

A Novel Randomised Particle Swarm Optimizer

Weibo Liu, Zidong Wang, Nianyin Zeng, Yuan Yuan, Fuad E. Alsaadi and Xiaohui Liu

Abstract—The particle swarm optimization (PSO) algorithm is a popular evolutionary computation approach that has received an ever-increasing interest in the past decade owing to its wide application potential. Despite the many variants of the PSO algorithm with improved search ability by means of both the convergence rate and the population diversity, the local optima problem remains a major obstacle that hinders the global optima from being found. In this paper, a novel randomized particle swarm optimizer (RPSO) is proposed where the Gaussian white noise with adjustable intensity is utilized to randomly perturb the acceleration coefficients in order for the problem space to be explored more thoroughly. With this new strategy, the RPSO algorithm not only maintains the population diversity but also enhances the possibility of escaping the local optima trap. Experimental results demonstrate that the proposed RPSO algorithm outperforms some existing popular variants of PSO algorithms on a series of widely used optimization benchmark functions.

Index Terms—Randomized algorithms, evolutionary computation, particle swarm optimization, Gaussian white noise, acceleration coefficients.

I. INTRODUCTION

Owing to their practical application insights, the optimization problems have drawn considerable research attention from both industrial and academic societies. The past few years have witnessed a rapid development of optimization techniques developed by various research communities including computer science, mathematics, control engineering and signal processing [2], [3], [21], [22]. In particular, as a powerful group of optimization techniques, the evolutionary computation (EC) approaches have proven to be highly efficient in solving global optimization problems with great application potentials, and have therefore attracted tremendous

This work was supported in part by the European Union's Horizon 2020 Research and Innovation Programme under Grant 820776 (INTEGRADDE), the UK-China Industry Academia Partnership Programme under Grant UK-CIAPP-276, the National Natural Science Foundation of China under Grants 61873148 and 61933007, the Royal Society of the UK, and the Alexander von Humboldt Foundation of Germany. (*Corresponding author: Zidong Wang*)

W. Liu and X. Liu are with the Department of Computer Science, Brunel University London, Uxbridge, Middlesex, UB8 3PH, United Kingdom.

Z. Wang is with the College of Electrical Engineering and Automation, Shandong University of Science and Technology, Qingdao 266590, China. He is also with the Department of Computer Science, Brunel University London, Uxbridge, Middlesex, UB8 3PH, United Kingdom. (email: Zidong.Wang@brunel.ac.uk)

N. Zeng is with the Department of Instrumental and Electrical Engineering, Xiamen University, Fujian 361005, China.

Y. Yuan is with the School of Astronautics, Northwestern Polytechnical University, Xi'an 710072, China.

Fuad E. Alsaadi is with the Department of Electrical and Computer Engineering, Faculty of Engineering, King Abdulaziz University, Jeddah 21589, Saudi Arabia.

research interest. Motivated by the biological evolution, many well-known EC approaches (e.g. particle swarm optimization (PSO) algorithm, evolution programming strategy, and genetic algorithm) have been successfully employed to a variety of real-world applications in the research areas of artificial intelligence, signal processing and system science [9], [11], [13], [31], [35], [36], [42]–[45]. Compared with other popular EC algorithms (e.g. the genetic algorithm, differential evolution, and simulate annealing), the PSO algorithm proposed in [17] exhibits competitive or even superior performance and is thus recognized as an excellent candidate algorithm due mainly to its technical merits of easy implementation and fast convergence towards the optimal solution.

Motivated by the mimics of the social interactions (e.g. fish schooling or birds flocking), the PSO algorithm aims to explore the search space by adjusting the velocity and position of particles according to the swarm intelligence. In fact, the PSO algorithm is capable of discovering the optimal solution both effectively and efficiently, and has been regarded as a rather powerful optimization technique. So far, the PSO algorithm has been successfully applied to solve the optimization problems in a wide range of real-world systems such as power systems [10], genetic regulatory networks [35], medical systems [42], [43] and path planning systems [32]. In the execution of a PSO algorithm, by cooperating and competing with other individual particles, each particle is encouraged to learn from its own experience and other competitors' experience to seek the globally optimal solution through the entire search space. During the evolution process at each iteration, each individual particle is guided by its historical personal best position and the global best position discovered by the entire yet dynamical swarm.

Population-based EC approaches are known to have the problems of easily getting trapped in the local optima especially when being used to solve large-scale optimization problems, and the PSO algorithm is not an exception. As such, despite the ongoing effort, it is still vitally important to further develop advanced algorithms in order to enhance global search capability of the PSO algorithms. Up to now, a great deal of research attention has been paid to the improvement of the search capability of the existing PSO algorithms by developing advanced PSO variants so as to alleviate the phenomenon of premature convergence. For example, a comprehensive learning particle swarm optimizer [19] has been developed to preserve the population diversity of the particles to avoid trapping in the local optima for complex multimodal problems. Recently, an adaptive PSO (APSO) algorithm has been introduced in [44] where an adaptive parameter selection scheme

has been proposed to automatically adjust the acceleration coefficients and the inertia weight, by which the search behavior of the PSO algorithm has been improved. More recently, a switching PSO (SPSO) algorithm has been put forward in [35] with the purpose of further improving the convergence rate of the APSO algorithm. Furthermore, a switching delayed PSO (SDPSO) algorithm has been proposed in [42] where time delays are introduced in the velocity updating model in order to make better use of historical information of the evolution process, thereby outperforming the SPSO algorithm with a better exploration of the problem space.

In a PSO algorithm, the acceleration coefficients are crucial parameters in achieving the balance between the global exploration and local searches through the entire problem space [13], [31], [34]. The selection of the acceleration coefficients plays an important role in successfully seeking the globally optimal solution. To enhance the possibility of escaping from the local optima, a great number of PSO variants have been developed with focuses on adjusting the acceleration coefficients, for example, controlling the acceleration coefficients in a time-varying manner [28]. Intuitively, a properly designed random perturbation (with adequate intensity) on the velocity updating model could lead to allowable variation of the acceleration coefficient that will not affect the convergence of the PSO algorithm but, rather, enhance the population diversity at each iteration, thereby further increasing the possibility of getting rid of local optima. In choosing a candidate for random perturbations, the well-known Gaussian white noise (GWN) appears to be an ideal candidate due to its constant power spectral density and easily tunable intensity at different frequencies when it comes to the implementation [37], [38].

Inspired by above argument of developing randomized algorithms [25], a seemingly natural idea is to introduce the GWNs into the acceleration coefficients of the PSO algorithm with hope to improve the population diversity and alleviate the premature convergence. The advantages of utilizing the GWNs are concluded as twofold: 1) the GWNs in the acceleration coefficients can alter the system dynamics (by means of iterations) which could contribute to a more thorough exploration and exploitation through the problem space; and 2) with the GWNs in place, the particles are entitled to exhibit more complicated dynamical behaviors (than the conventional PSO algorithms) which would enhance the capability of the particles escaping from the local optima and also improve the population diversity of the PSO algorithm.

