Complex & Intelligent Systems (2022) 8:1217–1228
https://doi.org/10.1007/s40747-021-00589-2

ORIGINAL ARTICLE

A novel PID-like particle swarm optimizer: on terminal convergence analysis

Chuang Wang[1,2] · Zidong Wang[3] · Fei Han[1,2] · Hongli Dong[1,2] · Hongjian Liu[1,4,5]

Received: 31 August 2021 / Accepted: 26 October 2021 / Published online: 25 November 2021

Abstract

In this paper, a novel proportion-integral-derivative-like particle swarm optimization (PIDLPSO) algorithm is presented with improved terminal convergence of the particle dynamics. A derivative control term is introduced into the traditional particle swarm optimization (PSO) algorithm so as to alleviate the overshoot problem during the stage of the terminal convergence. The velocity of the particle is updated according to the past momentum, the present positions (including the personal best position and the global best position), and the future trend of the positions, thereby accelerating the terminal convergence and adjusting the search direction to jump out of the area around the local optima. By using a combination of the Routh stability criterion and the final value theorem of the Z-transformation, the convergence conditions are obtained for the developed PIDLPSO algorithm. Finally, the experiment results reveal the superiority of the designed PIDLPSO algorithm over several other state-of-the-art PSO variants in terms of the population diversity, searching ability and convergence rate.

Keywords Particle swarm optimization · Terminal convergence analysis · Proportional-integral-derivative strategy · Z-transformtion · Routh stability criterion

Introduction

With the breakthroughs in both theory and applications of the evolutionary computing (EC), the evolutionary optimization algorithms have attracted a great deal of research interest, and a large amount of research results have been published in the literature [1,31,39,41]. Generally, the EC algorithms can be roughly divided into the categories of genetic algorithms, genetic programming, evolution strategies, and evolution programming. Through simulating the interactions among the individuals in the fish schooling or bird flocking, the particle swarm optimization (PSO) algorithm has been presented in [16] with the purpose of exploring the searching space, which is made possible by automatically adjusting the current velocities and the current positions of the particles according to the competition and cooperation among the particles. Serving as a powerful evolutionary technique, the PSO algorithm is capable of discovering the globally optimal solution in an efficient yet effective way in the research areas of parameter optimization, neural network training, clustering analysis, combination optimization, pattern recognition, image processing and so forth [1,5,6,8,17,20,22,30,32,36,41,46,50].

Unfortunately, like other population-based EC approaches, the PSO algorithm still has the issue of easily getting trapped in the local optima when dealing with large-scale complex

This work was supported in part by the National Natural Science Foundation of China under Grants 61873148, 61933007 and 620730070, the AHPU Youth Top-notch Talent Support Program of China under Grant 2018BJRC009, the Natural Science Foundation of Anhui Province of China under Grant 2108085MA07, the Royal Society of the UK, and the Alexander von Humboldt Foundation of Germany.

✉ Hongjian Liu
hjliu1980@gmail.com

Zidong Wang
Zidong.Wang@brunel.ac.uk

Hongli Dong
shiningdhl@vip.126.com

[1] Artificial Intelligence Energy Research Institute, Northeast Petroleum University, Daqing 163318, China

[2] Heilongjiang Provincial Key Laboratory of Networking and Intelligent Control, Northeast Petroleum University, Daqing 163318, China

[3] Department of Computer Science, Brunel University London, Uxbridge, Middlesex UB8 3PH, UK

[4] Key Laboratory of Advanced Perception and Intelligent Control of High-end Equipment, Ministry of Education, Anhui Polytechnic University, Wuhu 241000, China

[5] School of Mathematics and Physics, Anhui Polytechnic University, Wuhu 241000, China

optimization problems. As such, it is of vital importance to develop advanced strategies/variants to improve the optimization capability of conventional PSO algorithm [44]. Up to now, many researchers have devoted tremendous efforts in improving the searching ability of the existing PSO algorithms and developing advanced PSO variants with aim to alleviate premature convergence, see, e.g., [28,41,42]. To be more specific, a PSO algorithm with saturation and time-delay has been developed in [42] to ensure the convergence and increase the possibility of escaping from local optimum. Recently, an N-state Markovian jumping PSO variant has been presented in [28] to adjust evolutionary state according to the N-state Markov chain [4,7,18,21,56,57], showing better exploration ability than that of other algorithms. It is noted that the competitiveness of these advanced variants (in terms of both the convergence rate and the searching ability) have mostly been demonstrated via numerical simulations, and there is a lack of rigorous proof of the performance from the theoretical viewpoint. The objective of this paper is, therefore, to further improve the performance of the PSO algorithm from both aspects of theory and simulation.

Over the past few decades, much research effort has been made to the analysis of the convergence of the PSO algorithms, and a variety of efficient approaches have been presented in the literature [13,19,24,29,33,39]. To be more specific, in [33], the first and most important empirical study has been reported by Eberhart and Shi regarding the PSO algorithm. The convergence analysis of the PSO algorithm has been studied in [24] from the theoretical aspect, and further insights have been provided in [13,19,29,39]. Based on these existing results, we can draw the following conclusions on the performance of the PSO algorithm: 1) the exploration and exploitation ability of the PSO algorithm to control the population diversity are vitally important for its efficiency as an optimizer; and 2) as with the other population-based optimizers, higher population diversity is desirable in the early exploration stage, while lower population diversity is preferable in the later/terminal convergence stage. These conclusions, without any doubts, provide some insights into the mechanism of how the PSO algorithm behaves well. Nevertheless, there is still room to further improve the terminal convergence of the PSO algorithm. It is noticed that, in [49], the traditional particle swarm optimizer has been interpreted as a proportional-integral controller. Following this line, in this paper, we endeavor to develop the PSO algorithm based on the proportional-integral-derivative (PID) strategy and also analyze the terminal convergence of this PID-like PSO (PIDLPSO) algorithm.

As is well known, the proportional-integral-derivative (PID) control strategy has been widely applied in industry (e.g. aerospace and industrial robotics) owing to its advantages of simple structure, few tuning parameters, outstanding control performance and so on, see [2,52] and the reference therein. Proportional control can be easily implemented where the output of the controller is proportional to the error signal of the input. By using proportional control alone, the controlled system would suffer from the steady-state error that cannot be eliminated. As such, the integral control is introduced to form the proportional-integral (PI) control strategy, which ensures that the output of the controlled system traces the input precisely. In addition, the derivative control strategy has the advantages of quick action and advanced adjustment, which is conducive to improving the performance of controlled object with large time-delays effectively, though it cannot easily remove the residual error. Therefore, the PID control strategy has become more and more popular in practice because it combines the merits of (1) timeliness and rapidity of the proportional control; (2) the residual elimination ability of the integral control; and (3) the advanced adjustment ability of the derivative control.

Inspired by the insight that the particle swarm optimizer could be approximately a PI controller [48], in this paper, the derivative control strategy is introduced into the PSO algorithm with aim to further enhance the optimization ability and improve the convergence rate. As compared with the traditional PSO algorithm, such a PID-like PSO (PIDLPSO) algorithm owns the following two advantages: (1) the overshoot problem during the stage of the terminal convergence of the particle dynamics can be adequately resolved through adjusting the change of deviation signal; and (2) more historical information can now be utilized that is beneficial for explore the problem space more thoroughly.

