A Novel Sigmoid-Function-Based Adaptive Weighted Particle Swarm Optimizer

Weibo Liu, Zidong Wang, Yuan Yuan, Nianyin Zeng, Kate Hone and Xiaohui Liu

***Abstract*—In this paper, a novel particle swarm optimization (PSO) algorithm is put forward where a sigmoid-function-based weighting strategy is developed to adaptively adjust the acceleration coefficients. The newly proposed adaptive weighting strategy takes into account both the distances from the particle to the global best position and from the particle to its personal best position, thereby having the distinguishing feature of enhancing the convergence rate. Inspired by the activation function of neural networks, the new strategy is employed to update the acceleration coefficients by using the sigmoid function. The search capability of the developed adaptive weighting PSO (AWPSO) algorithm is comprehensively evaluated via eight well-known benchmark functions including both the unimodal and multimodal cases. Experiment results demonstrate that the designed AWPSO algorithm substantially improves the convergence rate of the particle swarm optimizer and also outperforms some currently popular PSO algorithms.**

***Index Terms*—Evolutionary computation, particle swarm optimization, acceleration coefficients, adaptive weighting, convergence rate.**

I. Introduction

Optimization problem has long been a fundamental research topic attracting an ever-increasing interest from a variety of communities owing to its clear application potential in almost all real-world systems including engineering systems, large-scaled complex networks, healthcare management systems and so on [1]–[4]. During the past decade, a great number of heuristic algorithms have been introduced with aim to effectively and efficiently solve the optimization problems (especially the NP-complete problems). In this regard, a famous heuristic approach, known as the particle swarm optimization (PSO) algorithm, has been successfully implemented in various practical applications in dealing with the optimization problems [5]. In a PSO algorithm, as motivated by the swarm intelligence and social behaviors (e.g., birds flocking), all the particles are randomly initialized and then encouraged to explore the problem space thoroughly based on the individual experience and the interaction with other particles [6], [7]. During the evolution process, the historically personal best position (*pbest*) of each particle as well as the historically global best position (*gbest*) discovered by the entire swarm are two important positions, based on which the particles are motivated to seek the optimal solution. According to [5], [8], the PSO algorithm has exhibited more competitive performance than many popular evolutionary computation approaches because of its easy implementation, fast convergence and comprehensive ability of converging to a satisfactory solution.

It is well known that, as a control parameter, the balance between global and local searches throughout the searching process plays a vital role in successfully finding the optimal solution [9], [10]. The inertia weight as well as acceleration coefficients, which serve as another two control parameters, are vitally important in the velocity updating model of the PSO algorithm and have been extensively investigated in recent years for better accuracy and faster convergence [9], [11], [12]. Up to now, some PSO variants have been focused on the modification of the aforementioned three control parameters. In [6], [7], a linear-decreasing-inertia-weight-based PSO (PSO-LDIW) algorithm has been proposed where the inertia weight is updated in a time-varying manner. For the purpose of efficiently controlling the local and global searches, the time-varying-acceleration-coefficient-based PSO (PSO-TVAC) algorithm has been introduced in [11]. In addition to the adaptation of the control parameters, topological structures have been introduced in some PSO algorithms with the hope to alleviate premature convergence, see e.g. [1], [3], [4], [13], [14]. In particular, time-delay terms have been taken into account through the velocity updating process due to their utilization of historical information during the evolution process which results in a better accuracy than the standard PSO algorithm, see e.g. [1], [3].

Although some popular PSO algorithms have exhibited competitive performance on searching the global optimum and increasing the possibility of avoiding the local optima, the enhancement of the search performance of PSO algorithms is often at the expense of sacrificing the convergence rate, which is certainly undesirable [15], [16]. As such, it is of practical significance to develop a new PSO algorithm that is capable of finding the globally optimal solution yet with a *satisfactory* convergence rate through adaptively updating the control parameters. Note that the inertia weight and acceleration coefficients only change along with time in most of the existing PSO algorithms. In this case, a seemingly natural idea is to make full use of the distances from each individual particle to its *pbest* and *gbest* at *each iteration*, and adaptively update the control parameters according to the outputs of a certain sigmoid function with the calculated distances as the inputs. In comparison with the time-varying parameter strategy

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 Engineering and Physical Sciences Research Council (EPSRC) of the UK, the Royal Society of the UK, and the Alexander von Humboldt Foundation of Germany. *(Corresponding author: Zidong Wang)*

W. Liu, Z. Wang, Y. Yuan, K. Hone and X. Liu are 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.

(see e.g. [6], [7], [11]), the sigmoid-function-based updating strategy possesses the following advantages: 1) the control parameters are adaptively chosen which could help guarantee the population diversity; and 2) the particles are motivated to move towards the *pbest* and *gbest* as fast as necessary which could help improve the convergence rate. It should be mentioned that the particles slow down once they get close to the *pbest* and *gbest*.

To summarize, the objective of our paper is to propose an adaptive weighting PSO (AWPSO) algorithm with a sigmoid-function-based parameter selection scheme. *The main contributions are outlined as follows: 1) a novel sigmoid-function-based AWPSO algorithm is proposed where an adaptive weighting strategy is designed to adaptively adjust the control parameters at each iteration; and 2) the acceleration coefficients are adaptively controlled according to the distances from the particle to its pbest and gbest, thereby facilitating a relatively fast exploitation of the problem space*.

The rest of this paper is organized below. The standard PSO algorithm and some popular PSO variants are studied in Section II. Section III describes the proposed adaptive weighting strategy and the AWPSO algorithm. Benchmark functions, test PSO algorithms, parameter setting, experiment results and discussions are illustrated in Section IV. Conclusions and future directions are presented in Section V.

II. PSO Algorithms

A. Basic PSO Algorithm

The PSO algorithm is a well-known population-based evolutionary computation method where each individual represents a candidate solution and a group of individuals refers to a swarm [5], [17]. For a N-dimensional optimization problem, the velocity of the ith particle is represented by a vector, i.e. $v_i = (v_{i1}, v_{i2}, \cdots, v_{iN})$. Similarly, the position vector of the ith particle is indicated by $x_i = (x_{i1}, x_{i2}, \cdots, x_{iN})$. The velocity and position updating equations of particle i are given as follows:

$$\begin{aligned} v_i(k+1) &= w \times v_i(k) + c_1 \times r_1 \times (p_i(k) - x_i(k)) \\ &\quad + c_2 \times r_2 \times (p_g(k) - x_i(k)) \\ x_i(k+1) &= x_i(k) + v_i(k+1) \end{aligned} \tag{1}$$

where k indicates the iteration number; w denotes the inertia weight; p_i represents the *pbest* of particle i; p_g is the *gbest* found by the entire swarm; c_1 is a constant value which is the cognitive acceleration coefficient, and c_2 is the social acceleration coefficient which is a constant value. r_1 and r_2 are two separate random numbers belonging to $[0, 1]$. It is worth mentioning that the position of the particle is limited to a specific range m which is the domain of the optimization problem.