Note that the randomised PSO algorithms have gained some research attention in the past few years. To be specific, the random perturbation on the *velocity updating model* has been considered in some variant PSO algorithms in order to reduce the possibility of getting trapped in local optima [18], [29], [39]. For example, in [39], a random perturbation has been added to slightly randomize the personal best position at each iteration. In [18], [29], a uniformly distributed noise term has been entered into the velocity updating model so as to avoid the stagnation problem. Rather than modifying the

velocity updating model, in this paper, we aim to design a new acceleration coefficient updating strategy by directly imposing the GWNs onto the acceleration coefficients that play a vitally important role in ensuring a proper balance between the global exploration and local exploitation with satisfactory convergence rate. Comparing with the algorithms developed in [18], [29], [39], our proposed RPSO algorithm exhibits the advantage of achieving multiple objectives (e.g. global exploration, local exploitation and convergence) through randomly yet purposely perturbing the acceleration coefficients.

To conclude the discussions made so far, in this paper, we endeavor to propose a randomized PSO (RPSO) algorithm where the GWN with adequately adjusted intensity is utilized to randomly perturb the acceleration coefficients in order for the problem space to be explored more thoroughly. The main contributions can be summarized as follows: 1) a novel RPSO algorithm is developed where the GWNs are embedded in the velocity updating model to adjust the acceleration coefficients at each iteration, which helps prevent the undesirable premature convergence; and 2) the proposed RPSO algorithm is comprehensively verified on a series of test functions (including both the unimodal and multimodal cases) and it is demonstrated that the RPSO algorithm outperforms some existing popular variants of PSO algorithms on a series of widely used optimization benchmark functions.

The rest of this paper is organized as follows. The basic PSO algorithm and its recent developments are introduced in Section II. In Section III, the proposed RPSO algorithm is explained in detail. Experimental results, parameter setting and discussions are presented in Section IV. Finally, conclusions and future directions are drawn in Section V.

II. DEVELOPMENT OF PSO ALGORITHMS

In the canonical PSO algorithm proposed in [17], the particles are learning from its own experience and cooperating with other members to seek the optimal solution of the optimization problem where each particle represents a candidate solution. All the particles are exploring and exploiting in the D -dimensional search space with a certain velocity. At the k th iteration, the position and velocity of the i th particle are represented by $X_i(k) = (X_{i1}(k), X_{i2}(k), \dots, X_{iD}(k))$ and $V_i(k) = (V_{i1}(k), V_{i2}(k), \dots, V_{iD}(k))$, respectively. Inspired by the swarm intelligence paradigm, the particles are motivated to move to their own personal best position (pbest) as well as the global best position (gbest) found by the entire swarm. Theoretically, the velocity and position of the i th particle are described as follows

$$\begin{aligned} V_i(k+1) &= wV_i(k) + c_1r_1(\text{pbest}_i(k) - X_i(k)) \\ &\quad + c_2r_2(\text{gbest}(k) - X_i(k)) \\ X_i(k+1) &= X_i(k) + V_i(k+1) \end{aligned} \quad (1)$$

where w represents the inertia weight; k denotes the iteration number; c_1 and c_2 are the cognitive acceleration coefficient and social acceleration coefficient which pull the particle towards its pbest and the gbest, respectively. pbest_i represents

the pbest of particle i ; g_{best} indicates the gbest discovered by the entire swarm; and r_1 and r_2 are two uniformly distributed random variables taking values on $[0, 1]$.

As mentioned previously, the individual particle in executing a PSO algorithm may easily get trapped in the local optima, and this leads to the so-called premature convergence. Under this circumstance, it is of critical significance to develop advanced approaches to improve the search ability of the PSO algorithms. Up to now, a variety of PSO variants have been introduced with the purpose of improving the search ability of PSO algorithm, reducing the possibility of trapping into local optima and alleviating premature convergence, see in [4], [7], [30]–[32], [35], [42], [44]. To be specific, three types of PSO variants have been introduced by: 1) putting forward novel strategies to adjust the control parameters; 2) designing new updating topological structures and 3) hybridizing with other EC algorithms.

A large number of famous PSO variants have been developed according to the adaption of the control parameters including the inertia weight and acceleration coefficients. This type of PSO variants aims to balance the global exploration and local exploitation during the search process by adjusting the control parameters. For example, a time-varying strategy has been designed in the PSO algorithms to alter the control parameters. For some representative algorithms in this regard, we refer the readers to PSO algorithm with a linearly decreased inertia weight (PSO-LDIW) [30], [31] and the PSO algorithm with time-varying acceleration coefficients (PSO-TVAC) [28]. Moreover, an advanced variant PSO algorithm with constriction factor (PSO-CK) has been developed in [12] to further improve the convergence performance of the PSO algorithm.

In addition to modifying the control parameters, some popular PSO variants with new topological structures have been developed during the past few decades. Notably, the introduced velocity updating model of the PSO algorithms could enhance the swarm diversity and alleviate premature convergence. The APSO algorithm proposed in [44] aims to adapt the control parameters based on the evolutionary state determined by the evolutionary factor. Recently, time-delay terms have been entered into the velocity updating model which not only alters the system dynamics but also makes better use of historical information through the evolution process, and thus improves the convergence behaviors of PSO algorithms and reduces the trapping possibility into the local optima, see [32], [33], [42]. Additionally, the PSO algorithm with a dynamical diversity coefficient has been proposed in [15] where a random velocity controlled by a diversity coefficient has been taken into consideration to further improve the PSO algorithm by enhancing the search ability. With the purpose to improve the diversity and avoiding the premature convergence problem, an augmented PSO algorithm in combination with multiple adaptive methods has been put forward in [16] where an intelligent selection mechanism has been developed to select an appropriate search approach. Very recently, a novel N-state Markovian jumping PSO algorithm has been developed where

the velocity updating equation has been adjusted based on the state evolution governed by a Markov chain [27]. It is worth pointing out that the impact of different communication topologies in the PSO algorithm has been investigated in [1].

On the other hand, the hybridization of the standard PSO algorithm and other EC algorithms is another well-studied research forefront. The genetic algorithm has been adopted to further enhance the search ability of the PSO algorithm in [14]. Recently, a switching local evolutionary PSO algorithm has been developed in [41] which utilizes the DE algorithm to further improve the search capability of the PSO algorithm and alleviate the premature convergence. Very recently, a hybrid PSO algorithm with simulated annealing has been proposed in [26], which could increase the population diversity of the swarm and improve the possibility of jumping out of the local optimum. Nevertheless, the above PSO algorithms may still get trapped in the local optimal solutions, and some of them suffer from high computational cost especially for complex and high-dimensional multimodal problems. Therefore, it is both theoretically important and practically significant to develop novel approaches that further enhance the search performance of the optimizer with regard to both diversity and convergence rate.

III. A NEW RPSO ALGORITHM

A novel RPSO algorithm is developed in this section where the GWNs are entered into the updating model for velocity for randomly perturbing (with adequate intensity) the acceleration coefficients with hope to decrease the trapping possibility into the local optima and also seek the optimal solution more thoroughly. The motivation and framework of the proposed RPSO algorithm are illustrated in details.

A. Motivation

To control the PSO algorithm in an effective way, the control parameters (such as the inertia weight, the social acceleration coefficient and the cognitive acceleration coefficient) are dominantly crucial during the evolution process. The inertia weight is a significant parameter in controlling the exploration of the search space which is commonly set to be a constant or a dynamically changing value [31]. Similarly, the acceleration coefficients (composed of the cognitive component and the social component) are used to control the movement of the particles towards their personal best position and the global best position discovered by the entire swarm, respectively. In general, the parameter setting of the acceleration coefficients plays an adequate role in achieving the balance between the local search and the global exploration through the optimization process. As such, an appropriate selection of the acceleration coefficients is of vital importance to seek the global optimum effectively and accurately. Unfortunately, some existing PSO algorithms which focus on adjusting the control parameters (such as the PSO-TVAC algorithm) may still easily get trapped in the local optima. Therefore, it is of crucial importance to

investigate an advanced parameter selection mechanism so as to reduce the possibility of trapping into local optima and further enhance the search capability of the PSO algorithms.

