

PID Neural Network Decoupling Control Based on Hybrid Particle Swarm Optimization and Differential Evolution

Hong-Tao Ye^{1,2} Zhen-Qiang Li^{1,2}

¹School of Electrical and Information Engineering, Guangxi University of Science and Technology, Liuzhou 545006, China

²Guangxi Key Laboratory of Automobile Components and Vehicle Technology, Guangxi University of Science and Technology, Liuzhou 545006, China

Abstract: For complex systems with high nonlinearity and strong coupling, the decoupling control technology based on proportion integration differentiation (PID) neural network (PIDNN) is used to eliminate the coupling between loops. The connection weights of the PIDNN are easy to fall into local optimum due to the use of the gradient descent learning method. In order to solve this problem, a hybrid particle swarm optimization (PSO) and differential evolution (DE) algorithm (PSO-DE) is proposed for optimizing the connection weights of the PIDNN. The DE algorithm is employed as an acceleration operation to help the swarm to get out of local optima traps in case that the optimal result has not been improved after several iterations. Two multivariable controlled plants with strong coupling between input and output pairs are employed to demonstrate the effectiveness of the proposed method. Simulation results show that the proposed method has better decoupling capabilities and control quality than the previous approaches.

Keywords: Particle swarm optimization, differential evolution, proportion integration differentiation (PID) neural network, hybrid approach, decoupling control.

1 Introduction

Proportion integration differentiation (PID) control is a loop feedback mechanism widely used in industrial control systems. The controller attempts to minimize the error between a measured process variable and a desired set-point by adjusting the process through use of a manipulated variable. It is easy to combine PID control with other algorithms such as fuzzy and neural network^[1–3]. PID neural network (PIDNN)^[4] is a new kind of networks and its hidden layer neurons simply work as PID controller terms through their activation functions. Thus, it simultaneously utilizes advantages of both PID controller and neural structure. The connection weights of PIDNN are vital to the system performance. However, the connection weights are easy to fall into local optimum due to the use of gradient descent learning method. Some researchers are interested in optimization of the weights of PIDNN with meta-heuristic^[5] algorithms including genetic algorithm (GA)^[6], fish swarm algorithm^[7], particle swarm optimization (PSO)^[8,9], and chaotic PSO^[10]. These methods have better quality than the traditional PID decoupling control method.

The PSO algorithm is an optimization technique developed by Kennedy and Eberhart^[11] in 1995. It is inspired by the social behavior of bird flocking or fish schooling. The PSO is capable of handling non-differentiable, discontinuous and multimodal objective functions. It has become a

popular algorithm due to its relative simplicity and quick convergence. Similar to GA, PSO is also a population-based optimization tool, which searches for optima by updating generations^[12]. Each particle adjusts its trajectory towards its own previous best position and towards the current best position attained by any other member in its neighbor^[13]. However, the main disadvantage of PSO is the risk of a premature search convergence, especially in complex multi-peak-search problems^[14].

To improve the performance and the convergence behavior of PSO, hybridization of PSO with differential evolution (DE) algorithm has garnered significant attention in the research literature^[15–17]. These approaches aim to aggregate the advantages of both algorithms to efficiently tackle the optimization problem. DE^[18] is a branch of evolutionary algorithms for optimization problems over continuous domains. Like other evolutionary algorithms, DE is a population-based stochastic search algorithm. DE has the distinguishing advantages of computation simplicity as well as convergence efficiency^[19]. It is relatively immune to differences in initial populations than one-point optimizers. Because it is a direct search method, DE is versatile enough to solve problems whose objective functions lack the analytical description needed to compute gradients. There are 3 crucial parameters in classical DE algorithms: 1) the population size NP , 2) the mutation scale factor F , 3) the crossover rate C_r .

