This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TEVC.2018.2878536, IEEE Transactions on Evolutionary Computation

A Novel Particle Swarm Optimization Approach for Patient Clustering from Emergency Departments

Weibo Liu, Zidong Wang, Xiaohui Liu, Nianyin Zeng and David Bell

***Abstract*—In this paper, a novel particle swarm optimization (PSO) algorithm is proposed in order to improve the accuracy of traditional clustering approaches with applications in analyzing real-time patient attendance data from an accident & emergency (A&E) department in a local UK hospital. In the proposed randomly occurring distributedly delayed particle swarm optimization (RODDPSO) algorithm, the evolutionary state is determined by evaluating the evolutionary factor in each iteration, based on which the velocity updating model switches from one mode to another. With the purpose of reducing the possibility of getting trapped in the local optima and also expanding the search space, randomly occurring time-delays that reflect the history of previous personal best and global best particles are introduced in the velocity updating model in a distributed manner. Eight well-known benchmark functions are employed to evaluate the proposed RODDPSO algorithm which is shown via extensive comparisons to outperform some currently popular PSO algorithms. To further illustrate the application potential, the RODDPSO algorithm is successfully exploited in the patient clustering problem for data analysis with respect to a local A&E department in West London. Experiment results demonstrate that the RODDPSO-based clustering method is superior over two other well-known clustering algorithms.**

***Index Terms*—Accident & emergency departments, clustering, distributed time-delay, evolutionary computation, particle swarm optimization.**

I. Introduction

Accident & emergency (A&E) departments in National Health Service (NHS) in the UK are open for 24 hours and 365 days a year. Targets for A&E departments aim to ensure that at least 98% of patients are treated from arrival to discharge, transfer or admission within 4 hours. Recently, increasing numbers of emergency cases are leading to overcrowding in many A&E departments, which causes that many A&E departments suffer from financial pressures [1], [41]. Furthermore, a number of non-emergency patients go to the A&E departments, which leads to the increasing burden on the human and medical resources. Note that the identification of illness severity plays an important role in medical resources management. Grouping patients with an appropriate triage category is an important element in improving the efficiency of medical treatment. Therefore, it is vital to identify the triage category of the patients, which can be treated as a clustering problem.

Clustering techniques are used to discover the natural groupings of a set of objects where the objects in the same cluster share similar characteristics. During the past few decades, clustering techniques have been successfully employed in a variety of research areas such as biology, signal processing, computer vision, market segmentation, and healthcare, see e.g., [7], [14], [30], [35], [37]. It has been shown in [6], [25] that many popular clustering algorithms are heavily dependent on the initial state of cluster centroids, and may get trapped in local optima. As such, it is reasonable to *optimize* the parameters of clustering algorithms (e.g. the number of clusters and the initial state of cluster centroids) in order to improve the clustering performance. It is well known that evolutionary computation (EC) serves as a powerful family of algorithms that can be effectively used to solve global optimization problems by using stochastic or metaheuristic searching strategies. Some important EC approaches, which include evolutionary programming, evolutionary strategies, genetic algorithms and generic programming, are motivated by biological evolution and have been successfully applied to various research fields such as artificial intelligence, see [8], [22]. In this context, various EC algorithms have been applied to optimally set the parameters with the purpose of improving the clustering performance with examples including the genetic algorithm (GA) [16], [20], the simulated annealing (SA) algorithm [26], [29], the particle swarm optimization (PSO) algorithm [19], [38], and the artificial bee colony (ABC) [52] algorithm.

Among the EC algorithms, the PSO algorithm which is a population-based heuristic algorithm has received much research attention owing to its easy implementation and competitiveness in finding a relatively satisfactory solution with a reasonable convergence speed, see e.g., [11], [12], [23], [34], [36], [39], [40], [48], [51]. Moreover, among the EC-based clustering approaches, PSO algorithms have proven to be a strong competitor to other optimization algorithms [2], [18], [38], [47]. For instance, a PSO-based clustering technique has been proposed in [38] where the initial swarm adopts clusters formed by the K-means clustering algorithm. A hybrid PSO-based clustering algorithm has been developed in [47] for gene clustering by employing the self-organizing map algorithm. Recently, a hybrid fuzzy clustering algorithm on the basis of the conventional PSO algorithm and fuzzy C-means clustering algorithm has been proposed in [18] with satisfactory performance on several well-known benchmark data sets. Very recently, a density-based PSO algorithm has

This work was supported in part by the the European Union's Horizon 2020 Research and Innovation Programme under Grant 820776 (INTEGRADDE), 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, X. Liu and D. Bell are with the Department of Computer Science, Brunel University London, Uxbridge, Middlesex, UB8 3PH, United Kingdom. (emails: Zidong.Wang@brunel.ac.uk)

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

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TEVC.2018.2878536, IEEE Transactions on Evolutionary Computation

been introduced in [2] for data clustering by combining the kernel density estimation method with the PSO algorithm.