B. Popular PSO Variants

A large number of researchers have devoted their effort to improving the search ability of the particle swarm optimizer. For example, the PSO-LDIW algorithm concentrates on the selection of the inertia weight [6], [7], where the updating equation of the inertia weight w at the kth iteration is given as follows:

$$w = w_1 - (w_1 - w_2) \times \frac{k}{maxiter} \tag{2}$$

where $maxiter$ represents the maximum iteration number, and w_1 and w_2 indicate the maximum and minimum inertia weight, respectively.

In a PSO algorithm, the inertia weight is normally utilized to balance the global search and the local search, where a larger value of the inertia weight contributes to a better global exploration, and a smaller value encourages a more thorough local exploitation [7]. The PSO-LDIW algorithm has satisfactory performance in many applications. However, for the PSO-LDIW algorithm, once the inertia weight decreases, the search ability of the swarm would be affected and new search areas cannot be explored [18]. Notably, similar to the inertia weight, acceleration coefficients have also attracted particular research interest for enhancing the search capability of the PSO algorithm. For example, in the PSO-TVAC algorithm [11], the cognitive acceleration coefficient c_1 is set to be linearly decreased and the social acceleration coefficient c_2 is set to be linearly increased. Moreover, the PSO algorithm with the constriction factor (PSO-CK) has been proposed in [19] where the constriction factor has been introduced to ensure the convergence of the PSO algorithm.

Apart from modifying the control parameters, some researchers have focused on designing different topological structures. With the newly proposed topological structures, the variant PSO algorithms may possess better population diversity or convergence than the standard PSO algorithm. In [13], an adaptive PSO (APSO) algorithm has been proposed with the introduction of an evolutionary factor to distinguish four evolutionary states and, with this learning strategy, the control parameters have been adaptively adjusted for the PSO algorithm. In [14], a switching PSO (SPSO) algorithm has been put forward to improve the convergence rate by updating the acceleration coefficients based on the switching of different evolutionary states. Recently, a competitive swarm optimizer (CSO) has been designed in [15] for large-scaled optimization problems where a pairwise competition mechanism is designed. With this pairwise competition mechanism, the particle that loses the competition adjusts the position according to the winner particle. More recently, time-delays have been employed in the PSO algorithms which change the system dynamics with the purpose of getting rid of local optima, see e.g., [1], [3]. Moreover, the time-delay terms consist of the historical information of the population evolution and the time-delayed PSO algorithms are then likely to have better accuracy than the classic PSO algorithm.

In the past few years, the traditional PSO algorithms have been improved in combination with the usage of some popular evolutionary computation approaches such as the differential evolution (DE) algorithm [20], [21] and the genetic algorithm (GA) [22], [23]. In particular, a switching local evolutionary PSO algorithm has been proposed in [21] by employing the DE algorithm to improve the search ability of the particles and increase the possibility of escaping from the local optima. A hybrid PSO-GA algorithm has been proposed in [23] where the genetic operators (e.g. crossover and mutation) are exploited to balance the global and local searching through the entire search space, and therefore ensure the satisfactory search ability of the particles.

III. A Novel AWPSO Algorithm

In a PSO algorithm, the acceleration coefficients are used to motivate the particles to move to the *pbest* and *gbest*. The distances from the position of each particle to its *pbest* and *gbest* are dominantly important in determining the movement of the particles. On the other hand, the adaptation of the control parameters is a significant factor in seeking the optimal solution with convincing efficiency and accuracy [11], [18]. Therefore, to control the PSO algorithm in an effective way, in this paper, we endeavor to propose a novel adaptive weighting mechanism with which the acceleration coefficients are adaptively adjusted as the iteration goes.

A. Adaptive Weighting Strategy

In the classic PSO algorithm, the velocity of an individual particle gets accelerated according to the *distances* from the particle to its *pbest* and *gbest*. As such, the selection of appropriate acceleration coefficients is of vital importance for finding the globally optimal solution through the problem space. In this case, it makes both theoretical and practical sense to *adaptively* update the acceleration coefficients iteration by iteration based on the aforementioned *distances* to efficiently improve the searching capability of the PSO algorithm.

In the literature, several popular updating strategies for acceleration coefficients have been proposed during the past decade with satisfactory performance [11] while avoiding premature convergence. Another PSO variant with a linearly decreasing strategy has been developed in [24] to update acceleration coefficients. However, these PSO variants only adjust the acceleration coefficients in a time-varying manner without taking the information of the population evolution into account.

It is clear that all the individuals are encouraged to explore the entire search space as much as possible in the early stage of the evolution process. Then, in the later stage of the optimization process, the individuals are motivated to converge to the global optimum and find the optimization solution as fast as possible. As can be seen in Eq. (1), the velocity of the particle updates is dependent on the distances from the particles to their own *pbest* and the *gbest*. In this case, it is reasonable to adjust the acceleration coefficient according to the distances from each individual particle to its *pbest* as well as the *gbest*.

Taking above all the mentioned concerns into consideration, an adaptive weighting strategy is proposed to adaptively control the acceleration coefficients. The main motivation is to accelerate the particles to find the optimal solution as fast as possible and thus enhances the convergence rate. Different from the time-varying updating strategy, the acceleration coefficients are altered according to the distance of the particle towards its *gbest* and *pbest*. If the particle is far away from its *pbest* and *gbest*, a relatively large acceleration coefficient is employed to accelerate the particle. However, the value of the acceleration coefficient is limited in an appropriate range to avoid premature convergence, which means that the velocity should be bounded to guarantee the searching capability of the algorithm.

Motivated by above discussions, we believe that an adaptive weighting updating function is appropriate to describe the relationship between the acceleration coefficient and the distances (from the particle to its *pbest* and *gbest*). In other words, the updates of the former acceleration coefficients should be *adaptive* to the latter distances, thereby fully justifying the velocity of the particle movements towards the global optimum. From a mathematical viewpoint, the proposed adaptive weighting updating rule can be described as follows:

$$\begin{aligned} c_{g_{pi}}(k) &= F(g_{pi}(k)) \\ c_{g_{gi}}(k) &= F(g_{gi}(k)) \end{aligned} \tag{3}$$

where the function $F(\cdot)$ represents the adaptive weighting updating function to be discussed later; and $g_{pi}(k)$ and $g_{gi}(k)$ are defined by

$$\begin{aligned} g_{pi}(k) &= p_i(k) - x_i(k) \\ g_{gi}(k) &= p_g(k) - x_i(k), \end{aligned} \tag{4}$$

which denote the distances from the particle i to its *pbest* and *gbest* at the kth iteration, respectively.

B. Selection of Adaptive Weighting Updating Function

Intuitively, the adaptive weighting updating function should have the following two properties: 1) the updating function is monotonically increasing; and 2) the updating function is bounded. The first property is mainly due to the characteristics of the acceleration coefficients. It is well known that the acceleration coefficients are the weighting terms which pull the particles to the *pbest* and *gbest*. A particle which is far away from its *pbest* and *gbest* requires a fast movement towards its *pbest* and *gbest*. Therefore, a monotonically increasing function is required. The second property is justified by the fact that the search space of a constrained optimization problem is normally bounded. Once a particle is close to its *pbest* and *gbest*, the movement should be slowed down to avoid missing its *pbest* and the *gbest*. Consequently, the acceleration coefficients should be bounded for the control of the velocity of the particle.