In this paper, to exploit the advantages of PSO in fast exploitation and the exploration ability of DE^[20], we will combine these 2 global optimization algorithms, and propose the hybrid algorithm PSO-DE. Unlike the existed hybrid PSO/DE algorithms, the hybrid PSO-DE employs DE

Research Article
Manuscript received May 15, 2014; accepted November 25, 2014;
published online November 6, 2015
Recommended by Associate Editor Chandrasekhar Kambhampati
© Institute of Automation, Chinese Academy of Sciences and
Springer-Verlag Berlin Heidelberg 2016

to accelerate the convergence of the algorithm in case that the optimal result has not improved after several iterations. The PSO-DE is used for optimizing the connection weights of PIDNN. Simulations have demonstrated that the proposed method can be effectively used to solve optimization problems of PIDNN decoupling control.

The remainder of this paper is organized as follows. Section 2 briefly describes the basic operations of the canonical PSO and the DE algorithms. Section 3 presents the hybrid method, namely PSO-DE, to optimize PID neural network decoupling control. Section 4 reveals the simulations and analysis. Finally, conclusions are given in Section 5.

2 The basic PSO and DE algorithms

2.1 The particle swarm optimization

In PSO, each particle consists of a position vector x_n which represents a candidate solution of the optimization problem, a velocity vector v_n and a memory vector $pbest_n$ which is the best candidate solution encountered by the particle. Each particle position is modified through iterations with the aim of finding the optimum position, where an optimum value for the fitness function or optimum state is achieved. Thus, at every iteration ($g+1$), the velocity and position of the particle are updated by

$$v_{n,g+1} = wv_{n,g} + c_1r_1(pbest_{n,g} - x_{n,g}) + c_2r_2(pbest_g - x_{n,g}) \quad (1)$$

$$x_{n,g+1} = x_{n,g} + v_{n,g} \quad (2)$$

where w is the inertia weight, which determines how much of the previous velocity of the particle is preserved. c_1 and c_2 are positive constants. r_1 and r_2 are two uniformly distributed random numbers in the interval $[0, 1]$. $gbest$ represents the best position achieved by any member of the population.

In this paper, we consider the canonical PSO version proposed by Clerc and Kennedy^[12], which incorporates the parameter χ , called the constriction factor. The constriction factor is used to control the magnitude of the velocities and alleviate the swarm explosion effect that sometimes prevents the convergence of the original PSO. Then, the velocity adjustment is executed by the equation as

$$v_{n,g+1} = \chi(wv_{n,g} + c_1r_1(pbest_{n,g} - x_{n,g}) + c_2r_2(pbest_g - x_{n,g})) \quad (3)$$

where $\chi = \frac{2k}{|2-\varphi-(\varphi^2-4\varphi)^{\frac{1}{2}}|}$ with $\varphi = c_1 + c_2 > 4$. The aforementioned scheme is typically utilized for the constant $\varphi = 4.1$, with $\chi = 0.72984$ and $c_1 = c_2 = 2.05$ ^[12, 21].

2.2 The differential evolution algorithm

The basic strategy of DE can be described as follows.

Initialization. DE is a parallel direct search method. It begins with a randomly initiated population of N D -

dimensional parameter vectors $x_{i,g}$, $i = 1, 2, \dots, N$ as a population for each generation. The initial population ($g = 0$) of the j -th parameter of the i -th vector is

$$x_{j,i,0} = x_{j,\min} + \text{rand}_{i,j}[0, 1](x_{j,\max} - x_{j,\min}) \quad (4)$$

where $x_{j,\min}$ and $x_{j,\max}$ are the lower and upper bounds, respectively. $\text{rand}_{i,j}[0, 1]$ is a uniformly distributed random number lying between 0 and 1.