In connection with the discussions made so far, the main objective of this paper is to put forward a novel PIDLPSO algorithm with rigorous mathematical proof of the terminal convergence. The main contributions of this paper are summarized in threefold as follows.

(1) A novel PIDLPSO algorithm is proposed to alleviate the overshoot problem and accelerate convergence during the later/terminal stage of the particle dynamics, where the velocity of the particle is updated according to the past momentum, the present positions (including the personal best position and the global best position), as well as the future trend of the positions.
(2) For the proposed PIDLPSO algorithm, the convergence conditions and the final positions are obtained by means of the Routh stability criterion and the final value theorem of the Z-transformation.
(3) The proposed PIDLPSO algorithm is comprehensively verified from the aspects of population diversity, searching ability and convergence rate. Also, it is demonstrated that the PIDLPSO algorithm has more competitive ability in achieving the global optimum than five other popular PSO algorithms.

The structure of this paper is outlined as follows. Section 2 formulates the problem to be studied for the PSO algorithm. Section 3 puts forward a novel PIDLPSO algorithm and analyze its convergence conditions. Experimental results are presented in Sect. 4 with detailed discussions, and Sect. 5 outlines the conclusions and future directions.

Notation. The Z-transform of a vector implies that every element of this vector has taken the Z-transform.

Problem formulation

In the typical PSO algorithm developed [16], the particles, referred to as the feasible candidates, are employed to explore and exploit in the D-dimensional searching space by continuously adjusting the velocity vector $v_i(k) = (v_{i1}(k), v_{i2}(k), ..., v_{iD}(k))$ and the position vector $x_i(k) = (x_{i1}(k), x_{i2}(k), ...x_{iD}(k))$, respectively. According to the competition and cooperation among the particles, the position of the i-th particle is justified towards two directions, where one direction is the personal best position p_{best} represented by $p_{\text{best}} = (p_1, p_2, \ldots, p_D)$, and the other one is the globally optimal position g_{best} represented by $g_{\text{best}} = (g_1, g_2, \ldots, g_D)$. Specifically, the velocity and the position of the i-th particle at the $(k + 1)$-th iteration are updated as follows:

$$\begin{cases} v_i(k+1) = \omega v_i(k) + c_1 r_1 (p_{\text{best}} - x_i(k)) \\ \qquad\qquad + c_2 r_2 (g_{\text{best}} - x_i(k)), \\ x_i(k+1) = x_i(k) + v_i(k+1), \end{cases} \tag{1}$$

where k is the number of recent iterations of the i-th particle in the D-dimensional problem space, ω is the inertia weight, c_1 and c_2 called the cognitive and social parameters are the acceleration coefficients, and r_1, r_2 are constants selected on the interval [0, 1].

The main objective of this paper is to 1) put forward a novel PIDLPSO algorithm; 2) analyze its terminal convergence by means of the Routh stability criterion and the final value theorem of the Z-transformation; and 3) obtain the conditions for convergence and the position of the final particle of the proposed PIDLPSO algorithm.

The PIDLPSO algorithm and its terminal convergence analysis

Motivated by interactions among the individuals in the fish schooling or bird flocking, the PSO algorithm has been proposed in [16] with the purpose of exploring the searching space by updating a linear summation of the particle's past momentum and current search direction. To the best of the authors' knowledge, in the later/terminal stage of the evolution of the particle dynamics. Very little attention has been paid to the overshoot problem of the particle dynamics caused by the past momentum, and such an overshoot phenomenon could lead to oscillations which, in turn, slow down the convergence significantly especially for high-dimensional complex optimization problems [48].

According to the similarity between the PSO algorithm and the PI strategy [49], in this paper, we would like to propose a novel PIDLPSO algorithm, which is a yet another PSO variant, to better keep the tradeoff between the exploration and the exploitation with hope to alleviate the overshoot problem during the terminal stage of the convergence of the particle dynamics, where the PIDLPSO updates the velocity and position based on three factors, namely, the past momentum, the present positions (including the personal best position and the global best position), and the future trend of the position. Meanwhile, by combining the Routh stability criterion and the final value theorem of the Z-transformation, we shall obtain the convergence conditions of the PIDLPSO algorithm to be developed. The framework and convergence proof of the proposed PIDLPSO algorithm will be illustrated in details.

As discussed previously, the traditional PSO algorithm could be interpreted as a PI strategy and, in this context, a novel PSO variant is developed by introducing the following *derivative* term

$$\xi_i(k) = k_D(e_i(k) - e_i(k-1)), \tag{2}$$

where k_D is control coefficient and

$$e_i(k) = c_1 r_1 p_{\text{best}} + c_2 r_2 g_{\text{best}} - c_1 r_1 x_i(k) - c_2 r_2 x_i(k).$$

In the proposed PIDLPSO algorithm, the velocity and position of the i-th particle at the $(k + 1)$-th iteration are updated as follows:

$$\begin{cases} v_i(k+1) = \omega v_i(k) + c_1 r_1 (p_{\text{best}} - x_i(k)) \\ \qquad\qquad + c_2 r_2 (g_{\text{best}} - x_i(k)) + \xi_i(k), \\ x_i(k+1) = x_i(k) + v_i(k+1), \end{cases} \tag{3}$$

The following lemma will be used in obtaining our main results.

Lemma 1 [14] *(1) The PSO system does not have an equilibrium point if $p_{\text{best}} \neq g_{\text{best}}$. (2) If $p_{\text{best}} = g_{\text{best}} = x$ is time invariant, then there is a unique equilibrium point at $v_* = 0$ and $x_* = g_{\text{best}}$.*

Remark 1 Compared to the conventional PSO algorithm, a new term $\xi_i(k)$ has been added to the particle dynamics whose coefficient, the derivative control gain k_D, will be adequately designed to alleviate the overshoot problem by smoothening the terminal convergence of the particle dynamics. The convergence conditions and the position of the final particle are to be investigated in a mathematically rigorous way, and this constitutes the main contribution of this paper.

Theorem 1 *The novel PIDLPSO algorithm is convergent if the following inequality holds*

$$\begin{cases} \dfrac{\omega-1}{c_1r_1+c_2r_2} < k_D < \dfrac{\omega+1}{c_1r_1+c_2r_2} - \dfrac{1}{2}, \\ 0 < c_1r_1+c_2r_2 < 4. \end{cases} \tag{4}$$

Moreover, the final position of the i-th particle in the PIDLPSO algorithm is

$$x_i(\infty) = g_{\text{best}}. \tag{5}$$

Proof Considering (2) and (3), we have immediately that

$$x_i(k+1) = x_i(k) + \omega v_i(k) + e_i(k) + \xi_i(k). \tag{6}$$

Letting $k = k+1$, it follows that

$$\begin{aligned} x_i(k+2) =& x_i(k+1) + \omega v_i(k+1) \\ &+ (k_D+1)e_i(k+1) - k_D e_i(k). \end{aligned} \tag{7}$$