In search of adequate updating functions that are both monotonically increasing and uniformly bounded, the activation functions employed in neural networks appear to be ideal candidates. There are some popular activation functions for the neural networks such as step functions and sigmoid functions, among which we decide to select the sigmoid function as the adaptive weighting updating function for three reasons: 1) the sigmoid function is monotonic and bounded; 2) the curve of the sigmoid function is S-shaped and this would avoid undesirable abrupt changes of the control parameters; and 3) the sigmoid function is smooth and differentiable, thereby reflecting the adaptive/dynamic nature of the weight updating iteration by iteration.

According to the above discussion, in this paper, a sigmoid function is employed to adjust the acceleration coefficients as follows:

$$F(D) = \frac{b}{1 + e^{-a\times(D-c)}} + d \tag{5}$$

where e is the natural logarithm base; a denotes the steepness of the curve which is a constant value; b represents the peak value of the curve; c represents the abscissa value of the central point of the curve; d is a positive constant value; and D is the input of the function which is determined by Eq. (4). Specifically, D is the distance between the particle and its *pbest* for the cognitive acceleration coefficient. For the social acceleration coefficient, D indicates the distance between the particle and the *gbest*.

Remark 1: In Eq. (5), it is of vital importance to choose appropriate values of the four parameters (a, b, c and d). Note that a is the parameter which denotes the steepness of the curve. It seems a natural idea to adjust the value of a according to the search range of each individual optimization problem. In our work, we set $a = 0.000035 \cdot m$ where m indicates the search range of the optimization problem. According to the characteristics of the sigmoid function and experimental experience, b, c, d are set to be 0.5, 0, and 1.5, respectively.

To conclude, the three major advantages of the proposed sigmoid-function-based adaptive weighting strategy are summarized as follows:

1) the acceleration coefficients are adaptively controlled within reasonable bounds, and the adaptive weighting strategy ensures the efficiency of the velocity updating process;
2) the adaptive weighting updating function, chosen as the sigmoid function, is utilized to reflect the monotonic yet relatively smooth changes of the acceleration coefficients, where a larger distance will lead to a larger value of acceleration coefficient; and
3) the particles are motivated to seek the optimal solution as fast as *necessary*, thereby improving both the accuracy and the convergence.

C. Framework of the AWPSO Algorithm

An AWPSO algorithm is developed in this paper where the velocity updating equation obeys an adaptive weighting strategy. During the population evolution process, the velocity and position of the ith particle are updated on the basis of the following equations:

$$\begin{aligned} v_i(k+1) &= w \times v_i(k) + c_{g_{pi}}(k) \times r_1 \times g_{pi}(k) \\ &\quad + c_{g_{gi}}(k) \times r_2 \times g_{gi}(k) \\ x_i(k+1) &= x_i(k) + v_i(k+1) \end{aligned} \tag{6}$$

where w is the inertia weight; $g_{pi}(k)$ and $g_{gi}(k)$ represent the distances from the particle i to its *pbest* and *gbest* at the kth iteration, respectively; $c_{g_{pi}}(k)$ denotes the acceleration constant determined by $g_{pi}(k)$, and $c_{g_{gi}}(k)$ indicates the acceleration constant determined by $g_{gi}(k)$.

The flowchart of the introduced AWPSO algorithm is depicted in Fig. 1.

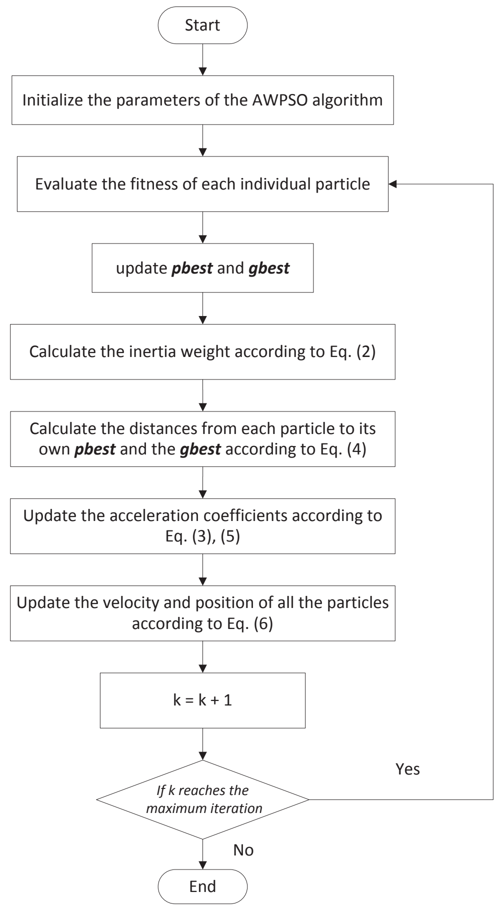

Fig. 1. Flowchart of the AWPSO algorithm

IV. Experiment Results and Discussions

In our paper, the AWPSO algorithm is compared with some popular variant PSO algorithms on a series of widely-used optimization benchmark functions consisting of both unimodal and multimodal cases for performance evaluation. In addition, the convergence performance of the adaptive weighting updating function is demonstrated with visible results. For all the benchmark functions, the swarm size is set to be 30 and the dimension of the problem space is set to be 30. In this simulation, each experiment has been repeated for 50 times independently, and the maximum iteration number is set to be 5000. It is worth pointing out that the Euclidean distance is chosen as the distance metric in this paper.

A. Benchmark Functions

It should be noticed that all the selected benchmark functions have been widely used in the evolutionary computing community [1], [3], [25]. The Sphere function $f_1(x)$ is a typical unimodal function. The Rosenbrock function $f_2(x)$ is called as the Rosenbrock's banana function which is a popular benchmark function. The Rastrigin function $f_3(x)$, the Penalized 1 function $f_5(x)$ and the Penalized 2 function $f_8(x)$ are classical multimodal problems consisting of many local optima, which are difficult to find the globally optimal solution. The Schwefel 2.22 function $f_4(x)$ and the Step function $f_6(x)$