Mutation. DE mutates and recombines the population to produce a population of N trial vectors. Specifically, for each individual $x_{i,g}$, a mutant vector $v_{i,g}$ is generated according to

$$v_{i,g} = x_{r_1^i,g} + F(x_{r_2^i,g} - x_{r_3^i,g}) \quad (5)$$

where F , commonly known as scale factor, is a positive real number. Three other individuals $x_{r_1^i,g}$, $x_{r_2^i,g}$, $x_{r_3^i,g}$ are sampled randomly from the current population. The mutation strategy described above is known as DE/rand/1.

Crossover. To complement the differential mutation search strategy, DE adopts a crossover operation, often referred to as discrete recombination. In particular, DE crosses each vector with a mutant vector.

$$u_{j,i,g} = \begin{cases} v_{j,i,g}, & \text{if } (\text{rand}_{i,j}[0, 1] \leq c_r \text{ or } j = j_{\text{rand}}) \\ x_{j,i,g}, & \text{otherwise} \end{cases} \quad (6)$$

where c_r is called the crossover rate.

Selection. To decide whether or not it should become a member of generation $g+1$, the trial vector $u_{i,j}$ is compared to the target vector $x_{i,g}$ using the greedy criterion. The selection operation is described as

$$x_{i,g+1} = \begin{cases} u_{i,g}, & \text{if } f(u_{i,g}) \leq f(x_{i,g}) \\ x_{i,g}, & \text{otherwise} \end{cases} \quad (7)$$

where $f(x)$ is the objective function to be minimized.

3 PSO-DE in optimizing PID neural network decoupling control

3.1 PID neural network decoupling control system

PIDNN is a new type of dynamic feed-forward network, and the output functions of the neurons in its hidden layer are different from each other, which are proportional (P) function, integral (I) function and differential (D) function so they are named as P-neuron, I-neuron and D-neuron, respectively. The input layer has 2 P neurons. One receives system setting input, and another connects system output. The output layer has only one neuron which completes the control output duty. The output layer completes the synthesis of PID control law and forms the input of the controlled objective. PIDNN completes system decoupling

and control by adjusting the connection weights. The decoupling and control ability comes from its nonlinear mapping property and PID style processing of the hidden layer. The basic structure of PIDNN decoupling control system is shown in Fig. 1.

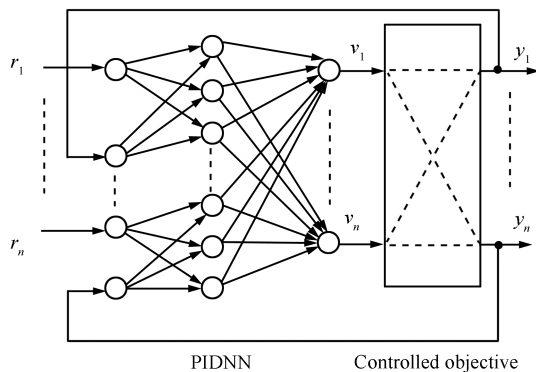


Fig. 1 Basic structure of PIDNN decoupling control system

The PIDNN consists of input layer, hidden layer, and output layer. The forward algorithm of PIDNN at any sampling time k for each layer is calculated as follows.

1) Input layer is

$$x_{si}(k) = u_{si}(k) \quad (8)$$

where $u_{si}(k)$ ($i=1, 2$) is input value of input layer neurons, x_{si} ($i=1, 2$) is output value of input layer neurons, and s ($s=1, 2, \dots, n$) is the number of controlled variables.

$$u_{s1}(k) = r_s(k) \quad (9)$$

$$u_{s2}(k) = y_s(k). \quad (10)$$

2) Hidden layer is

$$u'_{sj}(k) = \sum_{i=1}^2 w_{sij} x_{si}(k) \quad (11)$$

where $u'_{sj}(k)$ is input value of hidden layer neurons, and w_{sij} is net weight from input layer to hidden layer.

