHz. Five PSO approaches including PSO, CRPSO, GPSO, TVAC-PSO, and IPSO are used to solve the identification problems of 12 unknown IIR systems. From the simulations, it is observed that IPSO is able to converge very rapidly and achieves the lowest levels for all cases. In light of this, we can infer that IPSO is superior to the other four PSO approaches for IIR system identification problem.

Most cases considered in this paper are small scale, and thus, our future work will focus on the IIR system identification problems with larger scale. In addition, we will try to use some other potential heuristic algorithms to solve these large-scale problems.

Acknowledgments This work was supported by the National Natural Science Foundation of China (Nos. 61403174, 61503165), Jiangsu Province Science Foundation for Youths (No. BK20150239).

References

- Krusiński DJ, Jenkins WK (2003) Adaptive filtering via particle swarm optimization. In: Proceedings of the 37th Asilomar conference on signals, systems and computers, vol 1, pp 571–575. doi:[10.1109/ACSSC.2003.1291975](https://doi.org/10.1109/ACSSC.2003.1291975)
- Krusiński DJ, Jenkins WK (2004) Particle swarm optimization for adaptive IIR filter structure. IEEE Congr Evolut Comput CEC 1:965–970. doi:[10.1109/CEC.2004.1330966](https://doi.org/10.1109/CEC.2004.1330966)
- Karaboga D, Basturk B (2007) A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. J Global Optim 39(3):459–471. doi:[10.1007/s10898-007-9149-x](https://doi.org/10.1007/s10898-007-9149-x)
- Karaboga N (2009) A new design method based on artificial bee colony algorithm for digital IIR filters. J Frankl Inst 346(4):328–348. doi:[10.1016/j.jfranklin.2008.11.003](https://doi.org/10.1016/j.jfranklin.2008.11.003)
- Luitel B, Venayagamoorthy GK (2010) Particle swarm optimization with quantum infusion for system identification. Eng Appl Artif Intell 23(5):635–649. doi:[10.1016/j.engappai.2010.01.022](https://doi.org/10.1016/j.engappai.2010.01.022)
- Luitel B, Venayagamoorthy GK (2009) A PSO with quantum infusion algorithm for training simultaneous recurrent neural networks. In: IEEE-INNS international joint conference on neural networks (IJCNN), pp 1923–1930. doi:[10.1109/IJCNN.2009.5179082](https://doi.org/10.1109/IJCNN.2009.5179082)
- Dai CH, Chen WR, Zhu YF (2010) Seeker optimization algorithm for digital IIR filter design. IEEE Trans Ind Electron 57(5):1710–1718. doi:[10.1109/TIE.2009.2031194](https://doi.org/10.1109/TIE.2009.2031194)
- Panda G, Pradhan PM, Majhi B (2011) IIR system identification using cat swarm optimization. Expert Syst Appl 38(10):12671–12683. doi:[10.1016/j.eswa.2011.04.054](https://doi.org/10.1016/j.eswa.2011.04.054)
- Chu SC, Tsai PW (2007) Computational intelligence based on the behavior of cats. Int J Innov Comput Inf Control 3(1):163–173
- Upadhyay P, Kar R, Mandal D, Ghoshal SP (2014) Crazyness based particle swarm optimization algorithm for IIR system identification problem. AEU-Int J Electron Commun 68(5):369–378. doi:[10.1016/j.aeue.2013.10.003](https://doi.org/10.1016/j.aeue.2013.10.003)
- Kennedy J, Eberhart RC (1995) Particle swarm optimization. In: Proceedings of IEEE international conference on neural networks, pp 1942–1948. doi:[10.1109/ICNN.1995.488968](https://doi.org/10.1109/ICNN.1995.488968)