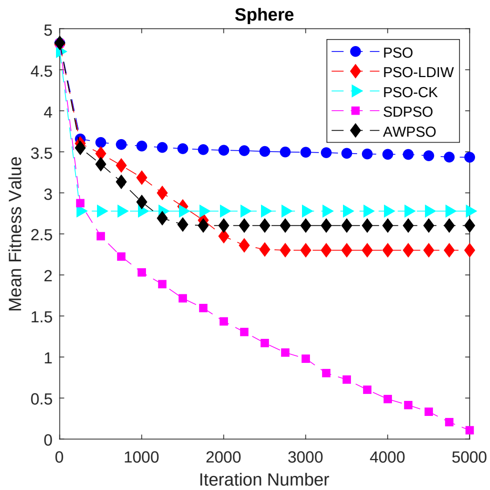

Fig. 2. Optimization performance for Sphere function $f_1(x)$

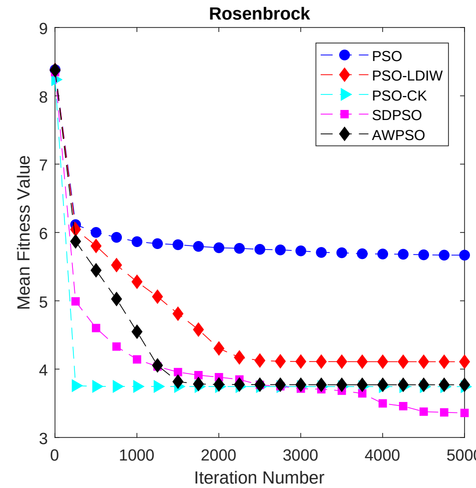

Fig. 3. Optimization performance for Rosenbrock function $f_2(x)$

are also frequently used benchmark functions for optimization. The Schwefel function $f_7(x)$ is a typical benchmark function with lots of local minima. The configurations of the benchmark functions are presented in Table I. The search range represents the range of the search space. Additionally, the threshold is a problem-based parameter which is utilized as a stopping criterion of the algorithm.

B. Experiment Results

In our paper, four currently popular PSO algorithms (including the basic PSO algorithm [5], the PSO-LDIW algorithm [6], the PSO-CK algorithm [19], and the SDPSO algorithm [3]) are selected for performance evaluation via eight widely-used benchmark functions.

Experiment results are displayed in Figs. 2-9 where the vertical coordinate indicates the mean fitness value in the logarithmic form, and the horizontal coordinate indicates the iteration number. From the figures, we can see that the AW-PSO algorithm exhibits competitive performance on most of the benchmark functions. Although the PSO-LDIW algorithm obtains better mean fitness value than the AWPSO algorithm on most of the benchmark functions, the superiority is not obvious. Furthermore, it is apparent that the AWPSO algorithm converges faster than most of the benchmark functions with satisfactory mean fitness value.

In this paper, the diversity of the swarm at the kth iteration is calculated as follows [26]:

$$S(k) = \frac{1}{M} \sum_{i=1}^{M} \sqrt{\sum_{j=1}^{N} (x_{ij}(k) - \bar{x}_j(k))^2} \tag{7}$$

where M is the swarm size, N is the dimensionality of the optimization problem, x_{ij} denotes the ith particle at the jth dimension, $\bar{x}_j(k)$ is the average value of the jth dimension over all particles at the kth iteration, i.e. $\bar{x}_j(k) = \frac{1}{M} \sum_{i=1}^{M} x_{ij}(k)$.

The population diversity of the classic PSO algorithm and our proposed AWPSO algorithm are shown in Figs. 10-17, where the vertical coordinate represents the diversity measure

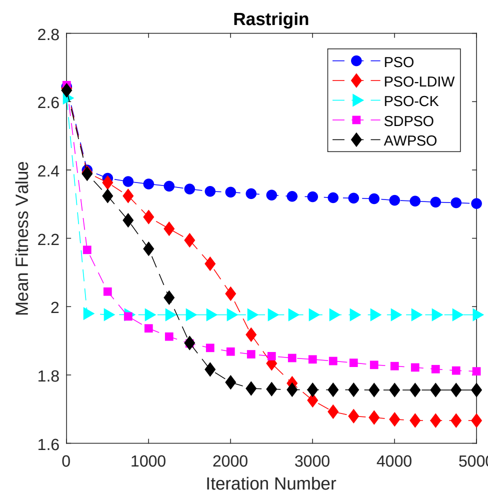

Fig. 4. Optimization performance for Rastrigin function $f_3(x)$

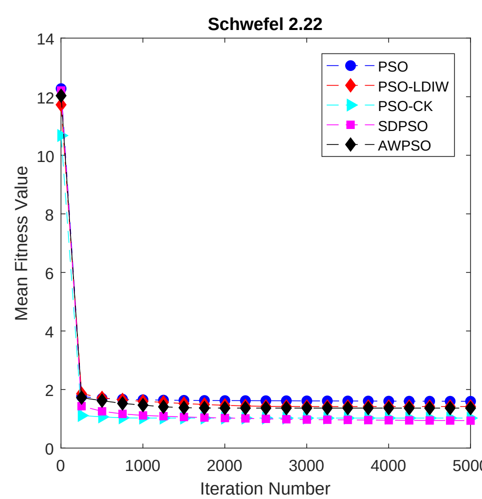

Fig. 5. Optimization performance for Schwefel 2.22 function $f_4(x)$

TABLE I
CONFIGURATION OF BENCHMARK FUNCTIONS

Function Number	Function Name	Problem Dimension	Search Range	Minimum	Threshold
$f_1(x)$	Sphere	30	$[-100, 100]$	0	0.1
$f_2(x)$	Rosenbrock	30	$[-30, 30]$	0	100
$f_3(x)$	Rastrigin	30	$[-5.12, 5.12]$	0	50
$f_4(x)$	Schwefel 2.22	30	$[-10, 10]$	0	0.1
$f_5(x)$	Penalized 1	30	$[-50, 50]$	0	0.1
$f_6(x)$	Step	30	$[-100, 100]$	0	0.1
$f_7(x)$	Schwefel	30	$[-500, 500]$	0	0.1
$f_8(x)$	Penalized 2	30	$[-50, 50]$	0	0.1