The function of P-neuron is the same as that of the input layer.

$$x'_{s1}(k) = u'_{s1}(k). \quad (12)$$

The function of I-neuron is

$$x'_{s2}(k) = u'_{s2}(k) + x'_{s2}(k-1). \quad (13)$$

The function of D-neuron is

$$x'_{s3}(k) = u'_{s3}(k) - u'_{s3}(k-1). \quad (14)$$

3) Output layer is

$$v_h(k) = \sum_{s=1}^n \sum_{j=1}^3 w'_{sjh} x'_{sj}(k) \quad (15)$$

where $v_h(k)$ is output value of output layer neurons, and w'_{sjh} is net weight from hidden layer to output layer.

3.2 PID neural network decoupling control based on hybrid PSO-DE

The main idea of the hybrid PSO-DE algorithm is to integrate the DE operators into the PSO, and thus increasing the diversity of the population and the ability of escaping from local minima. The swarm may be damped to equilibrium state. For an extreme case, if the particles have the same location and all in zero velocities at certain evolution stage, then the swarm is in stationary equilibrium with no possibility to evolution. If the swarm is going to be in equilibrium, the evolution process will be stagnated as time goes on. To prevent the trend, if the stagnating step of evolution process g_0 is larger than threshold value G_0 , this is incorrect, perhaps you mean: the particles perform as DE operators. In this paper, we utilize only the DE/rand/1/bin strategy. The initial connection weights of PIDNN are received at random. In order to avoid getting into a local minimum, the connection weights of the PIDNN are optimized by the PSO-DE algorithm. The stepwise operation can be described as follows.

Step 1. Initialization of particles in swarm. The PSO generates an initial population P with N (population size) particles and sets the current generation number $g = 1$. Each particle represents a weight of the PIDNN.

Step 2. Initial evaluation of fitness function. The fitness function is defined as the controlled variable error J of PIDNN decoupling control system and calculated as

$$J = \frac{1}{l} \sum_{p=1}^n \sum_{k=1}^l [r_p(k) - y_p(k)]^2 \quad (16)$$

where $r_p(k)$ and $y_p(k)$ represent the set value and actual value, respectively, and l is the number of samples.

Step 3. Updating of $pbest_{n,g}$ and $gbest_g$. For each particle $x_{n,g}$ in the swarm, if the fitness of $x_{n,g}$ is better than $pbest_{n,g}$, then $pbest_{n,g}$ is updated by $x_{n,g}$. Similarly, $pbest_{n,g}$ is updated accordingly if the new fitness function value is better than the previous.

Step 4. Updating of the position and velocity of the n -th particle according to (2) and (3).

Step 5. Evaluation of fitness function. After modification of the particle positions, the controlled variable error J is calculated.

Step 6. Judge the evolution process. If the stagnating step of evolution process g_0 is larger than threshold value G_0 , the particles perform as DE operators, otherwise go to Step 7. The 3 main DE evolution steps (mutation, crossover and selection) are applied to the best, please check if it is correct by (5)–(7).

Step 7. Update $pbest_{n,g}$ and $gbest_g$. The $pbest_{n,g}$ and $gbest_g$ values have to be updated according to the new fitness value. If the best position of all new particles is better than the current $gbest_g$, then $gbest_g$ is updated by the new solution. Similarly, $pbest_{n,g}$ is updated accordingly if the new fitness function value is better.

Step 8. Termination criteria. If the stopping criterion is met, then output the weights of the PIDNN, otherwise,

repeat Steps 2 to 7.

4 Algorithms' simulations and results

In this section, we present a simulation study to validate the proposed PIDNN decoupling control system based on the PSO-DE.