According to (2), we obtain

$$x_i(k+2) = \frac{c_1r_1p_{\text{best}} + c_2r_2g_{\text{best}} - e_i(k+2)}{c_1r_1+c_2r_2}, \tag{8}$$

$$x_i(k+1) = \frac{c_1r_1p_{\text{best}} + c_2r_2g_{\text{best}} - e_i(k+1)}{c_1r_1+c_2r_2}, \tag{9}$$

$$\omega v_i(k+1) = \omega\frac{e_i(k) - e_i(k+1)}{c_1r_1+c_2r_2}. \tag{10}$$

Substituting (8)–(10) into (7), one has

$$\begin{aligned} &e_i(k+2) + ((k_D+1)(c_1r_1+c_2r_2) - \omega - 1)e_i(k+1) \\ &+ (\omega - k_D(c_1r_1+c_2r_2))e_i(k) = 0. \end{aligned} \tag{11}$$

Taking the Z-transform of (11), we obtain

$$e_i(z) = \frac{Az^2 + Bz}{Cz^2 + Dz + E} \tag{12}$$

where

$$\begin{aligned} A &= e_i(0), \\ B &= 2e_i(1) + ((k_D+1)(c_1r_1+c_2r_2) - \omega - 1)e_i(0), \\ C &= 1, \\ D &= (k_D+1)(c_1r_1+c_2r_2) - \omega - 1, \\ E &= \omega - k_D(c_1r_1+c_2r_2). \end{aligned}$$

Taking the linear difference transformation

$$z = \frac{\mu+1}{\mu-1},$$

we arrive at

$$e_i(\mu) = \frac{A(\frac{\mu+1}{\mu-1})^2 + B\frac{\mu+1}{\mu-1}}{C(\frac{\mu+1}{\mu-1})^2 + D\frac{\mu+1}{\mu-1} + E} \tag{13}$$

The characteristic equation of (13) is calculated as follows:

$$\begin{aligned} &(c_1r_1+c_2r_2)\mu^2 + (2 - 2\omega + 2k_D(c_1r_1+c_2r_2))\mu \\ &+ 2 + 2\omega - (2k_D+1)(c_1r_1+c_2r_2) = 0. \end{aligned} \tag{14}$$

By utilizing the Routh stability criterion, we obtain the system stability conditions as follows:

$$\begin{cases} c_1r_1+c_2r_2 > 0, \\ 2 - 2\omega + 2k_D(c_1r_1+c_2r_2) > 0, \\ 2 + 2\omega - (2k_D+1)(c_1r_1+c_2r_2) > 0, \end{cases} \tag{15}$$

and therefore

$$\begin{cases} \dfrac{\omega-1}{c_1r_1+c_2r_2} < k_D < \dfrac{\omega+1}{c_1r_1+c_2r_2} - \dfrac{1}{2}, \\ 0 < c_1r_1+c_2r_2 < 4. \end{cases} \tag{16}$$

By means of the final value theorem of the Z-transform, we obtain that

$$e_i(\infty) = \lim_{z\to 1}(z-1)e_i(z) = 0. \tag{17}$$

Hence, the PIDLPSO algorithm will converge to

$$x_i(\infty) = \frac{c_1r_1p_{\text{best}} + c_2r_2g_{\text{best}}}{c_1r_1+c_2r_2}. \tag{18}$$

Finally, it follows from $p_{\text{best}} = g_{\text{best}}$ that

$$x_i(\infty) = g_{\text{best}} \tag{19}$$

which ends the proof. □

Remark 2 So far, a novel PIDLPSO algorithm has been proposed in which a new derivative control term is introduced so as to govern the smoothening process of the terminal convergence during the final stage of the dynamics evolution of the particles, thereby alleviating the overshoot/oscillation problems. In this PIDLPSO algorithm, the new derivative control term is mainly to fine-tune the future trend of the positions and hence provides yet another design freedom (in addition to the past momentum and the present positions) for improving the transient behaviors of the terminal convergence of the particle dynamics.

Remark 3 Comparing to the numerous versions of the PSO variants, our proposed PIDLPSO exhibits the following distinctive features: 1) a derivative control term is introduced to alleviate the overshoot problem and accelerate convergence during the later/terminal stage of the particle dynamics; 2) the convergence conditions and the final positions are obtained by means of the Routh stability criterion and the final value theorem of the Z-transformation; and 3) the PIDLPSO algorithm is more competitive (in achieving the global optimum) than several other popular PSO algorithms as demonstrated in the next section.

Simulation Experiments

In this section, the superiority of proposed PIDLPSO algorithm is demonstrated by comparing with five widely employed PSO variants in terms of population diversity, convergence rate and searching ability.

In the experiments, the population size is set as $m = 20$ and the dimension of the searching space is $D = 20$. For the PIDLPSO algorithm, the inertia weight $\omega = 0.729$, the acceleration coefficients $c_1 = c_2 = 1.5$, $r_1 = r_2 = 0.5$. The performance of the PIDLPSO algorithm with different settings of the derivative control K_D is shown in Table 1. It can be seen that the PIDLPSO algorithm demonstrates competitive performance when $k_D = 0.155$.

Population Diversity

In this paper, the variance (population spatial distribution) and the entropy (particle activity) are employed to describe the population diversity. The variance of the population in the k-th iteration is defined as follows

$$S_i(k) = \frac{1}{m}\sum_{i=1}^{m}\sum_{j=1}^{D}(x_i^j(k) - \bar{x}^j(k))^2, \tag{20}$$

where D denotes the dimension of the searching space, m is the number of the particles in the population, $x_i^j(k)$ indicates the i-th particle in the j-th dimension, and $\bar{x}^j(k)$ denotes the mean of the j-th dimension over all particles in the k-th iteration.

Figures 1 and 2 depict the variance of the population dynamics of the typical PSO algorithm and the proposed PIDLPSO algorithm when the iterations are set to 5000 and 20000, respectively. It is clearly seen from the figures that, with the number of iterations increasing, the population variance gradually decreases until it converges, which indicates that all the particles are dispersed to explore the whole space with the purpose of discovering globally optimal solution in the global exploration stage. Contrarily, during the local exploitation process, the others are inspired to move towards the globally optimal particle until convergence. Note that the population variance of the PIDLPSO is higher than that of the typical PSO algorithm, which illustrates the particle distribution of the PIDLPSO algorithm is more dispersed so as to search the whole space more thoroughly.

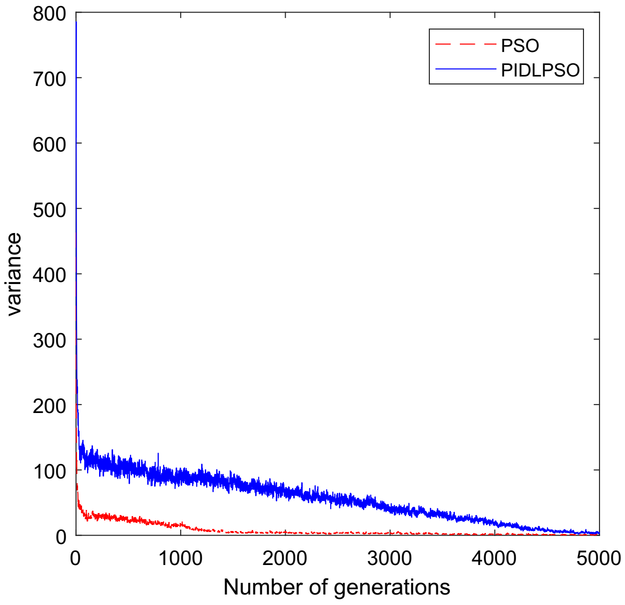

Fig. 1 Variation curves of population variance, when the maximum number of iterations is 5000