- Shi YH, Eberhart RC (1999) Empirical study of particle swarm optimization. In: Proceedings of the IEEE congress on evolutionary computation, pp 1945–1950. doi:[10.1109/CEC.1999.785511](https://doi.org/10.1109/CEC.1999.785511)
- White SA (1975) An adaptive recursive digital filter. In: Proceedings of the 9th asilomar conference: circuits, systems, computers, pp 21–25
- Shynk JJ (1989) Adaptive IIR filtering. IEEE Trans Acoust Speech Signal Process 6(2):4–21. doi:[10.1109/53.29644](https://doi.org/10.1109/53.29644)
- Ng S, Leung S, Chung C, Luk A, Lau W (1996) The genetic search approach: a new learning algorithm for adaptive iir filtering. IEEE Trans Signal Process 13(6):38–46. doi:[10.1109/79.543974](https://doi.org/10.1109/79.543974)
- Abe M, Kawamata M (1998) Evolutionary digital filtering for IIR adaptive digital filters based on the cloning and mating reproduction. IEICE Trans Fundam Electron Commun Computer Sci E81-A(3):398–406
- Kalinli A, Karaboga N (2005) Artificial immune algorithm for IIR filter design. Eng Appl Artif Intell 18(8):919–929. doi:[10.1016/j.engappai.2005.03.009](https://doi.org/10.1016/j.engappai.2005.03.009)
- Proakis JG, Manolakis DG (2007) Digital signal processing: principles, algorithms and applications, 4th edn. Pearson Education, New Jersey
- Zhao JJ, Ji GH, Xia Y, Zhang XL (2015) Cavitory nodule segmentation in computed tomography images based on self-generating neural networks and particle swarm optimisation. Int J Bio-Inspired Comput 7(1):62–67. doi:[10.1504/IJBIC.2015.067999](https://doi.org/10.1504/IJBIC.2015.067999)
- Wang Z, Qin L, Yang W (2015) A self-organising cooperative hunting by robotic swarm based on particle swarm optimisation localisation. Int J Bio-Inspired Comput 7(1):68–73. doi:[10.1504/IJBIC.2015.068001](https://doi.org/10.1504/IJBIC.2015.068001)
- Grillo H, Peidro D, Alemany M, Mula J (2015) Application of particle swarm optimisation with backward calculation to solve a fuzzy multi-objective supply chain master planning model. Int J Bio-Inspired Comput 7(3):157–169. doi:[10.1504/IJBIC.2015.069557](https://doi.org/10.1504/IJBIC.2015.069557)
- Wang GG, Gandomi AH, Yang XS, Alavi AH (2014) A novel improved accelerated particle swarm optimization algorithm for global numerical optimization. Eng Comput 31(7):1198–1220. doi:[10.1108/EC-10-2012-0232](https://doi.org/10.1108/EC-10-2012-0232)
- Lu Q, Han QL, Liu SR (2014) A finite-time particle swarm optimization algorithm for odor source localization. Inf Sci 277:111–140. doi:[10.1016/j.ins.2014.02.010](https://doi.org/10.1016/j.ins.2014.02.010)
- Wang SC, Yeh MF (2014) A modified particle swarm optimization for aggregate production planning. Expert Syst Appl 41(6):3069–3077. doi:[10.1016/j.eswa.2013.10.038](https://doi.org/10.1016/j.eswa.2013.10.038)
- Boubaker S, Djemai M, Manamanni N, M'Sahli F (2014) Active modes and switching instants identification for linear switched systems based on discrete particle swarm optimization. Appl Soft Comput 14:482–488. doi:[10.1016/j.asoc.2013.09.009](https://doi.org/10.1016/j.asoc.2013.09.009)
- Jamali S, Shaker V (2014) Defense against SYN flooding attacks: a particle swarm optimization approach. Comput Electr Eng 40(6):2013–2025. doi:[10.1016/j.compeleceng.2014.05.012](https://doi.org/10.1016/j.compeleceng.2014.05.012)