4.1 Test problem 1

The process to be controlled is a 3-input and 3-output coupling system described by

$$\begin{cases} y_1(k+1) = 0.3y_1(k) + 0.4y_2(k) + \frac{u_1(k)}{1+u_1^2(k)} + \\ \quad 0.4u_1^3(k) + 0.4u_2(k) \\ y_2(k+1) = 0.4y_2(k) + 0.4y_3(k) + \frac{u_2(k)}{1+u_2^2(k)} + \\ \quad 0.3u_2^3(k) + 0.2u_1(k) \\ y_3(k+1) = 0.2y_3(k) + 0.3y_1(k) + \frac{u_3(k)}{1+u_3^2(k)} + \\ \quad 0.2u_3^3(k) + 0.3u_2(k) \end{cases}$$

where $y = [y_1, y_2, y_3]^T$ and $u = [u_1, u_2, u_3]^T$ are system output and input vectors, respectively.

The controlled target y is set to $[0.7, 0.5, 0.4]^T$. The parameter settings used are the following. Population size: $N = 30$, maximal number of generations: $G = 500$, $c1 = c2 = 2.05$, $\chi = 0.72984$, $\varphi = 4.1$, $C_r = 0.9$, $F = 0.5$, $G_0 = 10$.

Output responses y_1, y_2 and y_3 of the coupling system with PIDNN decoupling controller based on PSO-DE (PSO-DE-PIDNN) are shown in Fig. 2. For comparison purpose, output responses using PIDNN decoupling controller based on PSO (PSO-PIDNN) are also given in the same figure. As shown in Fig. 2, the PSO-DE-PIDNN decoupling controller has a faster rise time compared to the PSO-PIDNN decoupling controller and there is almost no overshoot. The PSO-DE-PIDNN has properties of quick response, quick convergence and good stability than the PSO-PIDNN. The error J of the PSO-PIDNN and PSO-DE-PIDNN are shown in Fig. 3. It can be observed that the error of PSO-DE-PIDNN algorithm is less than PSO-PIDNN. The better performance of the PSO-DE-PIDNN for this study could be attributed to its greater ability to explore efficiently the search space with the aid of DE operators and thus enhancing the chances of finding the global optimum.

4.2 Test problem 2

Biochemical wastewater treatment is a complex system with high nonlinearity and strong coupling. The application of PIDNN decoupling control based on PSO-DE in wastewater treatment system will be discussed in this section.

In wastewater treatment system, the concentration of ammonia nitrogen and nitrate nitrogen reflects the system's internal nitrification and denitrification progress. Ammonia nitrogen and nitrate nitrogen couple to each other. Therefore, the decoupling is required to achieve nitrification

and denitrification process control and improve wastewater treatment efficiency.

According to activated sludge model No.1 (ASM1)^[22], ammonia nitrogen and nitrate nitrogen material balance equation is described as