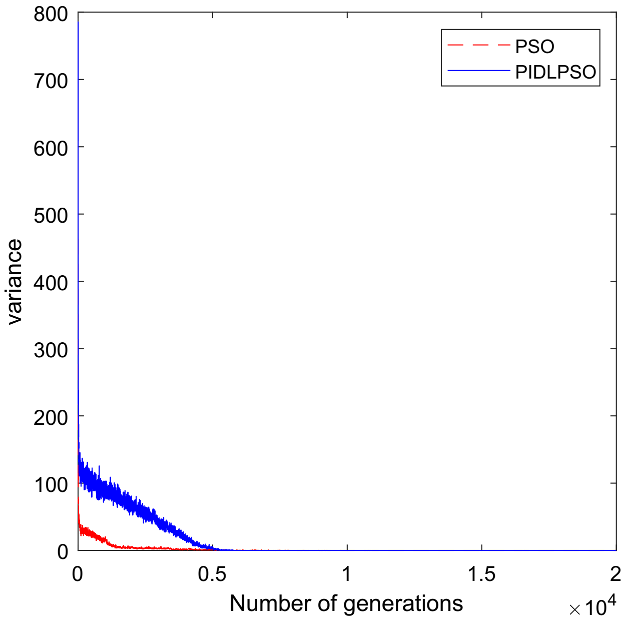

Fig. 2 Variation curves of population variance, when the maximum number of iterations is 20000

In the k-th iteration, the particles are divided into Q subsets denoted by $\{S_1(k), S_2(k), \ldots, S_Q(k)\}$. For $\forall p, q \in \{1, 2, \ldots, Q\}$, we have

$$S_p(k) \cap S_q(k) = \emptyset, \quad \bigcup_{q=1}^{Q} S_q(k) = A(k),$$

where $A(k)$ is the whole swarm set. The number of the particles in each subset is represented by

$$\{|S_1(k)|, |S_2(k)|, \ldots, |S_Q(k)|\},$$

and then the population entropy is defined as follows

$$Ep = -\sum_{j=1}^{Q} p_j \lg(p_j), \qquad (21)$$

where

$$p_j = \frac{|S_j(k)|}{m}$$

with m denoting the number of individuals in the whole swarm.

The dynamic curves of the population entropy with the number of iterations are depicted for the PSO algorithm and the PIDLPSO algorithm in Fig. 3. We can see that the population entropy is higher during the early stage of the iterations, the particles are distributed into the whole space to explore the globally optimal solution, and the curves of population entropy contain a large number of particle exploration information. While the population entropy is lower in the last stage of the iterations, which illustrates that other particles are encouraged to move towards the globally optimal particle. Therefore, the population entropy of the final curve is lower and more stable, which indicates that the PIDLPSO algorithm is convergent.

Optimization Performance

In this subsection, the searching ability of the PIDLPSO algorithm is tested and verified through a large number of simulation and comparison experiments. Note that the details of all the test functions are given in Table 2, which includes the dimension, the threshold, and etc.

In order to demonstrate the optimality of the PIDLPSO algorithm, in this paper, various improved PSO variants reported in recent literatures are employed for optimization capability evaluation via four widely-used test functions. In the consideration of evaluation factors, the comparisons are conducted from the following four aspects: (1) minimum; (2) mean; (3) standard deviation; and (4) success ratio.

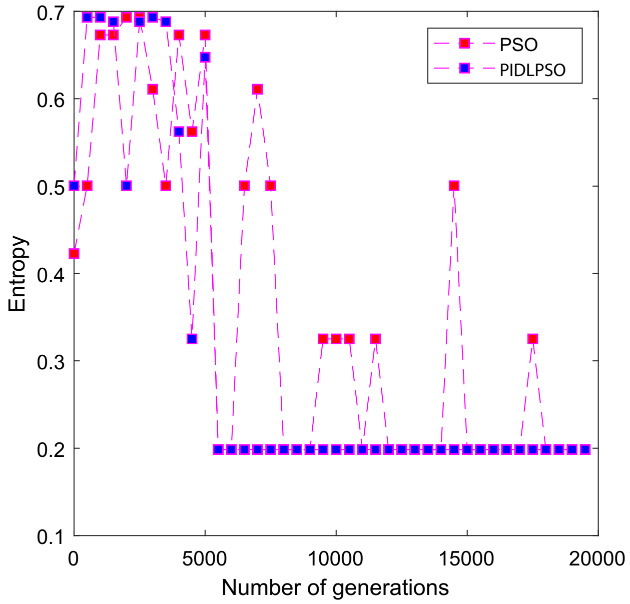

Fig. 3 Variation curves of population entropy

Table 3 lists the statistical results for various PSO variants. It is seen from Table 3 that, compared with other PSO variants, the PIDLPSO algorithm has the smallest or near the smallest minimum, mean and standard deviation, which illustrates that the proposed PIDLPSO algorithm is more competitive in searching performance. In addition, it should be noted that the success ratio is another significant indicator to assess the convergence characteristics, which illustrates the capability of jumping out of the local optimum. It is demonstrated from Table 3 that the success ratio of our PIDLPSO algorithm is the biggest reaching 100% for four benchmark functions, which is further verified that the PIDLPSO algorithm is better than other algorithms, especially in getting rid of local optimum.

Convergence analysis

Convergence rate

It is worth noting that the convergence rate is a crucial metric to assess the convergence of the PSO algorithms. The convergence rate of various PSO variants is illustrated in Figs. 4, 5, 6 and 7, where the abscissa and the ordinate respectively represents the iteration number and the mean fitness values of various PSO variants. It is clearly shown that the convergence rate of the proposed PIDLPSO algorithm is more competitive than that of other algorithms on the most test functions. In detail, it can be seen from Figs. 5, 6 and 7 that the convergence rate of the PIDLPSO algorithm is the fastest than that of other PSO variants. Although, in Fig. 4, the proposed PIDLPSO algorithm is not the best in convergence

Table 1 Statistics results of the PIDLPSO algorithm under different K_D conditions