- Sianoa P, Citro C (2014) Designing fuzzy logic controllers for DC-DC converters using multi-objective particle swarm optimization. Electr Power Syst Res 112:74–83. doi:[10.1016/j.epsr.2014.03.010](https://doi.org/10.1016/j.epsr.2014.03.010)
- Mandal S, Ghoshal SP, Kar R, Mandal D (2012) Design of optimal linear phase FIR highpass filter using crazyness based particle swarm optimization technique. J King Saud Univ-Comp Inf Sci 24:83–92. doi:[10.1016/j.jksuci.2011.10.007](https://doi.org/10.1016/j.jksuci.2011.10.007)

29. Mandal S, Ghoshal SP, Kar R, Mandal D (2011) Optimal linear phase FIR band passfilter design using craziness based particle swarm optimization algorithm. *J Shanghai Jiaotong Univ (Science)* 16(6):696–703. doi:[10.1007/s12204-011-1213-5](https://doi.org/10.1007/s12204-011-1213-5)
30. Mandal D, Ghoshal SP, Bhattacharjee AK (2010) Radiation pattern optimization for concentric circular antenna array with central element feeding using craziness based particle swarm optimization. *Int J RF Microw Comput Aided Eng* 20(5):577–586. doi:[10.1002/mmce.20467](https://doi.org/10.1002/mmce.20467)
31. Gao LQ, Li RP, Zou DX (2011) A global particle swarm optimization algorithm. *J Northeastern Univ (Natural Science)* 32(11):1538–1541
32. Mohammadi-Ivatloo B, Moradi-Dalvand M, Rabiee A (2013) Combined heat and power economic dispatch problem solution using particle swarm optimization with time varying acceleration coefficients. *Electr Power Syst Res* 95:9–18. doi:[10.1016/j.epsr.2012.08.005](https://doi.org/10.1016/j.epsr.2012.08.005)
33. Chaturvedi KT, Pandit M, Srivastava L (2009) Particle swarm optimization with time varying acceleration coefficients for non-convex economic power dispatch. *Electr Power Energy Syst* 31(6):249–257. doi:[10.1016/j.ijepes.2009.01.010](https://doi.org/10.1016/j.ijepes.2009.01.010)
34. Amaya I, Correa R (2015) Finding resonant frequencies of microwave cavities through a modified harmony search algorithm. *Int J Bio-Inspired Comput* 7(5):285–295. doi:[10.1504/IJBIC.2015.072258](https://doi.org/10.1504/IJBIC.2015.072258)
35. Bilbao MN, Ser JD, Salcedo-Sanz S, Casanova-Mateo C (2015) On the application of multi-objective harmony search heuristics to the predictive deployment of firefighting aircrafts: a realistic case study. *Int J Bio-Inspired Comput* 7(5):270–284. doi:[10.1504/IJBIC.2015.072257](https://doi.org/10.1504/IJBIC.2015.072257)
36. Coletta LF, Hruschka ER, Acharya A, Ghosh J (2015) A differential evolution algorithm to optimise the combination of classifier and cluster ensembles. *Int J Bio-Inspired Comput* 7(2):111–124. doi:[10.1504/IJBIC.2015.069288](https://doi.org/10.1504/IJBIC.2015.069288)
37. Amirjanov A, Sobolev K (2015) Changing range genetic algorithm for multimodal function optimisation. *Int J Bio-Inspired Comput* 7(4):209–221. doi:[10.1504/IJBIC.2015.071075](https://doi.org/10.1504/IJBIC.2015.071075)
38. Wang GG, Deb S, Gandomi AH, Zhang ZJ, Alavi AH (2015) Chaotic cuckoo search. *Soft Comput*. doi:[10.1007/s00500-015-1726-1](https://doi.org/10.1007/s00500-015-1726-1)
39. Yang XS, Deb S, Karamangolu M, He XS (2012) Cuckoo search for business optimization applications. In: *Proceedings of NCCCS2012, IEEE*, pp 1–5. doi:[10.1109/NCCCS.2012.6412973](https://doi.org/10.1109/NCCCS.2012.6412973)