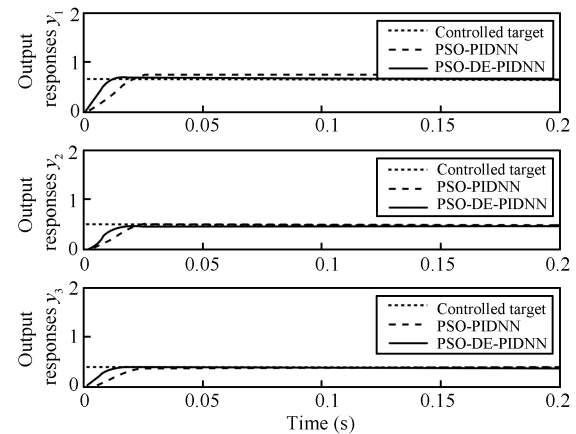


Fig. 2 Control performance of the PSO-PIDNN and PSO-DE-PIDNN

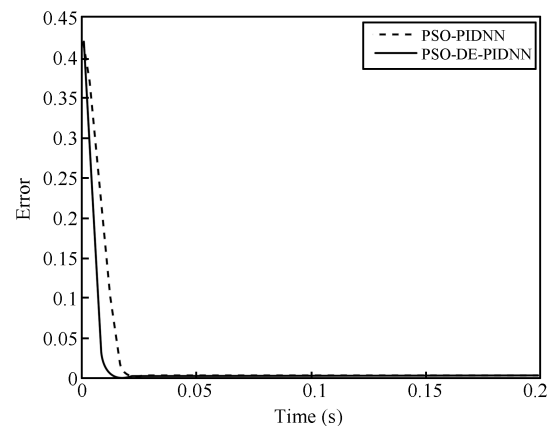


Fig. 3 Error curve of PSO and PSO-DE for PIDNN decoupling control

$$\begin{aligned} \frac{dS_{NH}}{dt} = & \left[-i_{XB}\hat{\mu}_H \left(\frac{S_S}{K_S + S_S} \right) \left\{ \left(\frac{S_O}{K_{OH} + S_O} \right) + \right. \right. \\ & \eta_g \left(\frac{K_{OH}}{K_{OH} + S_O} \right) \left(\frac{S_{NO}}{K_{NO} + S_{NO}} \right) \left. \right\} + \\ & k_a S_{NO} \left. \right] X_{BH} - \hat{\mu}_A \left(i_{XB} + \frac{1}{Y_A} \right) \\ & \left(\frac{S_{NH}}{K_{NH} + S_{NH}} \right) \left(\frac{S_O}{K_{OA} + S_O} \right) X_{BA} + \\ & \left(\frac{Q_a + Q_o + Q_r}{V_2} \right) (S_{NH,in} - S_{NH}) \end{aligned} \quad (17)$$

$$\begin{aligned} \frac{dS_{NO}}{dt} = & -\eta_g \hat{\mu}_H \left(\frac{1-Y_H}{2.86Y_H} \right) \left(\frac{S_S}{K_S+S_S} \right) \left(\frac{K_{OH}}{K_{OH}+S_O} \right) \\ & \left(\frac{S_{NO}}{K_{NO}+S_{NO}} \right) X_{BH} + \hat{\mu}_A \left(\frac{1}{Y_A} \right) \left(\frac{S_{NH}}{K_{NH}+S_{NH}} \right) \\ & \left(\frac{S_O}{K_{OA}+S_O} \right) X_{BA} + \\ & \left(\frac{Q_a+Q_o+Q_r}{V_2} \right) (S_{NO,in} - S_{NO}) \end{aligned} \quad (18)$$

where S_{NH} is ammonia nitrogen concentration, S_{NO} is nitrate nitrogen concentration, Q_a is inner circulation flow, Q_o is influent flow, Q_r is sludge recycle flow, S_S is easily biodegradable organic compounds concentration, $S_{NH,in}$ is ammonia nitrogen concentration of influent, $S_{NO,in}$ is nitrate nitrogen concentration of influent, and S_O is dissolved oxygen concentration.

The model of ammonia nitrogen and nitrate nitrogen is established by partial least squares regression^[23]. The inputs of the model include Q_a and S_O , and the outputs include S_{NH} and S_{NO} . The model can be described in discrete form as

$$S_{NH}(k) = -0.004S_O(k-1) - 0.014Q_a(k-1) + 0.49S_{NH}(k-1) + 0.46S_{NH}(k-2) \quad (19)$$

$$S_{NO}(k) = -0.005S_O(k-1) - 0.039Q_a(k-1) + 0.45S_{NO}(k-1) + 0.48S_{NO}(k-2). \quad (20)$$