		$f_1(x)$	$f_2(x)$	$f_3(x)$	$f_4(x)$
$k_D = -0.181$	Min	1.28×10^{-10}	1.16×10^{-3}	1.88	4.00
	Mean	6.58×10^{-4}	2.65	8.13	6.35×10^{1}
	Std.Dev	1.82×10^{-3}	2.29	4.20	1.09×10^{2}
	Ratio	100%	100%	0%	0%
$k_D = -0.155$	Min	8.64×10^{-12}	5.68×10^{-5}	1.55	4.00
	Mean	1.22×10^{-2}	2.44	6.25	5.21×10^{2}
	Std.Dev	5.03×10^{-2}	3.12	3.41	2.23×10^{3}
	Ratio	90%	100%	0%	0%
$k_D = -0.055$	Min	0	2.35×10^{-32}	5.39×10^{-38}	0
	Mean	4.94×10^{-324}	5.06×10^{-1}	6.03×10^{-29}	6.10
	Std.Dev	0.00	9.53×10^{-1}	2.63×10^{-28}	1.84×10^{1}
	Ratio	100%	100%	100%	40%
$k_D = 0.055$	Min	0	2.35×10^{-32}	2.22×10^{-51}	0
	Mean	0	1.24×10^{-1}	3.97×10^{-46}	5.00×10^{-2}
	Std.Dev	0	2.88×10^{-1}	1.19×10^{-45}	2.23×10^{-1}
	Ratio	100%	100%	100%	95%
$k_D = 0.155$	Min	3.16×10^{-317}	3.24×10^{-96}	4.45×10^{-74}	0
	Mean	5.01×10^{-294}	6.39×10^{-66}	8.02×10^{-65}	0
	Std.Dev	0	2.85×10^{-65}	3.16×10^{-64}	0
	Ratio	100%	100%	100%	100%
$k_D = 0.255$	Min	6.59×10^{-109}	2.35×10^{-32}	3.09×10^{-6}	0
	Mean	7.04×10^{-102}	2.35×10^{-32}	5.65×10^{-4}	0
	Std.Dev	2.82×10^{-101}	2.80×10^{-48}	8.69×10^{-4}	0
	Ratio	100%	100%	100%	100%
$k_D = 0.355$	Min	1.75×10^{-14}	3.44×10^{-18}	4.83×10^{-1}	0
	Mean	2.60×10^{-11}	3.05×10^{-13}	1.04	0
	Std.Dev	4.00×10^{-11}	8.51×10^{-13}	3.94×10^{-1}	0
	Ratio	100%	100%	0%	100%
$k_D = 0.455$	Min	6.65	8.68×10^{-3}	3.46	3.00
	Mean	1.94×10^{1}	1.70×10^{-1}	5.07	2.22×10^{1}
	Std.Dev	9.32	2.20×10^{-1}	7.64×10^{-1}	1.29×10^{1}
	Ratio	0%	100%	0%	0%
$k_D = 0.555$	Min	5.92×10^{1}	3.15×10^{-1}	5.39	3.30×10^{1}
	Mean	1.45×10^{2}	2.24	7.92	1.14×10^{2}
	Std.Dev	3.87×10^{1}	1.10	1.16	6.94×10^{1}
	Ratio	0%	100%	0%	0%
$k_D = 0.653$	Min	7.75×10^{1}	9.13×10^{-1}	4.80	7.80×10^{1}
	Mean	1.30×10^{2}	2.10	6.58	1.34×10^{2}
	Std.Dev	3.57×10^{1}	6.83×10^{-1}	1.01	3.11×10^{1}
	Ratio	0%	100%	0%	0%

Table 2 The test function configuration

Functions	Name	Dimension	Search space	Threshold	Minimum
$f_1(x)$	Sphere	20	$[-100\quad 100]$	0.01	0
$f_2(x)$	Penalized1	20	$[-50\quad 50]$	0.01	0
$f_3(x)$	Schwefel2.21	20	$[-100\quad 100]$	0.01	0
$f_4(x)$	Step	20	$[-100\quad 100]$	0.01	0

Table 3 Statistical results of six PSO algorithms

		PSO	PSO-LDIW	PSO-TVAC	PSO-CK	SDPSO	PIDLPSO
$f_1(x)$	Min	8.15×10^{-23}	2.29×10^{-205}	5.26×10^{-139}	0	1.10×10^{-12}	3.16×10^{-317}
	Mean	8.20	7.61×10^{-188}	1.07×10^{-82}	0	4.11×10^{-9}	5.01×10^{-294}
	Std.Dev	4.09×10^{-16}	0	5.86×10^{-82}	0	1.17×10^{-8}	0
	Ratio	100%	100%	100%	100%	100%	100%
$f_2(x)$	Min	1.10×10^{-18}	3.95×10^{-119}	3.03×10^{-40}	5.74×10^{-56}	1.56×10^{-7}	3.24×10^{-96}
	Mean	8.20	1.42×10^{1}	1.59×10^{-20}	4.20	2.80	6.39×10^{-66}
	Std.Dev	8.96	1.34×10^{1}	5.90×10^{-20}	6.09	5.73	2.85×10^{-65}
	Ratio	44%	36%	100%	64%	78%	100%
$f_3(x)$	Min	2.68×10^{-1}	8.03×10^{-28}	6.31×10^{-29}	3.56×10^{-105}	2.02×10^{-1}	4.45×10^{-74}
	Mean	1.67×10^{3}	2.86×10^{3}	1.00×10^{2}	1.80×10^{3}	2.05×10^{2}	8.02×10^{-65}
	Std.Dev	2.85×10^{3}	3.98×10^{3}	7.07×10^{2}	3.31×10^{3}	9.88×10^{2}	3.16×10^{-64}
	Ratio	0%	56%	98%	72%	0%	100%
$f_4(x)$	Min	1.49×10^{-24}	2.35×10^{-32}	2.35×10^{-32}	2.35×10^{-32}	6.93×10^{-14}	0
	Mean	5.08×10^{-19}	2.35×10^{-32}	3.11×10^{-3}	2.92×10^{-1}	1.27×10^{-8}	0
	Std.Dev	2.10×10^{-18}	8.29×10^{-48}	2.19×10^{-2}	6.31×10^{-1}	7.53×10^{-8}	0
	Ratio	100%	100%	98%	50%	100%	100%

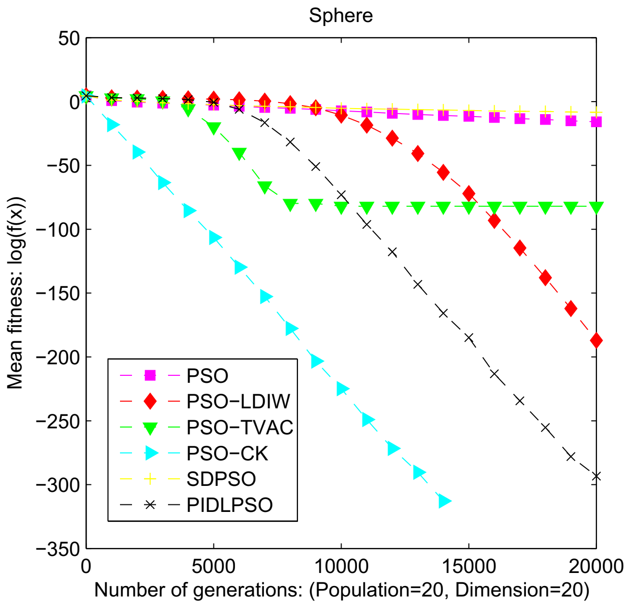

Fig. 4 PSO algorithms convergence characteristics of Sphere

rate, its fitness values are small enough to satisfy the convergence performance. In general, the convergence rate of the PIDLPSO algorithm is more excellent than that of others.

Step response

In essence, the particle swarm optimizer could be approximately regarded as the PI strategy, similarly, the proposed PIDLPSO algorithm could be considered as a PID strategy in this paper. The step response curves of the two methods are shown in Fig. 8. It can be clearly seen that, compared with the typical PSO algorithm, the proposed PIDLPSO algorithm provides better control performance and reduce the overshoot problem. It owes the introduction of derivative control, which expands the search space of the particles and increases the probability of escaping from the local optimum.