TABLE II
ALGORITHM EVALUATION ON EIGHT BENCHMARK FUNCTIONS

		PSO	PSO-LDIW	PSO-CK	SDPSO	AWPSO
$f_1(x)$	Minimum	1.75×10^3	2.03×10^{-33}	8.13×10^{-87}	4.11×10^{-3}	5.25×10^{-45}
	Mean	2.72×10^3	2.00×10^2	6.00×10^2	1.2816	4.00×10^2
	Std. Dev.	1.44×10^3	1.41×10^3	2.40×10^3	1.9592	1.98×10^3
	Ratio	0%	98%	94%	22%	96%
$f_2(x)$	Minimum	2.70×10^5	2.49×10^{-2}	1.51×10^{-4}	7.77×10^1	3.95×10^{-2}
	Mean	4.66×10^5	1.28×10^4	5.54×10^3	2.29×10^3	5.90×10^3
	Std. Dev.	1.07×10^5	3.15×10^4	2.16×10^4	1.27×10^4	2.15×10^4
	Ratio	0%	74%	88%	4%	68%
$f_3(x)$	Minimum	1.68×10^2	1.19×10^1	4.88×10^1	2.97×10^1	2.39×10^1
	Mean	2.00×10^2	4.64×10^1	9.46×10^1	6.47×10^1	5.70×10^1
	Std. Dev.	1.75×10^1	2.31×10^1	2.59×10^1	2.42×10^1	2.27×10^1
	Ratio	0%	62%	2%	36%	42%
$f_4(x)$	Minimum	1.89×10^1	6.32×10^{-22}	1.73×10^{-26}	1.13×10^{-2}	1.98×10^{-16}
	Mean	3.96×10^1	2.62×10^1	1.06×10^1	8.6727	2.32×10^1
	Std. Dev.	1.47×10^1	1.82×10^1	1.02×10^1	1.41×10^1	1.63×10^1
	Ratio	0%	10%	36%	4%	12%
$f_5(x)$	Minimum	1.81×10^1	1.57×10^{-32}	1.57×10^{-32}	1.08×10^{-4}	1.57×10^{-32}
	Mean	9.23×10^1	8.29×10^{-3}	2.77×10^{-1}	2.38×10^{-1}	2.70×10^{-2}
	Std. Dev.	1.47×10^2	2.84×10^{-2}	4.44×10^{-1}	3.37×10^{-1}	4.59×10^{-2}
	Ratio	0%	92%	46%	54%	74%
$f_6(x)$	Minimum	1.62×10^3	0.0000	0.0000	0.0000	0.0000
	Mean	3.17×10^3	2.00×10^{-2}	1.01×10^3	4.7400	4.00×10^{-2}
	Std. Dev.	2.40×10^3	1.41×10^{-1}	3.03×10^3	4.4758	1.98×10^{-1}
	Ratio	0%	98%	14%	14%	96%
$f_7(x)$	Minimum	4.78×10^3	1.90×10^3	3.22×10^3	3.21×10^3	1.54×10^3
	Mean	6.60×10^3	3.64×10^3	4.90×10^3	5.06×10^3	3.70×10^3
	Std. Dev.	1.03×10^3	1.55×10^3	8.85×10^2	1.30×10^3	2.23×10^3
	Ratio	0%	0%	0%	0%	0%
$f_8(x)$	Minimum	2.84×10^4	4.18×10^{-32}	1.35×10^{-32}	2.21×10^{-2}	1.35×10^{-32}
	Mean	1.35×10^5	2.42×10^{-3}	2.02×10^{-1}	5.49×10^{-1}	3.07×10^{-3}
	Std. Dev.	6.90×10^4	7.12×10^{-3}	6.19×10^{-1}	4.56×10^{-1}	8.59×10^{-3}
	Ratio	0%	100%	84%	10%	100%

of the swarm and the horizontal coordinate indicates the number of iteration. It can be seen that both the classic PSO algorithm and the AWPSO algorithm have large values of population diversity at the early stage of the optimization process. The population diversity of the classic PSO algorithm and the AWPSO algorithm decreases as the iteration number increases. It is worth mentioning that a small value of population diversity implies that the population converges to a certain region of the search space. We can see that the population diversity of the AWPSO algorithm is smaller than that of the classic PSO algorithm at the later stage of the optimization process, which indicates that the convergence of the AWPSO algorithm is better than the classic PSO algorithm. To summarize, our proposed AWPSO algorithm can maintain the population diversity by adaptively adjusting the control parameters through the optimization process.

The statistical results of the PSO algorithms are illustrated in Table II. Notably, the minimum, standard deviation and mean fitness value are utilized to evaluate the searching capability of the particle swarm optimizers. The success ratio is used to judge the convergence characteristics, which demonstrates the PSO algorithms' capability of getting rid of the local optima. Notice that all the selected benchmark functions are minimization problems. As such, a smaller fitness value indicates a better solution. In Table II, the proposed AWPSO algorithm obtains smaller minimum fitness value than the classic PSO algorithm, the PSO-LDIW algorithm, the SDPSO algorithm for function $f_1(x)$. In addition, the AWPSO algorithm exhibits better performance than the basic PSO algorithm, the PSO-CK algorithm and the SDPSO algorithm for function $f_3(x)$. We can see that the minimum fitness value of the AWPSO algorithm is the smallest comparing with all other PSO algo-