We are now ready to introduce our novel RPSO algorithm dedicatedly designed to enhance the search ability of the particles with the hope to thoroughly explore and exploit the entire problem space. The major novelty of the newly proposed RPSO model is to separately introduce the GWNs into the cognitive acceleration coefficient as well as the social acceleration coefficient to effectively and efficiently seek the optimal solution. By establishing such a new velocity updating model, the RPSO algorithm consists of the following two advantages: 1) the GWNs are separately added to the social and cognitive acceleration coefficients which randomly perturb the movement of the particles at each iteration; and 2) the dynamical behavior of the RPSO algorithm becomes more complicated than the basic PSO algorithm and the particles are therefore allowed to expand their search space, which leads to a more thorough exploration of the problem space with less trapping possibility into the local optima.

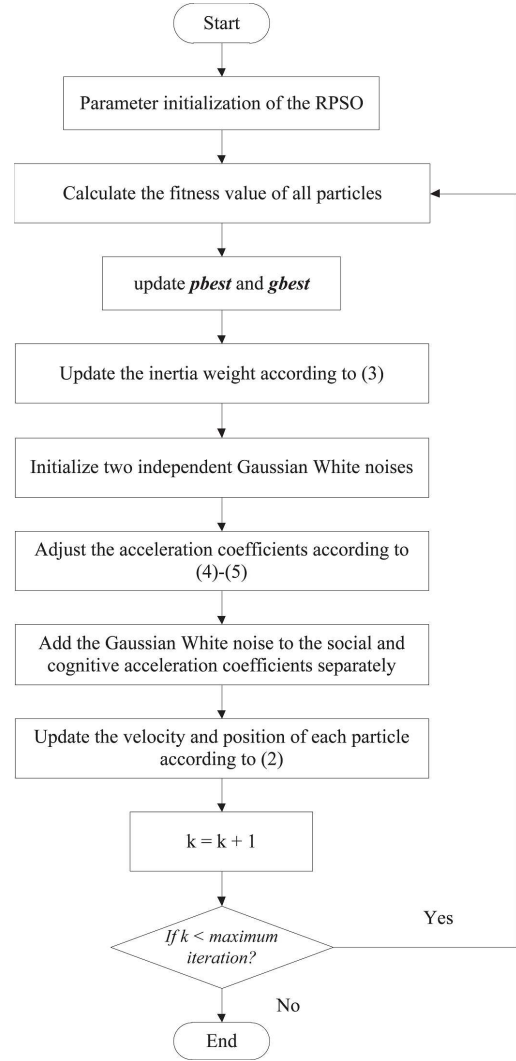


Fig. 1. Flowchart of the RPSO algorithm

The velocity and position of the i th particle are updated based on the following equations:

$$\begin{aligned}
 V_i(k+1) &= wV_i(k) + r_1(C_p + \delta_1(k))(pbest_i(k) - X_i(k)) \\
 &\quad + r_2(C_g + \delta_2(k))(gbest(k) - X_i(k)) \\
 X_i(k+1) &= X_i(k) + V_i(k+1)
 \end{aligned} \tag{2}$$

where k denotes the iteration number; C_p and C_g indicate the acceleration coefficients defined in Eq. (4) and Eq. (5), respectively; w is inertia weight represented by Eq. (3); $\delta_1(k)$ and $\delta_2(k)$ represent two independent GWNs; and r_1 and r_2 are two uniformly distributed random numbers on $[0, 1]$. Notably, the GWNs ($\delta_1(k)$ as well as $\delta_2(k)$) and the random numbers (r_1 and r_2) are mutually independent.

Motivated by the PSO-LDIW [31] and the PSO-TVAC [28] algorithms, the inertia weight w and the acceleration coefficients C_p and C_g of the RPSO algorithm are shown as follows:

$$w = w_{\max} - (w_{\max} - w_{\min}) \times \frac{k}{\text{maxit}} \tag{3}$$

$$C_p = (C_{p \max} - C_{p \min}) \times \frac{\text{maxit} - k}{\text{maxit}} + C_{p \min} \tag{4}$$

B. Framework of the RPSO Algorithm

For the novel RPSO algorithm, the flowchart is depicted in Fig. 1.

$$C_g = (C_{g \min} - C_{g \max}) \times \frac{\text{maxit} - k}{\text{maxit}} + C_{g \max} \quad (5)$$

where w_{\max} , $C_{p \max}$ and $C_{g \max}$ represent the maximum value of the inertia weight w , acceleration coefficients C_p and C_g , respectively; w_{\min} , $C_{p \min}$ and $C_{g \min}$ denote the minimum value of w , C_p and C_g , respectively; and maxit indicates the maximum iteration.

The pseudo-code of the proposed RPSO algorithm is shown in Algorithm 1.

Algorithm 1 RPSO Algorithm

Require: Initialize the swarm size P , the velocity and position of the particles V_i , X_i , the acceleration coefficients C_p , C_g , the inertia weight w , the maximum iteration, the Gaussian white noises δ_1 and δ_2 .

1. Calculate the fitness of each particle, and update the $pbest$ and $gbest$.
 2. Update the inertia weight according to Eq. (3).
 3. Adjust the acceleration coefficients based on Eqs. (4)-(5).
 4. Update the velocity and position equations according to Eq. (2).
 5. Repeat Steps 1 to 4 till the algorithm reaches the maximum iteration number.
-

IV. PERFORMANCE EVALUATION AND DISCUSSION OF THE RPSO ALGORITHM

In this section, the performance of the proposed RPSO algorithm is evaluated and discussed by comparing with some widely-used PSO algorithms. In the simulation, some popular test functions including both of the unimodal and multimodal cases are taken into account to provide a comprehensive review of the optimization performance of the developed RPSO algorithm. In this paper, all the experiments are implemented by using MATLAB 2017b on a PC with the Intel Core i5 – 4590 3.30 GHz CPU and the Microsoft Windows 7 Enterprise 64-bit operating system.

In our simulation, the swarm size is set as $S = 30$ and the dimension of the problem space is $D = 30$. Furthermore, the maximum number of iteration is $k = 10000$ for all the tested PSO algorithms. To strengthen the reliability of the simulation results, each experiment is repeated independently for 50 times. The parameters of the compared PSO algorithms are set up according to the literature [8], [17], [28], [30], [35], [42]. For the RPSO algorithm, the inertia weight w is linearly decreased from 0.9 to 0.4. The acceleration coefficients C_p and C_g are set in the range of [0.5, 2.5] where $C_{p \max} = C_{g \max} = 2.5$ and $C_{p \min} = C_{g \min} = 0.5$. The mean value and the variance of the GWNs $\delta_1(k)$ and $\delta_2(k)$ are set to be 0 and 0.07, respectively.