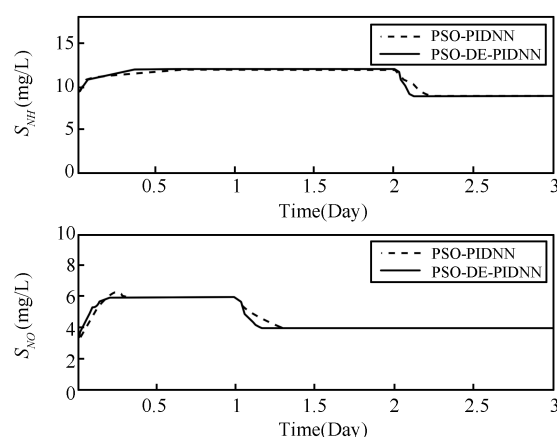


Fig. 4 Simulation curves of nitrate nitrogen and ammonia nitrogen

To compare the decoupling performance of PSO-DE-PIDNN and PSO-PIDNN, a simulation is established for concentration control of ammonia nitrogen and nitrate nitrogen. The initial value of S_{NO} is set to 6 mg/L, and S_{NH} is set to 12 mg/L. Then, S_{NO} is set to 4 mg/L when one day is passed and S_{NH} is set to 9 mg/L when two days are passed. The results are shown in Fig. 4. It can be observed

that the settling time of PSO-DE-PIDNN algorithm is less than PSO-PIDNN. The value of S_{NH} is not affected by the change of S_{NO} , and the value of S_{NO} is also not affected by the change of S_{NH} . The PSO-DE-PIDNN decoupling controller has better accuracy compared to the PSO-PIDNN decoupling controller.

5 Conclusions

In this paper, a new method named PSO-DE, which improves the performance of the PSO by incorporating DE, is proposed to solve optimization problems of PIDNN decoupling control. The PSO-DE is shown to outperform the canonical PSO in terms of the ability to find the optimum solution. The results of this study demonstrate that the proposed PSO-DE algorithm can be effectively used to solve optimization problems of PIDNN decoupling control.

Acknowledgements

This work was supported by the Key Project of Chinese Ministry of Education (No.212135), the Guangxi Natural Science Foundation (No.2012GXNSFBA053165), the Project of Education Department of Guangxi (No.201203YB131), and the Project of Guangxi Key Laboratory (No.14-045-44)

References

- [1] F. C. Liu, L. H. Liang, J. J. Gao. Fuzzy PID control of space manipulator for both ground alignment and space applications. *International Journal of Automation and Computing*, vol. 11, no. 4, pp. 353–360, 2014.
- [2] X. J. Jing, L. Cheng. An optimal PID control algorithm for training feedforward neural networks. *IEEE Transactions on Industrial Electronics*, vol. 60, no. 6, pp. 2273–2283, 2013.
- [3] Y. P. Pan, H. Y. Yu, M. J. Er. Adaptive neural PD control with semiglobal asymptotic stabilization guarantee. *IEEE Transactions on Neural Networks and Learning Systems*, vol. 25, no. 12, pp. 2264–2274, 2014.
- [4] H. L. Shu, Y. G. Pi. PID neural networks for time-delay systems. *Computers and Chemical Engineering*, vol. 24, no. 2–7, pp. 859–862, 2000.
- [5] L. Wang, D. Z. Zheng. Meta-heuristic algorithms: A review. *Control and Decision*, vol. 15, no. 3, pp. 257–262, 2000. (in Chinese)
- [6] G. D. Li, Q. C. Zhang, Y. C. Liang. GA-based PID neural network control for magnetic bearing systems. *Chinese Journal of Mechanical Engineering*, vol. 20, no. 2, pp. 56–59, 2007.
- [7] D. Q. Feng, C. Y. Ma, Y. H. Liu. Nonlinear control system of PID neural network based on artificial fish swarm algorithm. *Computer Simulation*, vol. 29, no. 9, pp. 247–296, 2012. (in Chinese)
- [8] J. S. Wang, F. W. Cong, Y. Zhang. PID-ANN decoupling controller of ball mill pulverizing system based on particle