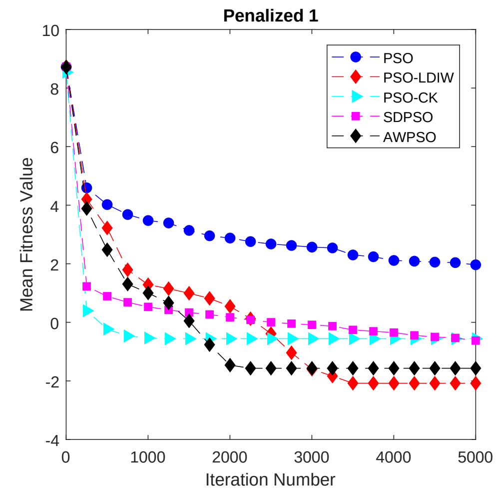

Fig. 6. Optimization performance for Penalized 1 function $f_5(x)$

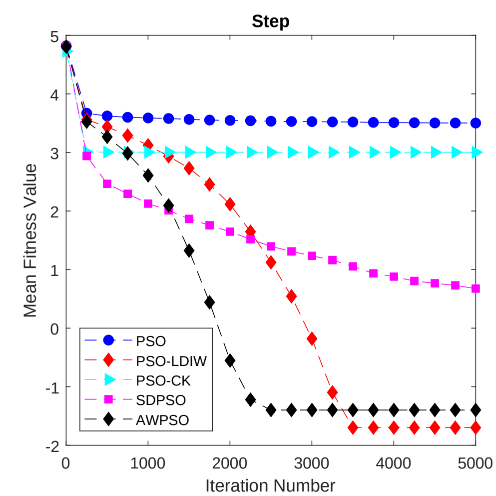

Fig. 7. Optimization performance for Step function $f_6(x)$

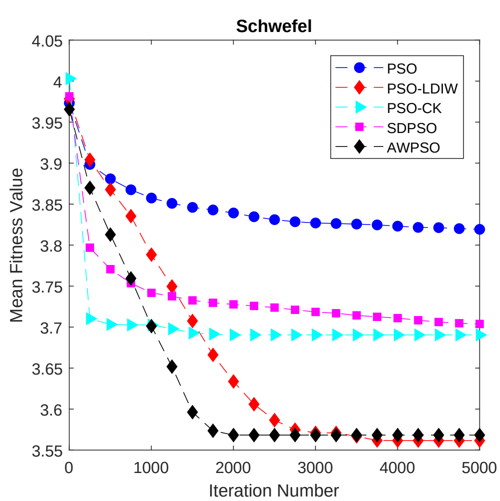

Fig. 8. Optimization performance for Schwefel function $f_7(x)$

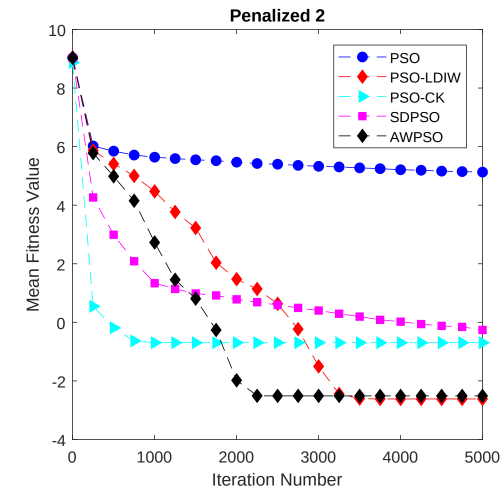

Fig. 9. Optimization performance for Penalized 2 function $f_8(x)$

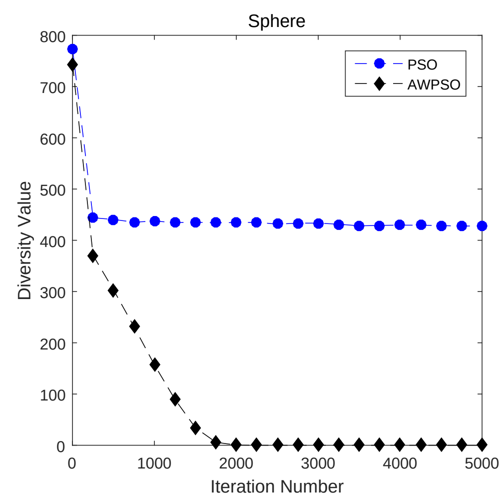

Fig. 10. Diversity measure for Sphere function $f_1(x)$

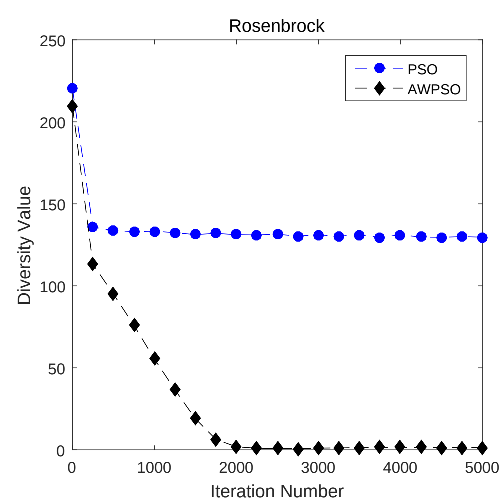

Fig. 11. Diversity measure for Rosenbrock function $f_2(x)$

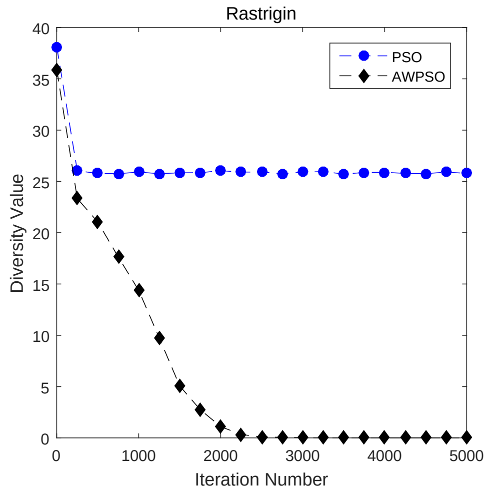

Fig. 12. Diversity measure for Rastrigin function $f_3(x)$

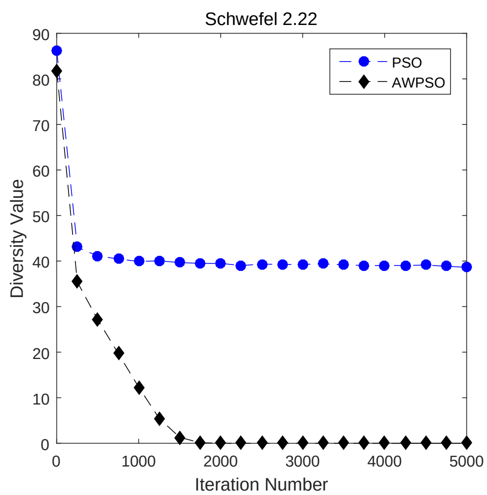

Fig. 13. Diversity measure for Schwefel 2.22 function $f_4(x)$

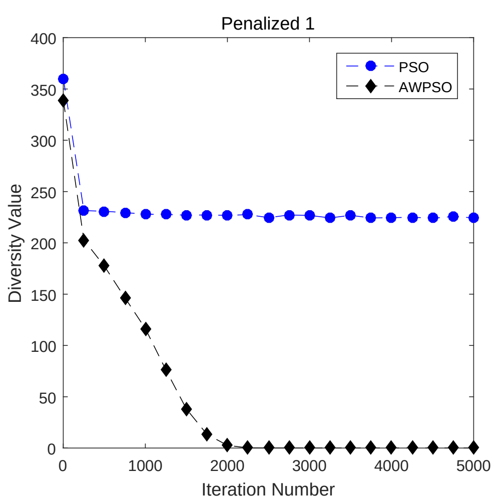

Fig. 14. Diversity measure for Penalized 1 function $f_5(x)$

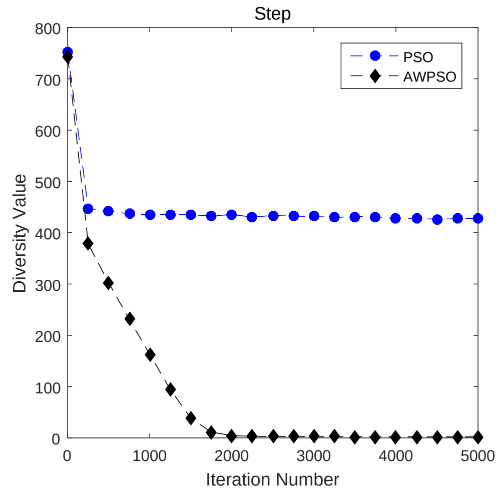

Fig. 15. Diversity measure for Step function $f_6(x)$

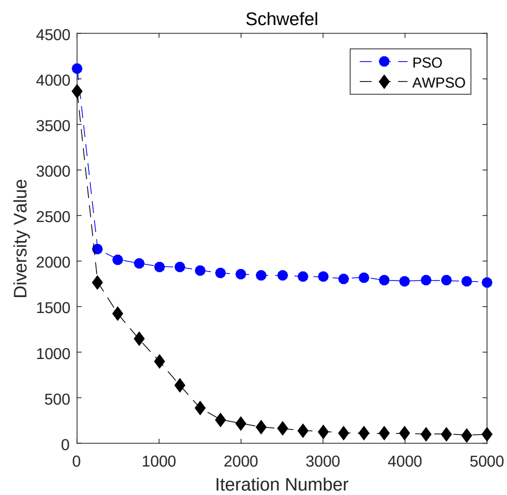

Fig. 16. Diversity measure for Schwefel function $f_7(x)$

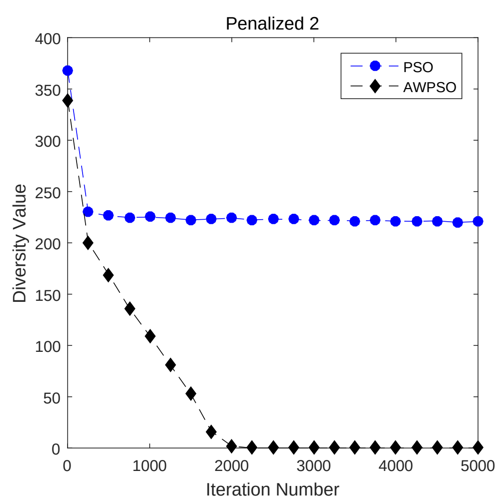

Fig. 17. Diversity measure for Penalized 2 function $f_8(x)$

rithms for functions $f_5(x)$ to $f_8(x)$. The standard deviation of the AWPSO algorithm is neither too large nor small which indicates that the population diversity of AWPSO algorithm is satisfactory. Moreover, the AWPSO algorithm achieves the satisfactory results for most of the benchmark functions by comparing the mean fitness value.