A. Test Functions

In this paper, eight well-known test functions are selected for evaluating the search ability of the proposed RPSO algorithm by comparing with six popular PSO algorithms. The

selected PSO algorithms include the standard PSO algorithm [17], the PSO-LDIW algorithm [30], the PSO-TVAC algorithm [28], the PSO-CK algorithm [8], the SPSO algorithm [35], and the SDPSO algorithm [42]. Among the selected test functions, $f_1(x)$ (the Sphere function) is a typical unimodal function which is often utilized to justify the convergence rate of the EC approaches; $f_2(x)$ (the Rosenbrock function), referred to as the Rosenbrock's banana function, is a widely-used test problem for optimization algorithms; $f_3(x)$ (the Rastrigin function) and $f_5(x)$ (the Griewank function) have a large number of local optima, which are hard to discover the globally optimal solution; and other selected test functions are also popular benchmark functions. It should be pointed out that all the test functions are minimization problems and all of them have a global minimum. Let $x = (x_1, x_2, \dots, x_D)$ where $D = 30$ is the dimension of the problem space. The mathematical formulations of the test functions are given as follows.

$$\text{Sphere} : f_1(x) = \sum_{i=1}^D x_i^2. \quad (6)$$

$$\text{Rosenbrock} : f_2(x) = \sum_{i=1}^{D-1} (100(x_{i+1} - x_i)^2 + (x_i - 1)^2). \quad (7)$$

$$\text{Rastrigin} : f_3(x) = \sum_{i=1}^D (x_i^2 - 10 \cos 2\pi x_i + 10). \quad (8)$$

$$\text{Schwefel 1.2} : f_4(x) = \sum_{i=1}^D \left(\sum_{j=1}^i x_j \right)^2. \quad (9)$$

$$\text{Griewank} : f_5(x) = 1 + \frac{1}{4000} \sum_{i=1}^D x_i^2 - \prod_{i=1}^D \cos\left(\frac{x_i}{\sqrt{i}}\right). \quad (10)$$

$$\begin{aligned} \text{Penalized 1} : f_6(x) &= \frac{\pi}{D} \left(10 \sin^2(\pi y_1) \right. \\ &+ \sum_{i=1}^{D-1} (y_i - 1)^2 (1 + 10 \sin^2(\pi y_{i+1})) \\ &\left. + (y_D - 1)^2 \right) + \sum_{i=1}^D u(x_i). \end{aligned} \quad (11)$$

$$y_i = 1 + 1/4(x_i + 1),$$

$$u(x_i) = \begin{cases} 100(-x_i - 10)^4, & x_i < -10, \\ 0, & |x_i| \leq 10, \\ 100(x_i - 10)^4, & x_i > 10. \end{cases}$$

$$\text{Step} : f_7(x) = \sum_{i=1}^D (|x_i + 0.5|)^2. \quad (12)$$

$$\begin{aligned} \text{Penalized 2} : f_8(x) &= 0.1 \left(\sin^2(3\pi x_1) \right. \\ &+ \sum_{i=1}^{D-1} (x_i - 1)^2 (1 + \sin^2(3\pi x_{i+1})) \\ &\left. + (x_D - 1)^2 (1 + \sin^2(2\pi x_D)) \right) \end{aligned} \quad (13)$$

$$+ \sum_{i=1}^D u(x_i).$$

$$u(x_i) = \begin{cases} 100(-x_i - 5)^4, & x_i < -5, \\ 0, & |x_i| \leq 5, \\ 100(x_i - 5)^4, & x_i > 5. \end{cases}$$

The detailed information of the test functions is shown in Table I including the name of the test functions, the search range of each optimization problem, the maximum velocity of the particle for each test function, the threshold, and minimum of the test functions. Notably, the search range of the test function which indicates the range of the search space is determined by the literature [40].

To further verify the validity of the RPSO algorithm on some currently popular optimization functions, six functions selected from the CEC 2015 competition on learning-based real-parameter single objective optimization are employed [20]. Among the selected benchmark functions, $f_9(x)$ and $f_{10}(x)$ are unimodal functions. $f_{11}(x)$ is a simple multimodal function with a large number of local optima. $f_{12}(x)$ is a hybrid function. $f_{13}(x)$ and $f_{14}(x)$ are composition functions which are difficult to discover the globe optimum. The dimension of the selected CEC 2015 benchmark functions is set to be 30. The detailed information is displayed in Table II. Furthermore, the maximum velocity of each particle in PSO approaches is usually limited by a certain value with the hope to avoid searching outside the predefined search range. Due to empirical investigations on the test functions, the maximum velocity is often set up by 10 – 20% of the dynamic range of each dimension for different test functions [13], [28]. In our simulation, the maximum velocity is set to be 20% of the dynamical range.

B. Experimental Studies of the RPSO Algorithm

To evaluate the solution quality of the proposed RPSO algorithm, three popular performance indicators (including the convergence rate, success ratio and population diversity) are utilized. Note that the success ratio is an important criterion to measure the accuracy of the PSO approaches. The population diversity is used to evaluate the solution quality. In this paper, the convergence plots of the PSO algorithms are depicted from Figs. 2-9 to demonstrate the convergence rate of the PSO algorithms where the vertical coordinate indicates the logarithm value of the average fitness value and the horizontal coordinate denotes the number of iteration. Furthermore, the minimum, mean and standard deviation fitness values of the PSO algorithms are employed to demonstrate the solution quality of the adopted PSO approaches in Table III.

From the figures, it is clear that the proposed RPSO algorithm exhibits competitive performance than other selected PSO variants. In Figs. 3-5 and Figs. 7-8, the RPSO algorithm obtains the best average fitness value among the selected PSO algorithms, and the convergence rate is satisfactory. Additionally, the RPSO algorithm discovers the globally optimal solution of function (12). Although the convergence