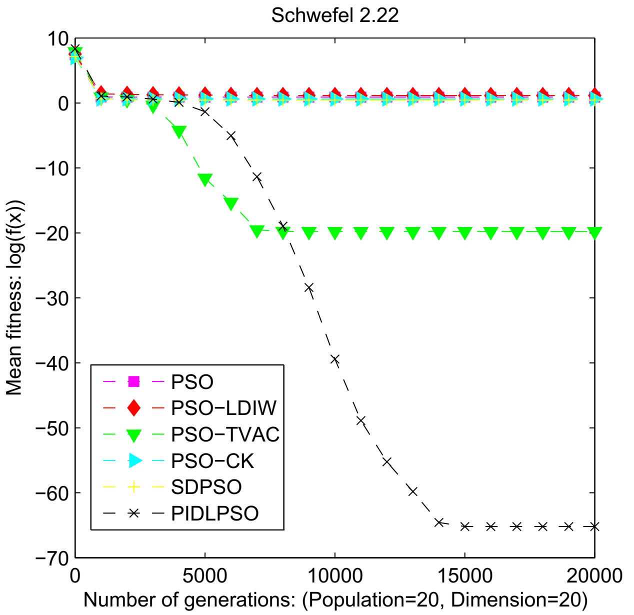

Fig. 5 PSO algorithms convergence characteristics of Schwefel 2.22

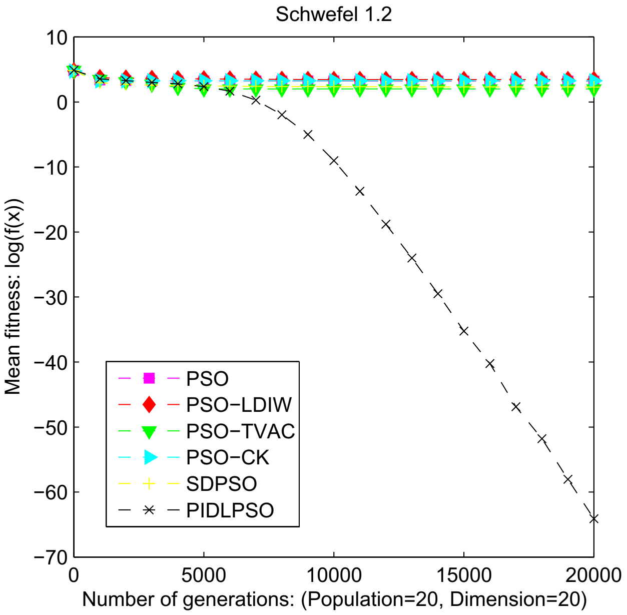

Fig. 6 PSO algorithms convergence characteristics of Schwefel 1.2

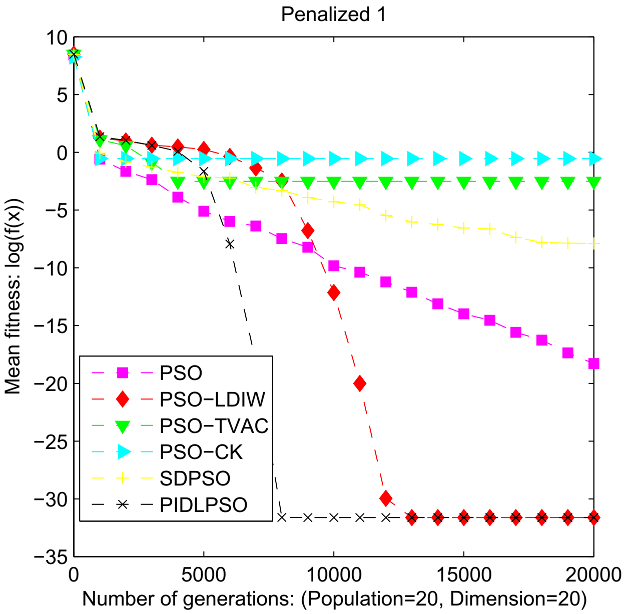

Fig. 7 PSO algorithms convergence characteristics of Penalized 1

Theoretical simulation

According to Theorem 1, all the particles will eventually converge to

$$x_i(\infty) = g_{\text{best}}. \tag{22}$$

Figures 9 and 10 plot the tendencies of $x_i(k)$ and g_{best} when the maximum number of iterations is 4000 and 20000, respectively. It can be seen from the figures that, as the number of iterations increases, the two curves of $x_i(k)$ and g_{best} will coincide, that is, the particles will converge.

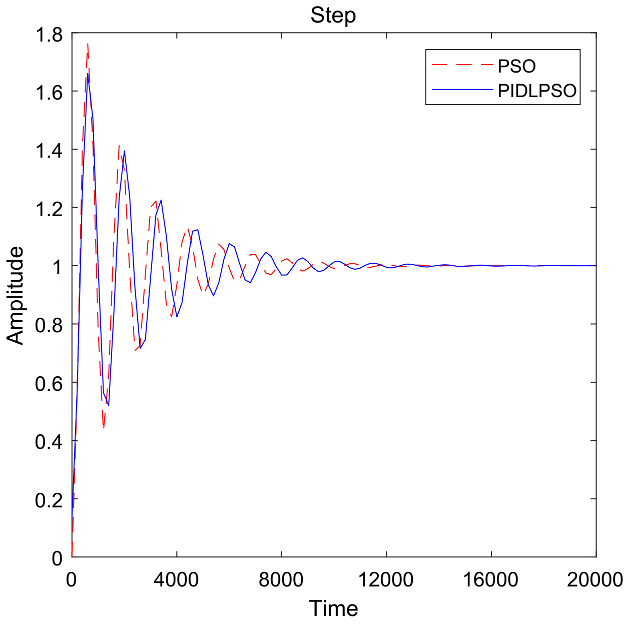

Fig. 8 The step response curve of the algorithms

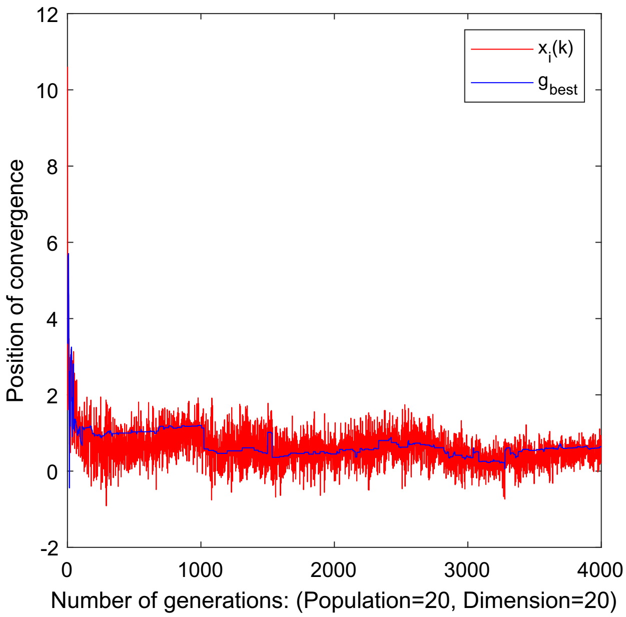

Fig. 9 The convergence curve of the PIDLPSO algorithm, when the maximum number of iterations is 4000