On the other hand, the success ratio is an important criterion to evaluate the evolutionary algorithms. In Table II, only the PSO-LDIW algorithm and the AWPSO algorithm achieve 100% success ratio on function $f_8(x)$, which indicates the difficulty of finding the global optimum for the selected benchmark functions. Note that the success ratio of all the benchmark algorithms for Rastrigin function $f_3(x)$ and the Schwefel function $f_7(x)$ are not satisfactory because these two functions have a large number of local minima, which are hard to find the globally optimal solution, and thus results in a low success ratio. Comparing the success ratio of the PSO algorithms, the AWPSO algorithm demonstrates competitive performance on most of the benchmark functions.

Note that the convergence rate is also a significant performance indicator. In this paper, the stopping criterion is set as the algorithm finds the globally optimal solution within the threshold. In this case, a smaller number of iteration indicates a better convergence performance of the PSO algorithm. To avoid random phenomena, we repeat the experiment for 50 times on each benchmark function and calculate the mean iteration number. The convergence plot of PSO algorithms is depicted in Fig. 18 where the vertical coordinate denotes the number of iteration when the algorithm converges, and the horizontal coordinate represents the number of benchmark function. In Fig. 18, we can see that the AWPSO algorithm outperforms the basic PSO algorithm, the PSO-LDIW algorithm and the SDPSO algorithm. The PSO-CK algorithm converges faster than the AWPSO algorithm on function $f_1(x)$, function $f_2(x)$, function $f_4(x)$ and function $f_8(x)$. Nevertheless, it is worth mentioning that the overall difference of average convergence rate between the AWPSO algorithm and the PSO-CK algorithm is not large. Importantly, the AWPSO algorithm demonstrates higher success ratio than the PSO-CK algorithm. Therefore, we could arrive at the conclusion that the proposed AWPSO algorithm demonstrates competitive performance on the population diversity and the convergence rate.

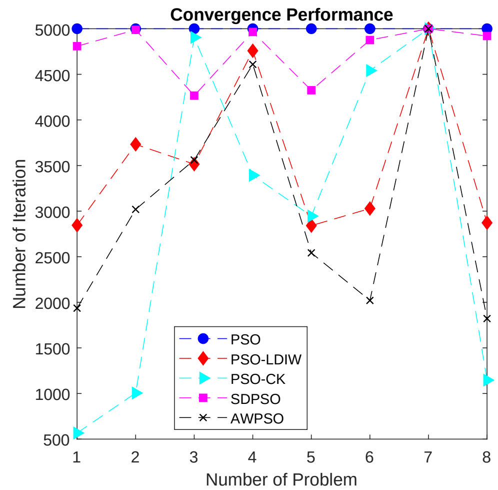

Fig. 18. Convergence plot of PSO algorithms

V. Conclusion

A novel PSO algorithm called the AWPSO algorithm has been proposed in this paper with the hope to improve the convergence rate of the traditional particle swarm optimizer. A sigmoid-function-based adaptive weighting strategy has been introduced where the acceleration coefficients are adaptively controlled by employing a sigmoid function based on the distances from the particle to the global best position and from the particle to its personal best position. The AWPSO algorithm has demonstrated competitive performance on the convergence rate by comparing with four popular PSO algorithms on eight widely-used optimization benchmark functions including both unimodal and multimodal cases. In our future research directions, we aim to 1) improve the AWPSO algorithm in terms of the population diversity and study the movement behaviors of particles by using the Wilcoxon rank sum test [27], [28]; and 2) apply the AWPSO algorithm to other research fields, such as system engineering and signal processing [29]–[36].

References

[1] 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.

[2] 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.

[3] 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.

[4] 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.

[5] 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.

[6] Y. 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.

[7] Y. 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.

[8] R. C. Eberhart and Y. 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.

[9] 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.

[10] X. Yu, W.-N. Chen, T. Gu, H. Zhang, H. Yuan, S. Kwong, and J. Zhang, “Set-based discrete particle swarm optimization based on decomposition for permutation-based multiobjective combinatorial optimization problems,” *IEEE Transactions on Cybernetics*, vol. 48, no. 7, pp. 2139-2153, Jul. 2018.

[11] 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.

[12] Z.-H. Zhan, J. Li, J. Cao, J. Zhang, H. S.-H. Chung, and Y. Shi, "Multiple populations for multiple objectives: a coevolutionary technique for solving multiobjective optimization problems," *IEEE Transactions on Cybernetics*, vol. 43, no. 2, pp. 445-463, Apr. 2013.

[13] 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.

[14] 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.

[15] 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.

[16] W.-N. Chen, J. Zhang, Y. Lin, N. Chen, Z.-H. Zhan, H. S.-H. Chung, Y. Li, and Y. Shi, "Particle swarm optimization with an aging leader and challengers," *IEEE Transactions on Evolutionary Computation*, vol. 17, no. 2, pp. 241-258, Apr. 2013.

[17] R. C. Eberhart and Y. 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.

[18] 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.

[19] 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, Aug. 2002.

[20] N. Zeng, Y. S. Hung, Y. Li, and M. Du, "A novel switching local evolutionary PSO for quantitative analysis of lateral flow immunoassay," *Expert Systems with Application*, vol. 41, no. 4, pp. 1708-1715, Mar. 2014.

[21] 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.

[22] P. Ghamisi and J. A. Benediktsson, "Feature selection based on hybridization of genetic algorithm and particle swarm optimization," *IEEE Geoscience and Remote Sensing Letters*, vol. 12, no. 2, pp. 309-313, Jul. 2015.

[23] H. Garg, "A hybrid PSO-GA algorithm for constrained optimization problems," *Applied Mathematics and Computation*, vol. 274, pp. 292-305, Feb. 2016.

[24] P. N. Suganthan, "Particle swarm optimiser with neighbourhood operator," In: *Proceedings of 1999 IEEE Congress on Evolutionary Computation*, Washington DC, USA, Jul. 1999, pp. 1958-1962.

[25] 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.

[26] R. Thangaraj, M. Pant, and A. Abraham, "A new diversity guided particle swarm optimization with mutation," In: *Proceedings of the 2009 World Congress on Nature & Biologically Inspired Computing*, Coimbatore, India, Dec. 2009, pp. 294-299.