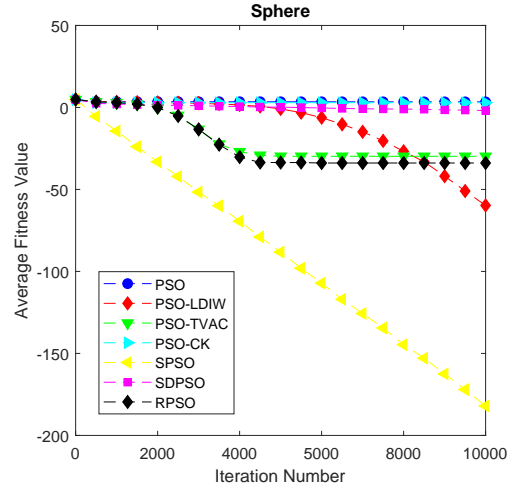


Fig. 2. Algorithm Convergence Characteristics for Sphere function $f_1(x)$

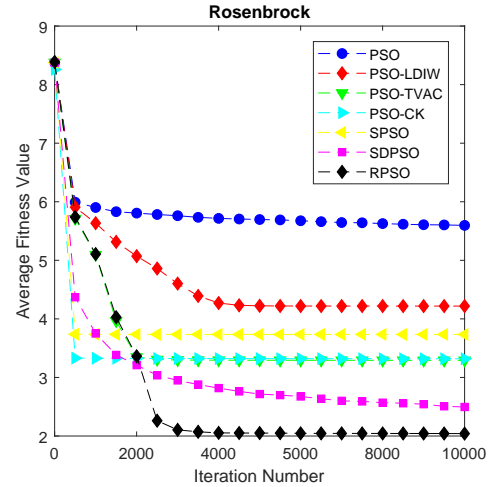


Fig. 3. Algorithm Convergence Characteristics for Ackley function $f_2(x)$

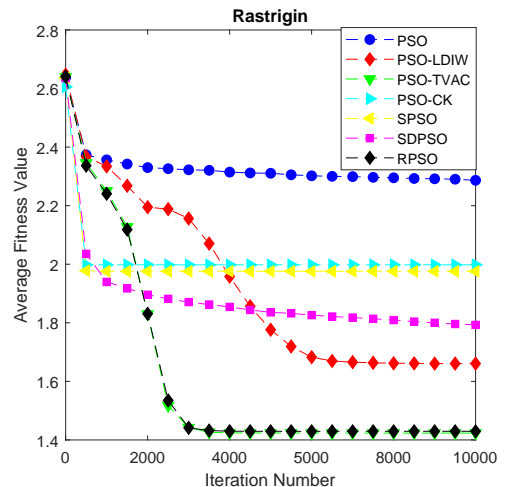


Fig. 4. Algorithm Convergence Characteristics for Rastrigin function $f_3(x)$

TABLE I
TEST FUNCTION CONFIGURATION

Test Functions	Dimension	Search Range	Maximum Velocity	Threshold	Minimum
$f_1(x)$: Sphere	30	$[-100, 100]$	40	0.01	0
$f_2(x)$: Rosenbrock	30	$[-30, 30]$	12	100	0
$f_3(x)$: Rastrigin	30	$[-5.12, 5.12]$	2.048	50	0
$f_4(x)$: Schwefel 1.2	30	$[-100, 100]$	40	0.01	0
$f_5(x)$: Griewank	30	$[-600, 600]$	240	0.01	0
$f_6(x)$: Penalized 1	30	$[-50, 50]$	20	0.01	0
$f_7(x)$: Step	30	$[-100, 100]$	40	0.01	0
$f_8(x)$: Penalized 2	30	$[-50, 50]$	20	0.01	0

TABLE II
SELECTED CEC 2015 BENCHMARK FUNCTIONS

Test Functions	Dimension	Search Range	Minimum
$f_9(x)$: Rotated High Conditioned Elliptic Function	30	$[-100, 100]$	0
$f_{10}(x)$: Rotated Cigar Function	30	$[-100, 100]$	0
$f_{11}(x)$: Shifted and Rotated Schwefels Function	30	$[-100, 100]$	0
$f_{12}(x)$: Hybrid Function 3	30	$[-100, 100]$	0
$f_{13}(x)$: Composition Function 2	30	$[-100, 100]$	0
$f_{14}(x)$: Composition Function 6	30	$[-100, 100]$	0

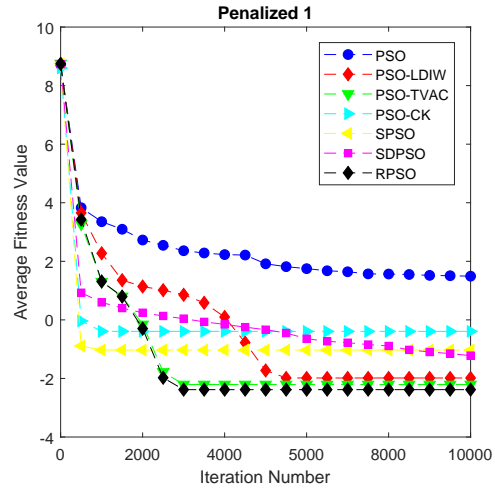
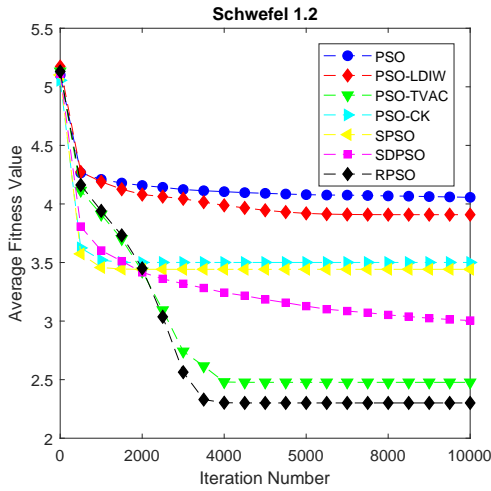


Fig. 5. Algorithm Convergence Characteristics for Schwefel 2.22 function $f_4(x)$

Fig. 7. Algorithm Convergence Characteristics for Penalized 1 function $f_6(x)$

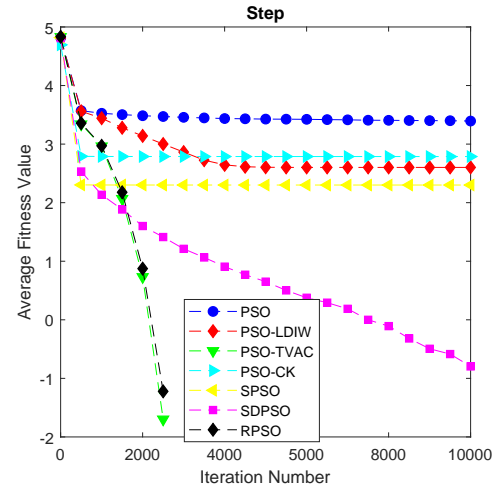
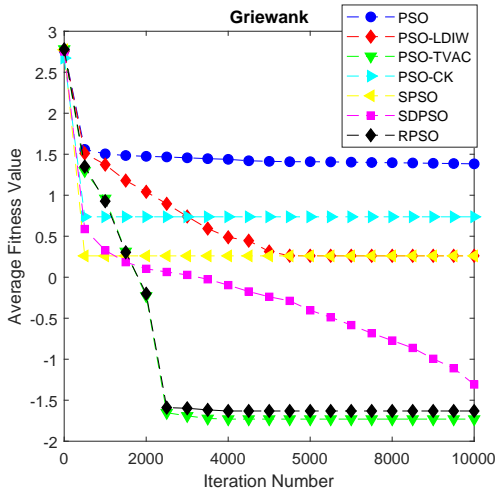