40. Wang G-G, Gandomi AH, Zhao X, Chu HE (2016) Hybridizing harmony search algorithm with cuckoo search for global numerical optimization. *Soft Comput* 20(1):273–285. doi:[10.1007/s00500-014-1502-7](https://doi.org/10.1007/s00500-014-1502-7)
41. Wang GG, Deb S, Coelho LdS (2015) Earthworm optimization algorithm: a bio-inspired metaheuristic algorithm for global optimization problems. *Int J Bio-Inspired Comput* (in press)
42. Wang GG, Deb S, Gao X-Z, Coelho LdS (2016) A new meta-heuristic optimization algorithm motivated by elephant herding behavior. *Int J Bio-Inspired Comput* (in press)
43. Wang GG, Deb S, Coelho LdS (2015) Elephant herding optimization. Paper presented at the 2015 3rd international symposium on computational and business intelligence (ISCBI 2015), Bali, Indonesia, December 7–9
44. Feng YH, Wang GG, Deb S, Lu M, Zhao XJ (2015) Solving 0-1 knapsack problem by a novel binary monarch butterfly optimization. *Neural Comput Appl*. doi:[10.1007/s00521-015-2135-1](https://doi.org/10.1007/s00521-015-2135-1)
45. Wang GG, Zhao X, Deb S (2015) A novel monarch butterfly optimization with greedy strategy and self-adaptive crossover operator. Paper presented at the 2015 2nd intelligence conference on soft computing & machine intelligence (ISCMi 2015), Hong Kong, 23–24 Nov 2015
46. Yang XS, Deb S (2014) Fong S (2014) Bat algorithm is better than intermittent search strategy. *J Multi-Valued Logic Soft Comput* 22(3):223–237
47. Xue F, Cai Y, Cao Y, Cui Z, Li F (2015) Optimal parameter settings for bat algorithm. *Int J Bio-Inspired Comput* 7(2):125–128. doi:[10.1504/ijbic.2015.069304](https://doi.org/10.1504/ijbic.2015.069304)
48. Wang GG, Chu HCE, Mirjalili S (2016) Three-dimensional path planning forUCAV using an improved bat algorithm. *Aerosp Sci Technol* 49:231–238. doi:[10.1016/j.ast.2015.11.040](https://doi.org/10.1016/j.ast.2015.11.040)
49. Wang GG, Gandomi AH, Alavi AH (2014) An effective krill herd algorithm with migration operator in biogeography-based optimization. *Appl Math Model* 38(9–10):2454–2462. doi:[10.1016/j.apm.2013.10.052](https://doi.org/10.1016/j.apm.2013.10.052)
50. Wang GG, Deb S, Gandomi AH, Alavi AH (2016) Opposition-based krill herd algorithm with Cauchy mutation and position clamping. *Neurocomputing* 177:147–157. doi:[10.1016/j.neucom.2015.11.018](https://doi.org/10.1016/j.neucom.2015.11.018)
51. Wang GG, Gandomi AH, Alavi AH, Hao G-S (2014) Hybrid krill herd algorithm with differential evolution for global numerical optimization. *Neural Comput Appl* 25(2):297–308. doi:[10.1007/s00521-013-1485-9](https://doi.org/10.1007/s00521-013-1485-9)
52. Wang GG, Gandomi AH, Alavi AH (2013) A chaotic particle-swarm krill herd algorithm for global numerical optimization. *Kybernetes* 42(6):962–978. doi:[10.1108/K-11-2012-0108](https://doi.org/10.1108/K-11-2012-0108)
53. Henein MY, Collaborators GR, Zhao Y, Nicoll R, Sun L, Khir AW, Franklin K, Lindqvist P (2011) The human heart: application of the golden ratio and angle. *Int J Cardiol* 150(3):239–242. doi:[10.1016/j.ijcard.2011.05.094](https://doi.org/10.1016/j.ijcard.2011.05.094)
54. Xie ZF (2011) The golden ratio and super central configurations of the n-body problem. *J Differ Equ* 251(1):58–72. doi:[10.1016/j.jde.2011.03.002](https://doi.org/10.1016/j.jde.2011.03.002)