[27] Q. Lu, Q.-L. Han, and S. Liu, "A finite-time particle swarm optimization algorithm for odor source localization," *Information Sciences*, vol. 277, pp. 111-140, Sept. 2014.

[28] Q. Lu, Q.-L. Han, and S. Liu, "A cooperative control framework for a collective decision on movement behaviors of particles," *IEEE Transactions on Evolutionary Computation*, vol. 20, no. 6, pp. 859-873, Dec. 2016.

[29] F. Chen, L. Wang, B. Jiang, and C. Wen, "An arterial traffic signal control system based on a novel intersections model and improved hill climbing algorithm," *Cognitive Computation*, vol. 7, no. 4, pp. 464-476, Aug. 2015.

[30] J. Hu, Z. Wang, and H. Gao, "Joint state and fault estimation for uncertain time-varying nonlinear systems with randomly occurring faults and sensor saturations," *Automatica*, vol. 97, pp. 150-160, Nov. 2018.

[31] S. Liu, Z. Wang, G. Wei, and M. Li, "Distributed set-membership filtering for multirate systems under the Round-Robin scheduling over sensor networks," *IEEE Transactions on Cybernetics*, to be published, doi: 10.1109/TCYB.2018.2885653.

[32] X. Wan, Z. Wang, M. Wu, and X. Liu, "State estimation for discrete time-delayed genetic regulatory networks with stochastic noises under the Round-Robin protocols," *IEEE Transactions on Nanobioscience*, vol. 17, no. 2, pp. 145-154, Apr. 2018.

[33] L. Wang, Z. Wang, G. Wei, and F. E. Alsaadi, "Observer-based consensus control for discrete-time multiagent systems with coding-decoding communication protocol," *IEEE Transactions on Cybernetics*, to be published, doi: 10.1109/TCYB.2018.2863664.

[34] N. Zeng, Z. Wang, Y. Li, M. Du, and X. Liu, "A hybrid EKF and switching PSO algorithm for joint state and parameter estimation of lateral flow immunoassay models," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 9, no. 2, pp. 321-329, Mar. 2012.

[35] L. Zou, Z. Wang, Q.-L. Han and D. Zhou, "Recursive filtering for time-varying systems with random access protocol," *IEEE Transactions on Automatic Control*, vol. 64, no. 2, pp. 720-727, Feb. 2019.

[36] L. Zou, Z. Wang, Q.-L. Han and D. Zhou, "Moving horizon estimation for networked time-delay systems under Round-Robin protocol," *IEEE Transactions on Automatic Control*, to be published, doi: 10.1109/TAC.2019.2910167.

Weibo Liu received his B.S. degree in electrical engineering from the Department of Electrical Engineering & Electronics, University of Liverpool, Liverpool, UK, in 2015. He is currently pursuing the Ph.D. degree in Computer Science at Brunel University London, Uxbridge, UK. His research interests include big data analysis and deep learning techniques.

Zidong Wang (SM'03-F'14) was born in Jiangsu, China, in 1966. He received the B.Sc. degree in mathematics in 1986 from Suzhou University, Suzhou, China, and the M.Sc. degree in applied mathematics in 1990 and the Ph.D. degree in electrical engineering in 1994, both from Nanjing University of Science and Technology, Nanjing, China.

He is currently a Professor of Dynamical Systems and Computing in the Department of Computer Science, Brunel University London, U.K. From 1990 to 2002, he held teaching and research appointments in universities in China, Germany and the UK. Prof. Wang's research interests include dynamical systems, signal processing, bioinformatics, control theory and applications. He has published more than 400 papers in refereed international journals. He is a holder of the Alexander von Humboldt Research Fellowship of Germany, the JSPS Research Fellowship of Japan, William Mong Visiting Research Fellowship of Hong Kong.

Prof. Wang serves (or has served) as the Editor-in-Chief for Neurocomputing, Deputy Editor-in-Chief for International Journal of Systems Science, and an Associate Editor for 12 international journals including IEEE Transactions on Automatic Control, IEEE Transactions on Control Systems Technology, IEEE Transactions on Neural Networks, IEEE Transactions on Signal Processing, and IEEE Transactions on Systems, Man, and Cybernetics-Part C. He is a Fellow of the IEEE, a Fellow of the Royal Statistical Society and a member of program committee for many international conferences.

Yuan Yuan was born in Xian, Shaanxi Province, China in 1986. He received his B.Sc degree in the School of Instrumental Science and Opto-electronics Engineering, Beihang University, Beijing, China in 2009. In 2015, he received his Ph.D. degree in computer science and technology from Department of Computer Science and Technology, Tsinghua University, Beijing, China. He is currently a research fellow in the Department of Computer Science, Brunel University London, Uxbridge, U.K. His research interests include anti-disturbance control, stochastic control, and game theoretic control with their applications to the aerospace systems.

Nianyin Zeng was born in Fujian Province, China, in 1986. He received the B.Eng. degree in electrical engineering and automation in 2008 and the Ph.D. degree in electrical engineering in 2013, both from Fuzhou University. From October 2012 to March 2013, he was a Research Assistant in the Department of Electrical and Electronic Engineering at the University of Hong Kong.

Currently, he is an Assistant Professor with the Department of Instrumental & Electrical Engineering of Xiamen University. Especially, Dr. Zeng is an ISEF Fellow founded by the Korea Foundation for Advance Studies and also a Visiting Professor at the Korea Advanced Institute of Science and Technology from September 2017. His current research interests include intelligent data analysis, computational intelligent, time-series modeling and applications. He is the author or co-author of several technical papers and also a very active reviewer for many international journals and conferences.

Dr. Zeng is currently serving as an Associate Editor for Neurocomputing, and also Editorial Board members for Computers in Biology and Medicine and Biomedical Engineering Online.

Kate Hone received the B.A. degree in Experimental Psychology from the University of Oxford, U.K. and the M.Sc. Degree in Work Design and Ergonomics and the Ph.D. degree from the University of Birmingham, U.K.

Professor Hone is Head of the Department of Computer Science at Brunel University London, U.K. She previously held academic posts at the University of Nottingham, U.K. before joining Brunel in 2000. At Brunel she has held a number of posts including Director of the Graduate School between 2009 and 2018.

Professor Hones research interests include spoken dialogue systems, affective computing, social signals processing, health informatics and intelligent data analysis.

Xiaohui Liu received the B.Eng. degree in computing from Hohai University, Nanjing, China, in 1982 and the Ph.D. degree in computer science from Heriot-Watt University, Edinburgh, U.K., in 1988.

He is currently a Professor of Computing at Brunel University. He leads the Intelligent Data Analysis (IDA) Group, performing interdisciplinary research involving artificial intelligence, dynamic systems, image and signal processing, and statistics, particularly for applications in biology, engineering and medicine. Professor Liu serves on editorial boards of four computing journals, founded the biennial international conference series on IDA in 1995, and has given numerous invited talks in bioinformatics, data mining and statistics conferences.