Fig. 6. Algorithm Convergence Characteristics for Griewank function $f_5(x)$

Fig. 8. Algorithm Convergence Characteristics for Step function $f_7(x)$

TABLE III
STATISTICAL RESULTS OF PSO ALGORITHMS ON WIDELY-USED BENCHMARK FUNCTIONS

		PSO	PSO-LDIW	PSO-TVAC	PSO-CK	SPSO	SDPSO	RPSO
$f_1(x)$	Minimum	1.44×10^3	2.38×10^{-64}	3.97×10^{-57}	6.07×10^{-175}	3.23×10^{-196}	3.42×10^{-5}	1.78×10^{-55}
	Mean	2.55×10^3	1.81×10^{-60}	1.62×10^{-30}	8.00×10^2	8.58×10^{-183}	3.59×10^{-67}	1.30×10^{-34}
	Std. Dev.	1.45×10^3	8.82×10^{-60}	9.45×10^{-30}	2.74×10^3	0.0000	2.88×10^{-2}	5.61×10^{-34}
	Ratio	0%	100%	100%	92%	100%	74%	100%
$f_2(x)$	Minimum	2.28×10^5	5.75×10^{-3}	2.31×10^{-3}	7.67×10^{-6}	5.33×10^{-4}	2.77×10^1	3.82×10^{-4}
	Mean	3.95×10^5	1.65×10^4	1.96×10^3	2.09×10^3	5.43×10^3	3.13×10^2	1.10×10^2
	Std. Dev.	1.12×10^5	3.48×10^4	1.27×10^4	1.27×10^4	2.16×10^4	6.95×10^2	4.32×10^2
	Ratio	0%	64%	90%	80%	92%	54%	94%
$f_3(x)$	Minimum	1.34×10^2	1.29×10^1	1.39×10^1	4.97×10^1	3.18×10^1	2.69×10^1	1.29×10^1
	Mean	1.93×10^2	4.58×10^1	2.65×10^1	9.96×10^1	9.46×10^1	6.21×10^1	2.69×10^1
	Std. Dev.	2.22×10^1	2.18×10^1	6.5499	2.96×10^1	3.16×10^1	2.15×10^1	1.00×10^1
	Ratio	0%	60%	100%	2%	8%	36%	96%
$f_4(x)$	Minimum	4.05×10^3	9.74×10^{-4}	1.71×10^{-8}	2.87×10^{-24}	5.79×10^{-14}	8.60×10^1	1.13×10^{-9}
	Mean	1.14×10^4	8.10×10^3	3.00×10^2	3.17×10^3	2.77×10^3	1.01×10^3	2.00×10^2
	Std. Dev.	5.67×10^3	7.93×10^3	1.20×10^3	3.94×10^3	4.47×10^3	1.81×10^3	9.90×10^2
	Ratio	0%	10%	92%	54%	62%	0%	92%
$f_5(x)$	Minimum	1.45×10^1	0.0000	0.0000	0.0000	0.0000	1.83×10^{-5}	0.0000
	Mean	2.42×10^1	1.8255	1.86×10^{-2}	5.4554	1.8216	4.92×10^{-2}	2.34×10^{-2}
	Std. Dev.	1.32×10^1	1.28×10^1	2.16×10^{-2}	2.17×10^1	1.28×10^1	7.86×10^{-2}	2.25×10^{-2}
	Ratio	0%	36%	54%	30%	56%	18%	38%
$f_6(x)$	Minimum	1.79×10^1	1.57×10^{-32}	1.57×10^{-32}	1.57×10^{-32}	1.57×10^{-32}	5.07×10^{-5}	1.57×10^{-32}
	Mean	3.11×10^1	1.04×10^{-2}	6.22×10^{-3}	4.05×10^{-1}	9.33×10^{-2}	6.01×10^{-2}	4.15×10^{-3}
	Std. Dev.	1.12×10^1	3.14×10^{-2}	2.49×10^{-2}	6.66×10^{-1}	1.44×10^{-1}	1.88×10^{-1}	2.05×10^{-2}
	Ratio	0%	90%	94%	40%	54%	72%	96%
$f_7(x)$	Minimum	1.48×10^3	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	Mean	2.47×10^3	4.00×10^2	0.0000	6.13×10^2	2.00×10^2	1.60×10^{-1}	0.0000
	Std. Dev.	1.41×10^3	1.98×10^3	0.0000	2.40×10^3	1.41×10^3	5.48×10^{-1}	0.0000
	Ratio	0%	96%	100%	24%	92%	90%	100%
$f_8(x)$	Minimum	1.49×10^4	1.35×10^{-32}	1.35×10^{-32}	1.35×10^{-32}	5.83×10^{-31}	2.11×10^{-5}	1.35×10^{-32}
	Mean	1.06×10^5	8.79×10^{-4}	1.76×10^{-3}	1.69×10^{-1}	1.13×10^9	2.58×10^{-2}	1.10×10^{-3}
	Std. Dev.	5.98×10^4	3.01×10^{-3}	4.07×10^{-3}	7.10×10^{-1}	3.00×10^8	3.78×10^{-2}	3.33×10^{-3}
	Ratio	0%	92%	84%	50%	2%	48%	90%

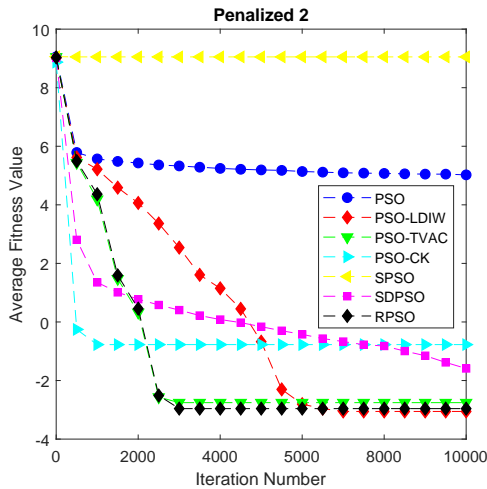


Fig. 9. Algorithm Convergence Characteristics for Penalized 2 function $f_8(x)$

characteristics of the RPSO algorithm is not the best among all the PSO variants for function (6), function (10) and function (13), the difference between the best minimum value is not significant. Overall, the convergence behavior of the RPSO algorithm outperforms other selected PSO algorithms.

The statistical results of the performance evaluation of the selected PSO approaches are shown in Table III. By analyzing

TABLE IV
STATISTICAL RESULTS OF PSO ALGORITHMS ON SELECTED CEC 2015 COMPETITION ON LEARNING-BASED REAL-PARAMETER SINGLE OBJECTIVE OPTIMIZATION PROBLEMS

		PSO	RPSO
$f_9(x)$	Minimum	5.60×10^7	9.57×10^4
	Mean	1.13×10^8	3.42×10^6
	Std. Dev.	3.21×10^7	3.12×10^6
$f_{10}(x)$	Minimum	5.27×10^9	2.01×10^2
	Mean	8.94×10^9	8.33×10^3
	Std. Dev.	1.52×10^9	1.05×10^6
$f_{11}(x)$	Minimum	6.58×10^3	1.60×10^3
	Mean	7.36×10^3	3.51×10^3
	Std. Dev.	3.05×10^2	1.07×10^3
$f_{12}(x)$	Minimum	4.31×10^5	2.05×10^4
	Mean	1.04×10^6	1.60×10^5
	Std. Dev.	3.71×10^5	1.04×10^5
$f_{13}(x)$	Minimum	3.47×10^5	7.09×10^3
	Mean	1.96×10^6	2.09×10^5
	Std. Dev.	8.67×10^5	1.24×10^5
$f_{14}(x)$	Minimum	3.28×10^4	3.10×10^4
	Mean	3.59×10^4	3.33×10^4
	Std. Dev.	2.37×10^3	2.17×10^3

the experimental results of the RPSO algorithm on the chosen 30-D optimization problems, the proposed RPSO algorithm outperforms six popular PSO algorithms in terms of both of the success ratio and the population diversity. Although it is

tough to find the optimal solution, the algorithm meets the requirement if the fitness value reaches the specific threshold of each test function. The RPSO algorithm obtains best success ratio in function (6), function (7), function (9), function (11) and function (12). For function (8), function (10) and function (13), the RPSO algorithm also exhibits competitive success ratio than other PSO algorithms. Notably, the success ratio of each PSO algorithms on function (10) is very low because the Griewank function consists of a large number of local minima, which is difficult to discover the globally optimal solution. Overall, we can draw the conclusion that the RPSO algorithm is capable of escaping from the local optima.