- swarm optimization method. *Journal of Chemical Industry and Engineering*, vol. 59, no. 7, pp. 1743–1748, 2008. (in Chinese)
- [9] P. Y. Zhang, X. Q. Wu, H. Q. Yuan. Research on decoupling control by MPSO-PID neural network in MIMO system. *Industrial Control Computer*, vol. 25, no. 2, pp. 45–47, 2012. (in Chinese)
- [10] W. F. Teng, H. P. Pan, J. Ren. Neural network PID decoupling control based on chaos particle swarm optimization. In *Proceedings of the 33rd Chinese Control Conference*, IEEE, Nanjing, China, pp. 5017–5020, 2014.
- [11] J. Kennedy, R. C. Eberhart. Particle swarm optimization. In *Proceedings of International Conference on Neural Networks*, IEEE, Perth, USA, pp. 1942–1948, 1995.
- [12] M. Clerc, J. Kennedy. The particle swarm – Explosion, stability, and convergence in a multidimensional complex space. *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 1, pp. 58–73, 2002.
- [13] K. E. Parsopoulos, M. N. Vrahatis. Recent approaches to global optimization problems through particle swarm optimization. *Natural Computing*, vol. 1, no. 2–3, pp. 235–306, 2002.
- [14] A. Sedki, D. Ouazar. Hybrid particle swarm optimization and differential evolution for optimal design of water distribution systems. *Advanced Engineering Informatics*, vol. 26, no. 3, pp. 582–591, 2012.
- [15] H. Liu, Z. X. Cai, Y. Wang. Hybridizing particle swarm optimization with differential evolution for constrained numerical and engineering optimization. *Applied Soft Computing*, vol. 10, no. 2, pp. 629–640, 2010.
- [16] B. Xin, J. Chen, J. Zhang, H. Fang, Z. H. Peng. Hybridizing differential evolution and particle swarm optimization to design powerful optimizers: A review and taxonomy. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, vol. 42, no. 5, pp. 744–767, 2012.
- [17] C. S. Zhang, J. X. Ning, S. Lua, D. T. Ouyang, T. N. Ding. A novel hybrid differential evolution and particle swarm optimization algorithm for unconstrained optimization. *Operations Research Letters*, vol. 37, no. 2, pp. 117–122, 2009.
- [18] K. V. Price, R. M. Storn, J. A. Lampinen. *Differential Evolution: A Practical Approach to Global Optimization*, Berlin, Germany: Springer, 2005.
- [19] X. Z. Gao, X. Wang, S. J. Ovaska. Fusion of clonal selection algorithm and differential evolution method in training cascade-correlation neural network. *Neurocomputing*, vol. 72, no. 10–12, pp. 2483–2490, 2009.
- [20] J. Chen, B. Xin, Z. H. Peng, L. H. Dou, J. Zhang. Optimal contraction theorem for exploration-exploitation trade-off in search and optimization. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, vol. 39, no. 3, pp. 680–691, 2009.
- [21] D. Bratton, J. Kennedy. Defining a standard for particle swarm optimization. In *Proceedings of Swarm Intelligence Symposium*, IEEE, Honolulu, USA, pp. 120–127, 2007.
- [22] M. Henze, C. P. Grady, W. Gujor. Activated Sludge Model No. 1. London, England: Int. Association on Water Pollution Research and Control, Scientific and Technical Report No. 1, 1986.
- [23] H. W. Wang, Z. B. Wu, J. Meng. *Partial Least-squares Regression: Linear and Nonlinear Methods*, Beijing, China: National Defence Industry Press, 2006.



Hong-Tao Ye received the Ph.D. degree in control theory and control engineering from South China University of Technology, China in 2011. He is currently a professor in Guangxi University of Science and Technology, China. He has authored more than 20 referred journal and conference papers. He also serves as a reviewer for several international journals. He is a council member of Guangxi Association of Automation.

His research interests include computational intelligence and intelligent control.

E-mail: yehongtao@126.com (Corresponding author)

ORCID iD: 0000-0002-9459-9274



Zhen-Qiang Li received the Ph.D. degree in information science and electrical engineering from Kyushu University, Japan in 2010. He is currently an associate professor in Guangxi University of Science and Technology, China.

His research interests include computational intelligence and optimization theory.

E-mail: lizhenqiang67@163.com