Conclusion

In this paper, motivated by the similarity in traditional PSO algorithm and PI control strategies, a novel PIDLPSO algorithm has been designed by introducing the derivative

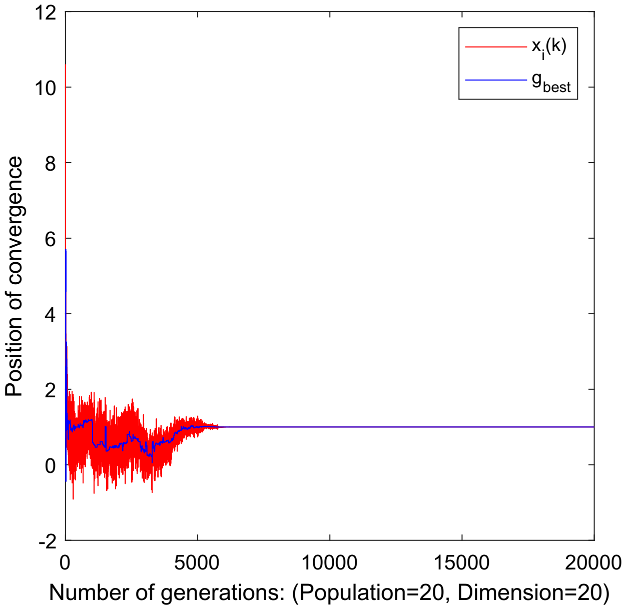

Fig. 10 The convergence curve of the PIDLPSO algorithm, when the maximum number of iterations is 20000

item with the purpose of alleviating the overshoot problem caused by past momentum. With this novel tactics, the PIDLPSO algorithm has not only maintained the accuracy of the solution but also enhanced the convergence rate. Furthermore, with the help of Routh stability criterion and final value theorem of the Z-transformation, the convergence conditions and the final positions have been gained for the PIDLPSO algorithm. The superiorities of proposed algorithm have been evaluated from the perspectives of population diversity, convergence rate and searching ability. Experimental results have exhibited the superiorities of designed PIDLPSO algorithm over other state-of-the-art PSO variants on four wide-ranging benchmark functions including both one-peak and multi-peak cases. In the future, we will research into some new directions which include, but are not limited to, the investigations on (1) how to analyze the convergence of the modified PSO algorithms with time-varying parameters [55] and (2) how to apply the PIDLPSO algorithm to other research fields such as deep learning [10,15,26,38,43,54,58], fault detection [3,11,12], signal processing [23,25,27,34,35,37,40,45,47,51] and multi-objective optimization [9,53].

Declarations

Conflict of interest On behalf of all authors, the corresponding author states that there is no conflict of interest.

References

1. Amin M, Saeid G, Seyed H, Behnam ZG (2020) Application of neural network and weighted improved PSO for uncertainty modeling and optimal allocating of renewable energies along with battery energy storage. Appl Soft Comput 88:105979
2. Ang KH, Chong G, Li Y (2005) PID control system analysis, design, and technology. IEEE Trans Control Syst Technol 3(4):559–576
3. Chen W, Hu J, Wu Z, Yu X, Chen D (2020) Finite-time memory fault detection filter design for nonlinear discrete systems with deception attacks. Int J Syst Sci 51(8):1464–1481
4. Chen Y, Chen Z, Chen Z, Xue A (2020) Observer-based passive control of non-homogeneous Markov jump systems with random communication delays. Int J Syst Sci 51(6):1133–1147
5. Dereli S, Köker R (2021) Strengthening the PSO algorithm with a new technique inspired by the golf game and solving the complex engineering problem. Complex Intell Syst 7:1515–1526
6. Dong Y, Song Y, Wei G (2021) Efficient model-predictive control for networked interval type-2 T-S fuzzy system with stochastic communication protocol. IEEE Trans Fuzzy Syst 29(2):286–297
7. Geng H, Liang Y, Cheng Y (2020) Target state and Markovian jump ionospheric height bias estimation for OTHR tracking systems. IEEE Trans Syst Man Cybern Syst 50(7):2599–2611
8. Guan Q, Wei G, Wang L, Song Y (2021) A novel feature points tracking algorithm in terms of IMU-aided information fusion. IEEE Trans Industr Inf 17(8):5304–5313
9. Hu L, Naeem W, Rajabally E, Watson G, Mills T, Bhuiyan Z, Raeburn C, Salter I, Pekcan C (2020) A multiobjective optimization approach for COLREGs-compliant path planning of autonomous surface vehicles verified on networked bridge simulators. IEEE Trans Intell Transp Syst 21(3):1167–1179
10. Huang C, Lan Y, Xu G, Zhai X, Wu J, Lin F, Zeng N, Hong Q, Ng EYK, Peng Y, Chen F, Zhang G (2020) A deep segmentation network of multi-scale feature fusion based on attention mechanism for IVOCT lumen contour. IEEE/ACM Trans Comput Biol Bioinf 18(1):62–69
11. Ju Y, Liu Y, He X, Zhang B (2021) Finite-horizon H_∞ filtering and fault isolation for a class of time-varying systems with sensor saturation. Int J Syst Sci 52(2):321–333
12. Ju Y, Wei G, Ding D, Zhang S (2019) Fault detection for discrete time-delay networked systems with round-robin protocol in finite-frequency domain. Int J Syst Sci 50(13):2497–2509
13. Juan LF, Esperanza G (2011) Stochastic stability analysis of the linear continuous and discrete PSO models. IEEE Trans Evol Comput 15(3):405–423
14. Kadirkamanathan V, Selvarajah K, Fleming PJ (2006) Stability analysis of the particle dynamics in particle swarm optimizer. IEEE Trans Evol Comput 10(3):245–255
15. Ke L, Zhang Y, Yang B, Luo Z, Liu Z (2021) Fault diagnosis with synchrosqueezing transform and optimized deep convolutional

neural network: An application in modular multilevel converters. Neurocomputing 430:24–33