In Table III, the mean, minimum, and standard deviation of the fitness values are also presented. Note that the minimum fitness value represents the optimal solution found by the PSO algorithm. It should be mentioned that all the test functions utilized in this paper are minimization problems. In this regard, a smaller fitness value indicates a better solution explored by the PSO algorithm. The RPSO algorithm obtains the smallest minimum fitness value on function (8) and functions (10) to (13). In addition, the RPSO algorithm obtains satisfactory minimum fitness values on the rest of the test functions. It is worth mentioning that the RPSO algorithm also gets the smallest mean and standard deviation of the fitness value on function (7), function (9), and functions (11) to (13), which indicates that the solution quality of the RPSO algorithm is competitive than other compared PSO algorithms. Furthermore, the RPSO algorithm outperforms the standard PSO algorithm, the PSO-LDIW algorithm, the PSO-CK algorithm, the SPSO algorithm and the SDPSO algorithm on the mean, minimum, and standard deviation of the fitness values as well as the success ratio for function (8). For function (12), the success ratio of the RPSO algorithm is 100% which indicates that the globally optimal solution is discovered for all the repeats. In this case, we can draw the conclusion that the search ability of the RPSO algorithm is competitive than the compared PSO algorithms.

Experiment results on the selected CEC 2015 benchmark functions are shown in Table IV, where the mean, minimum, and standard deviation of the fitness values are adopted to verify the validity of the RPSO algorithm by comparing with the standard PSO algorithm. It can be seen in Table IV that the minimum fitness value of the RPSO algorithm is smaller than that of the standard PSO algorithm for $f_9(x)$ to $f_{14}(x)$, which indicates that the RPSO algorithm discovers a better solution than the standard PSO algorithm. Additionally, the RPSO algorithm outperforms the standard PSO algorithm on the mean fitness value for $f_9(x)$ to $f_{14}(x)$. To summarize, the proposed RPSO algorithm demonstrates superior performance over the compared popular PSO algorithms in terms of the strong ability to escape from the local optima, the satisfactory convergence performance and population diversity.

V. CONCLUSION

In this paper, a RPSO algorithm has been proposed to improve the search ability of the basic PSO approach. The GWNs have been added to the social acceleration coefficient and the cognitive acceleration coefficient separately to randomly change the acceleration coefficients. Experimental results have shown that the introduced RPSO algorithm outperforms six existing PSO algorithms on eight popular test functions. In the near future, we will further investigate advanced approaches to improving the population diversity of the RPSO algorithm for large-scale and complex optimization problems. Additionally, we aim to apply the proposed RPSO algorithm to other research topics, e.g., the community detection, the link prediction, the robust control of complex networks, system science, and telecommunication [5], [6], [23], [24], [37], [38], [46]–[48].

REFERENCES

- [1] T. Blackwell and J. Kennedy, "Impact of communication topology in particle swarm optimization," *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 4, pp. 689–702, Nov. 2018.
- [2] J. Cao, Z. Bu, G. Gao and H. Tao, Weighted modularity optimization for crisp and fuzzy community detection in large-scale networks, *Physica A: Statistical Mechanics and its Applications*, vol. 462, pp. 386–395, Nov. 2016.
- [3] J. Cao, Z. Bu, Y. Wang, H. Yang, J. Jiang and H.-J. Li, Detecting prosumer-community group in smart grids from the multiagent perspective, *IEEE Transactions on Systems Man Cybernetics-Systems*, vol. 49, no. 8, pp. 1652–1664, Aug. 2019.
- [4] W.-N. Chen, J. Zhang, Y. Lin, N. Chen, Z.-H. Zhan, H. S.-H. Chung, Y. Li and Y.-H. Shi, "Particle swarm optimization with an aging leader and challengers," *IEEE Transactions on Evolutionary Computation*, vol. 17, no. 2, pp. 241–258, Apr. 2013.
- [5] Y. Chen, Z. Wang, L. Wang and W. Sheng, Mixed H_2/H_∞ state estimation for discrete-time switched complex networks with random coupling strengths through redundant channels, *IEEE Transactions on Neural Networks and Learning Systems*, in press, DOI: 10.1109/TNNLS.2019.2952249.
- [6] Y. Chen, Z. Wang, L. Wang and W. Sheng, Finite-horizon H_∞ state estimation for stochastic coupled networks with random inner couplings using Round-Robin protocol, *IEEE Transactions on Cybernetics*, in press, DOI: 10.1109/TCYB.2020.3004288.
- [7] R. Cheng and Y. Jin, "A competitive swarm optimizer for large scale optimization," *IEEE Transactions on Cybernetics*, vol. 45, no. 2, pp. 191–204, Feb. 2015.
- [8] M. Clerc and J. Kennedy, "The particle swarm: explosion, stability, and convergence in a multi-dimensional complex space," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 1, pp. 58–73, Feb. 2002.
- [9] L. Cui, G. Li, Q. Lin, J. Chen, and N. Lu, "Adaptive differential evolution algorithm with novel mutation strategies in multiple sub-populations," *Computers & Operations Research*, vol. 67, pp. 155–173, Mar. 2016.
- [10] Y. Del Valle, G. K. Venayagamoorthy, S. Mohagheghi, J.-C. Hernandez, and R. G. Harley, "Particle swarm optimization: basic concepts, variants and applications in power systems," *IEEE Transactions on Evolutionary Computation*, vol. 12, no. 2, pp. 171–195, Apr. 2008.
- [11] R. C. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," In: *Proceedings of the 6th International Symposium on Micro Machine and Human Science*, Nagoya, Japan, Oct. 1995, pp. 39–43.
- [12] R. C. Eberhart and Y.-H. Shi, "Comparing inertia weights and constriction factors in particle swarm optimization," In: *Proceedings of the 2000 Congress on Evolutionary Computation*, San Diego, USA, Jul. 2000, pp. 84–88.
- [13] R. C. Eberhart and Y.-H. Shi, "Particle swarm optimization: developments, applications and resources," In: *Proceedings of the 2001 Congress on Evolutionary Computation*, Seoul, South Korea, May 2001, vol. 1, pp. 81–86.