55. Schmid K, Marx D, Samal A (2008) Computation of a face attractiveness index based on neoclassical canons, symmetry, and golden ratios. *Pattern Recogn* 41(8):2710–2717. doi:[10.1016/j.patcog.2007.11.022](https://doi.org/10.1016/j.patcog.2007.11.022)
56. Pallett PM, Link S, Lee K (2010) New “golden” ratios for facial beauty. *Vis Res* 50(2):149–154. doi:[10.1016/j.visres.2009.11.003](https://doi.org/10.1016/j.visres.2009.11.003)
57. Majhi B, Panda G, Choubey A (2008) Efficient scheme of pole-zero system identification using particle swarm optimization technique. In: *IEEE congress on evolutionary computation*, pp 446–451. doi:[10.1109/CEC.2008.4630836](https://doi.org/10.1109/CEC.2008.4630836)
58. Durmus B, Gun A (2011) Parameter identification using particle swarm optimization. In: *6th International advanced technologies symposium*, pp 188–92
59. Yu X, Liu J, Li H (2009) An adaptive inertia weight particle swarm optimization algorithm for IIR digital filter. *IEEE Int Conf Artif Comput Intell* 1:114–118. doi:[10.1109/AICI.2009.28](https://doi.org/10.1109/AICI.2009.28)
60. Wang GG, Guo LH, Wang HQ, Duan H, Liu L, Li J (2014) Incorporating mutation scheme into krill herd algorithm for global numerical optimization. *Neural Comput Appl* 24(3):853–871. doi:[10.1007/s00521-012-1304-8](https://doi.org/10.1007/s00521-012-1304-8)
61. Wang GG, Guo LH, Gandomi AH, Hao GS, Wang HQ (2014) Chaotic krill herd algorithm. *Inf Sci* 274:17–34. doi:[10.1016/j.ins.2014.02.123](https://doi.org/10.1016/j.ins.2014.02.123)
62. Wang GG, Gandomi AH, Alavi AH (2014) Stud krill herd algorithm. *Neurocomputing* 128:363–370. doi:[10.1016/j.neucom.2013.08.031](https://doi.org/10.1016/j.neucom.2013.08.031)
63. Guo LH, Wang GG, Gandomi AH, Alavi AH, Duan H (2014) A new improved krill herd algorithm for global numerical optimization. *Neurocomputing* 138:392–402. doi:[10.1016/j.neucom.2014.01.023](https://doi.org/10.1016/j.neucom.2014.01.023)
64. Wang GG, Gandomi AH, Yang XS, Alavi AH (2014) A new hybrid method based on krill herd and cuckoo search for global optimization tasks. *Int J Bio-Inspired Comput* (in press)

65. Yang XS, Deb S, Fong S (2014) Metaheuristic algorithms: optimal balance of intensification & diversification. *Appl Math Inf Sci* 8(3):977–983. doi:[10.12785/amis/080306](https://doi.org/10.12785/amis/080306)
66. Yang XS, Deb S, Hanne T, He X (2015) Attraction and diffusion in nature-inspired optimization algorithms. *Neural Comput Appl*. doi:[10.1007/s00521-015-1925-9](https://doi.org/10.1007/s00521-015-1925-9)
67. Cuevas E, González A, Zaldívar D, Pérez-Cisneros M (2015) An optimisation algorithm based on the behaviour of locust swarms. *Int J Bio-Inspired Comput* 7(6):402–407. doi:[10.1504/IJBIC.2015.073178](https://doi.org/10.1504/IJBIC.2015.073178)
68. Guo L, Wang G-G, Wang H, Wang D (2013) An effective hybrid firefly algorithm with harmony search for global numerical optimization. *Sci World J* 2013:9, Article ID 125625. doi:[10.1155/2013/125625](https://doi.org/10.1155/2013/125625)
69. Duan H, Zhao W, Wang G-G, Feng X (2012) Test-sheet composition using analytic hierarchy process and hybrid metaheuristic algorithm TS/BBO. *Math Probl Eng* 2012:22, Article ID 712752. doi:[10.1155/2012/712752](https://doi.org/10.1155/2012/712752)
70. Wang G-G, Guo L, Duan H, Liu L, Wang H, Wang J (2012) A hybrid meta-heuristic DE/CS algorithm for UCAV path planning. *J Inf Comput Sci* 9(16):4811–4818