16. Kennedy J, Eberhart RC (1995) Particle swarm optimization. In: Proceedings of the 1995 IEEE International Conference on Neural Networks, Perth, pp 1942–1948
17. Li N, Li Q, Suo J (2021) Dynamic event-triggered H_∞ state estimation for delayed complex networks with randomly occurring nonlinearities. Neurocomputing 421:97–104
18. Li Q, Liang J (2020) Dissipativity of the stochastic Markovian switching CVNNs with randomly occurring uncertainties and general uncertain transition rates. Int J Syst Sci 51(6):1102–1118
19. Liu Y, Chen S, Guan B, Xu P (2019) Layout optimization of large-scale oil-gas gathering system based on combined optimization strategy. Neurcomputing 332(7):159–183
20. Liu Y, Liu J, Jin Y (2021) Surrogate-assisted multipopulation particle swarm optimizer for high-dimensional expensive optimization. IEEE Trans Syst Man Cybern Syst (in press). https://doi.org/10.1109/TSMC.2021.3102298
21. Liu Y, Shen B, Li Q (2019) State estimation for neural networks with Markov-based nonuniform sampling: The partly unknown transition probability case. Neurocomputing 357:261–270
22. Luo X, Liu Z, Jin L, Zhou Y, Zhou M (2021) Symmetric nonnegative matrix factorization-based community detection models and their convergence analysis. IEEE Trans Neural Netw Learn Syst (in press). https://doi.org/10.1109/TNNLS.2020.3041360
23. Mao J, Ding D, Wei G, Liu H (2019) Networked recursive filtering for time-delayed nonlinear stochastic systems with uniform quantisation under Round-Robin protocol. Int J Syst Sci 50(4):871–884
24. Ozcan E, Mohan CK (1998) Analysis of a simplified particle swarm optimization problem. In: Intelligent engineering systems trough artificial neural networks, New York, USA, pp 253–258
25. Qian W, Li Y, Chen Y, Liu W (2020) L_2-L_∞ filtering for stochastic delayed systems with randomly occurring nonlinearities and sensor saturation. Int J Syst Sci 51(13):2360–2377
26. Qian W, Li Y, Zhao Y, Chen Y (2020) New optimal method for L_2-L_∞ state estimation of delayed neural networks. Neurocomputing 415:258–265
27. Qu B, Shen B, Shen Y, Li Q (2020) Dynamic state estimation for islanded microgrids with multiple fading measurements. Neurocomputing 406:196–203
28. Rahman IU, Wang Z, Liu W, Ye B, Zakarya M, Liu X (2019) An N-State markovian jumping particle swarm optimization algorithm. IEEE Trans Syst Man Cybern Syst (in press). https://doi.org/10.1109/TSMC.2019.2958550
29. Ran C, Jin Y (2014) A competitive swarm optimizer for large scale optimization. IEEE Trans Cybern 45(2):191–204
30. Ranaee V, Ebrahimzadeh A, Ghaderi R (2010) Application of the PSO-SVM model for recognition of control chart patterns. ISA Trans 49(4):577–586
31. Ratnaweera A, Halgamuge SK, Watson HC (2004) Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients. IEEE Trans Evol Comput 8(3):240–255
32. Rosendo M, Pozo A (2010) A hybrid particle swarm optimization algorithm for combinatorial optimization problems. In: IEEE congress on evolutionary computation, Barcelona, Spain, pp 1–8
33. Shi Y, Eberhart RC (1999) Empirical study of particle swarm optimization. In: Proceedings of the 1999 IEEE congress on evolutionary computation, Washington, DC, USA, pp 1945–1950
34. Song G, Shi P, Wang S, Pan J-S (2019) A new finite-time cooperative control algorithm for uncertain multi-agent systems. Int J Syst Sci 50(5):1006–1016
35. Song J, Ding D, Liu H, Wang X (2020) Non-fragile distributed state estimation over sensor networks subject to DoS attacks: the almost sure stability. Int J Syst Sci 51(6):1119–1132
36. Song Q, Chen Y, Zhao Z, Liu Y, Alsaadi FE (2021) Robust stability of fractional-order quaternion-valued neural networks with neutral delays and parameter uncertainties. Neurocomputing 420:70–81
37. Tan H, Shen B, Peng K, Liu H (2020) Robust recursive filtering for uncertain stochastic systems with amplify-and-forward relays. Int J Syst Sci 51(7):1188–1199
38. Tang Z, Tian E, Wang Y, Wang L, Yang T (2021) Nondestructive defect detection in castings by using spatial attention bilinear convolutional neural network. IEEE Trans Industr Inf 17(1):82–89
39. Trelea IC (2003) The particle swarm optimization algorithm: convergence analysis and parameter selection. Inf Process Lett 85:317–325
40. Wan X, Fang Z, Wu M, Du Y (2020) Automatic detection of HFOs based on singular value decomposition and improved fuzzy c-means clustering for localization of seizure onset zones. Neurocomputing 400:1–10
41. Wang C, Han F, Zhang Y, Lu J (2020) An SAE-based resampling SVM ensemble learning paradigm for pipeline leakage detection. Neurocomputing 403:237–246
42. Wang C, Zhang Y, Song J, Liu Q, Dong H (2019) A novel optimized SVM algorithm based on PSO with saturation and mixed time-delays for classification of oil pipeline leak detection. Syst Sci Control Eng 7(1):75–88
43. Wang J, Zhang S, Wang Y, Zhu Z (2021) Learning efficient multi-task stereo matching network with richer feature information. Neurocomputing 421:151–160
44. Wang X, Zhang K, Wang J, Jin Y (2021) An enhanced competitive swarm optimizer with strongly convex sparse operator for large-scale multi-objective optimization. IEEE Trans Evolut Comput (in press). https://doi.org/10.1109/TEVC.2021.3111209
45. Wei G, Liu L, Wang L, Ding D (2020) Event-triggered control for discrete-time systems with unknown nonlinearities: an interval observer-based approach. Int J Syst Sci 51(6):1019–1031
46. Wu M, Wan T, Wan X, Fang Z, Du Y (2021) A new localization method for epileptic seizure onset zones based on time-frequency and clustering analysis. Pattern Recognit vol 111, art. no. 107687
47. Wu Q, Song Q, Zhao Z, Liu Y, Alsaadi FE (2021) Stabilization of T-S fuzzy fractional rectangular descriptor time-delay system. Int J Syst Sci 52(11):2268–2282
48. Xiang Z, Shao X, Wu H, Ji D, Yu F, Li Y (2020) An adaptive integral separated proportional-integral controller based strategy for particle swarm optimization. Knowl-Based Syst vol. 195, art. no. 105696
49. Xiang Z, Ji D, Zhang H, Wu H, Li Y (2019) A simple PID-based strategy for particle swarm optimization algorithm. Inf Sci 502:558–574
50. Yan L, Zhang S, Wei G, Liu S (2019) Event-triggered set-membership filtering for discrete-time memristive neural networks subject to measurement saturation and fadings. Neurocomputing 346:20–29
51. Yang Y, Tan J, Yue D, Xie X, Yue W (2021) Observer-based containment control for a class of nonlinear multiagent systems with uncertainties. IEEE Trans Syst Man Cybern Syst 51(1):588–600
52. Ying H (2000) Theory and application of a novel fuzzy PID controller using a simplified Takagi-Sugeno rule scheme. Inf Sci 123(3):281–293
53. Zeng N, Song D, Li H, You Y, Liu Y, Alsaadi FE (2021) A competitive mechanism integrated multi-objective whale optimization algorithm with differential evolution. Neurocomputing 432:170–182
54. Zeng N, Zhang H, Song B, Liu W, Li Y, Dobaie AM (2018) Facial expression recognition via learning deep sparse autoencoders. Neurocomputing 273:643–649
55. Zeng N, Zhang H, Liu W, Liang J, Alsaadi FE (2017) A switching delayed PSO optimized extreme learning machine for short-term load forecasting. Neurocomputing 240:175–182

56. Zhang B, Song Y (2020) Asynchronous constrained resilient robust model predictive control for Markovian jump systems. IEEE Trans Ind Inf 16(11):7025–7034
57. Zhang L, Guo G (2019) Control of a group of systems whose communication channels are assigned by a semi-Markov process. Int J Syst Sci 50(12):2306–2315
58. Zheng Y, Zhou W, Yang W, Liu L, Liu Y, Zhang Y (2021) Multivariate/minor fault diagnosis with severity level based on Bayesian decision theory and multidimensional RBC. J Process Control 101:68–77

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.