- [14] H. Garg, "A hybrid PSO-GA algorithm for constrained optimization problems," *Applied Mathematics and Computation*, vol. 274, pp. 292-305, Feb. 2016.
- [15] A. Garcia-Villoria and R. Pastor, "Introducing dynamic diversity into a discrete particle swarm optimization," *Computers & Operations Research*, vol. 36, no. 3, pp. 951-966, Mar. 2009.
- [16] M. Hu, T. Wu, and J. D. Weir, "An adaptive particle swarm optimization with multiple adaptive methods," *IEEE Transactions on Evolutionary Computation*, vol. 17, no. 5, pp. 705-720, Oct. 2013.
- [17] J. Kennedy, and R. C. Eberhart, "Particle swarm optimization," In: *Proceedings of the 1995 IEEE International Conference on Neural Networks*, Perth, Australia, Nov. 1995, pp. 1942-1948.
- [18] P. K. Lehre and C. Witt, "Finite first hitting time versus stochastic convergence in particle swarm optimisation," May 2011, arXiv:1105.5540.
- [19] J. J. Liang, A. K. Qin, P. N. Suganthan, and S. Baskar, "Comprehensive learning particle swarm optimizer for global optimization of multimodal functions," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 3, pp. 281-295, Jun. 2006.
- [20] J. J. Liang, B. Y. Qu, P. N. Suganthan, and Q. Chen, "Problem definitions and evaluation criteria for the CEC 2015 competition on learning-based real-parameter single objective optimization," *Technical Report*, Zhengzhou University and Nanyang Technological University, 2014.
- [21] Y. Liu, Q. Cheng, Y. Gan, Y. Wang, Z. Li and J. Zhao, Multi-objective optimization of energy consumption in crude oil pipeline transportation system operation based on exergy loss analysis, *Neurocomputing*, vol. 332, pp. 100-110, Mar. 2019.
- [22] Y. Liu, S. Chen, B. Guan and P. Xu, Layout optimization of large-scale oil-gas gathering system based on combined optimization strategy, *Neurocomputing*, vol. 332, pp. 159-183, Mar. 2019.
- [23] L. Ma, J. Li, Q. Lin, M. Gong, C. A. C. Coello, and Z. Ming, "Reliable link inference for network data with community structures," *IEEE Transactions on Cybernetics*, vol. 49, no. 9, pp. 3347-3361, Aug. 2018.
- [24] L. Ma, J. Li, Q. Lin, M. Gong, C. A. C. Coello, and Z. Ming, "Cost-aware robust control of signed networks by using a memetic algorithm," *IEEE Transactions on Cybernetics*, 2019, doi: 10.1109/T-CYB.2019.2932996.
- [25] R. Motwani and P. Raghavan, *Randomized Algorithms*, Cambridge University Press, New York, 1995.
- [26] X. Pan, L. Xue, Y. Lu, and N. Sun, "Hybrid particle swarm optimization with simulated annealing," *Multimedia Tools and Applications*, vol. 78, no. 21, pp. 29921-29936, Nov. 2019.
- [27] I. U. Rahman, Z. Wang, W. Liu, B. Ye, M. Zakarya, and X. Liu, "An n-state Markovian jumping particle swarm optimization algorithm," *IEEE Transactions on Systems, Man, and Cybernetics - Systems*, 2020, doi:10.1109/TSMC.2019.2958550.
- [28] A. Ratnaweera, S. K. Halgamuge, and H. C. Watson, "Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients," *IEEE Transactions on Evolutionary Computation*, vol. 8, no. 3, pp. 240-255, Jun. 2004.
- [29] M. Schmitt, and R. Wanka, "Particle swarm optimization almost surely finds local optima," *Theoretical Computer Science*, vol. 561, pp. 57-72, Jan. 2015.
- [30] Y.-H. Shi and R. C. Eberhart, "Parameter selection in particle swarm optimization," In: *Proceedings of the 7th International Conference on Evolutionary Programming*, San Diego, USA, Mar. 1998, pp. 591-600.
- [31] Y.-H. Shi and R. C. Eberhart, "Empirical study of particle swarm optimization," In: *Proceedings of the 1999 IEEE Congress on Evolutionary Computation*, Washington DC, USA, Jul. 1999, pp. 1945-1950.
- [32] B. Song, Z. Wang, and L. Zou, "On global smooth path planning for mobile robots using a novel multimodal delayed PSO algorithm," *Cognitive Computation*, vol. 9, no. 1, pp. 5-17, Feb. 2017.
- [33] B. Song, Z. Wang, L. Zou, L. Xu, and F. E. Alsaadi, "A new approach to smooth global path planning of mobile robots with kinematic constraints," *International Journal of Machine Learning and Cybernetics*, vol. 10, no. 1, pp. 107-119, Jan. 2019.
- [34] M. Taherkhani and R. Safabakhsh, "A novel stability-based adaptive inertia weight for particle swarm optimization," *Applied Soft Computing*, vol. 38, pp. 281-295, Jan. 2016.
- [35] Y. Tang, Z. Wang, and J. Fang, "Parameters identification of unknown delayed genetic regulatory networks by a switching particle swarm optimization algorithm," *Expert Systems with Applications*, vol. 38, pp. 2523-2535, Mar. 2011.
- [36] D. W. Van der Merwe and A. P. Engelbrecht, "Data clustering using particle swarm optimization," In: *Proceedings of the 2003 IEEE Congress on Evolutionary Computation*, Canberra, Australia, Dec. 2003, vol. 1, pp. 215-220.
- [37] L. Wang, Z. Wang, Q.-L. Han, and G. Wei, "Event-based variance-constrained \mathcal{H}_∞ filtering for stochastic parameter systems over sensor networks with successive missing measurements," *IEEE Transactions on Cybernetics*, vol. 48, no. 3, pp. 1007-1017, Mar. 2018.
- [38] L. Wang, Z. Wang, G. Wei, and F. E. Alsaadi, "Finite-time state estimation for recurrent delayed neural networks with component-based event-triggering protocol," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 4, pp. 1046-1057, Apr. 2018.
- [39] X. Xiao, C. Mei, and G. Liu, "Improved particle swarm optimization algorithm based on random perturbations," In: *Proceedings of the 2010 Third International Joint Conference on Computational Science and Optimization*, Huangshan, China, May 2010, pp. 404-408.
- [40] X. Yao, Y. Liu, and G. Lin, "Evolutionary programming made faster," *IEEE Transactions on Evolutionary Computation*, vol. 3, no. 2, pp. 82-102, Jul. 1999.
- [41] N. Zeng, H. Zhang, Y. Chen, B. Chen, and Y. Liu, "Path planning for intelligent robot based on switching local evolutionary PSO algorithm," *Assembly Automation*, vol. 36, no. 2, pp. 120-126, Feb. 2016.
- [42] N. Zeng, Z. Wang, H. Zhang, and F. E. Alsaadi, "A novel switching delayed PSO algorithm for estimating unknown parameters of lateral flow immunoassay," *Cognitive Computation*, vol. 8, no. 2, pp. 143-152, Apr. 2016.
- [43] N. Zeng, H. Qiu, Z. Wang, W. Liu, H. Zhang, and Y. Li, "A new switching-delayed-PSO-based optimized SVM algorithm for diagnosis of Alzheimer's disease," *Neurocomputing*, vol. 320, pp. 195-202, Dec. 2018.
- [44] Z.-H. Zhan, J. Zhang, Y. Li, and H. S.-H. Chung, "Adaptive particle swarm optimization," *IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics*, vol. 39, no. 6, pp. 1362-1381, Dec. 2009.
- [45] Z.-H. Zhan, J. Zhang, Y. Li, and Y.-H. Shi, "Orthogonal learning particle swarm optimization," *IEEE Transactions on Evolutionary Computation*, vol. 15, no. 6, pp. 832-847, Dec. 2011.
- [46] L. Zou, Z. Wang, H. Gao and X. Liu, "State estimation for discrete-time dynamical networks with time-varying delays and stochastic disturbances under Round-Robin protocol", *IEEE Transactions on Neural Networks and Learning Systems*, vol. 28, no. 5, pp. 1139-1151, May 2017.
- [47] L. Zou, Z. Wang, Q.-L. Han and D. H. Zhou, Moving horizon estimation of networked nonlinear systems with random access protocol, *IEEE Transactions on Systems, Man, and Cybernetics-Systems*, in press, DOI: 10.1109/TSMC.2019.2918002.
- [48] L. Zou, Z. Wang, Q.-L. Han and D. H. Zhou, Full information estimation for time-varying systems subject to Round-Robin scheduling: A recursive filter approach, *IEEE Transactions on Systems, Man, and Cybernetics-Systems*, in press, DOI: 10.1109/TSMC.2019.2907620.