As with almost all EC algorithms, the PSO algorithms suffer from the problem of trapping local optima especially in high-dimensional optimization processes. Consequently, it is of practical significance to develop advanced approaches to further improve the search ability of the PSO algorithms in terms of both the convergence and the diversity. It should be mentioned that the PSO algorithms perform well by adding certain time-delays in the velocity updating model, see [34], [36], [48]. In the existing delayed PSO algorithms, the time-delay terms (composed of both personal and global best particles in the velocity updating model) contribute significantly to the full use of historical information and the thorough exploration of the search space, by which the convergence behaviors of PSO algorithms are improved and the capability of getting rid of local optima is enhanced. Time-delay is a physical phenomenon existing in dynamical systems such as single-frequency global positioning systems [13] and genetic regulatory networks [36]. According the way they occur, time delays can be categorized as constant, time-varying, discrete and distributed ones, see e.g. [33], [44].

Distributed time-delays exhibit a distinct spatial nature that models delays in signal propagations distributed through an amount of parallel channels/pathways during a certain time period. So far, the dynamical behaviors of complex systems (e.g. neural networks [33], [44]) with distributed time-delays have been well studied. Intuitively, a natural idea is to introduce certain distributed time-delays in the PSO algorithm with the hope to enhance the capability of escaping from the local optima and getting rid of the problem of premature convergence. As compared with the discrete time-delays in [34], [36], [48], distributed time-delays could have the following two advantages: 1) a better use of longer (more accumulated) history of the population evolution leading to a better accuracy; and 2) a more complicated dynamical behavior leading to less possibility of trapping local optima. Furthermore, to play an adequate tradeoff between the convergence and the diversity, the introduced distributed time-delays could be made randomly occurring with reasonably small probability. As such, the main purpose of this paper is to launch a major study on a novel randomly occurring distributedly delayed PSO (RODDPSO) algorithm with applications in healthcare informatics.

Motivated by the above discussions, the purpose of this paper is to propose a RODDPSO-based clustering algorithm with applications on analyzing A&E data. The main contributions of this paper can be summarized in three aspects as follows:

1) A novel RODDPSO algorithm is introduced where the randomly occurring distributed time-delay terms not only contribute to a) a thorough exploration of the entire search space; b) a significant reduction of the possibility of trapping local optima; and c) a proper balance between the local and global search abilities.
2) A hybrid clustering algorithm is proposed which combines the proposed RODDPSO algorithm with the traditional K-means clustering algorithm. The proposed RODDPSO-based clustering algorithm is not dependent on the initial states of the cluster centroids, thereby facilitating a better cluster partition.
3) The proposed RODDPSO-based clustering algorithm is successfully employed to analyze the A&E data in order to verify the triage categorization. With an appropriate triage category (resulting in improved patient routing), the patients' waiting time within A&E departments could be much decreased and patients with serious injury or illness can then be treated with specific care. As such, the efficiency of both human and non-human resource management in A&E departments can be improved.

The remaining part of this paper is organized as follows. The basic PSO algorithm and several well-known variants of PSO algorithms are discussed in Section II. A novel RODDPSO algorithm is introduced in Section III. Detailed information of the RODDPSO-based clustering algorithm is discussed in Section IV. Simulation results of the RODDPSO algorithm and the RODDPSO-based clustering algorithm are presented in Section V and Section VI, respectively. Finally, conclusions and discussions on relevant future work are presented in Section VII.

II. PSO Algorithms

The PSO algorithm is an evolutionary computation algorithm proposed in [10]. Inspired by a metaphor of social interaction, the PSO algorithm is developed to simulate the social behavior of fish schooling or birds flocking, where each particle represents a candidate solution of the research problem.

Note that all the particles move at a certain speed in a D-dimensional search space. The velocity and position of the ith particle at the kth iteration are denoted by two vectors, which are the velocity vector $v_i(k) = (v_{i1}(k), v_{i2}(k), \cdots, v_{iD}(k))$ and the position vector $x_i(k) = (x_{i1}(k), x_{i2}(k), \cdots, x_{iD}(k))$, respectively. According to the swarm intelligence, the position of each particle is automatically updated in the direction of the global optimum, one is the personal best position found by itself (pbest) denoted by $p_i = (p_{i1}, p_{i2}, \cdots, p_{iD})$, and the other one is the global best position throughout the whole swarm (gbest) represented by $p_g = (p_{g1}, p_{g2}, \cdots, p_{gD})$. The velocity and the position of the ith particle at the $(k+1)$th iteration are updated as follows:

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

where k is the current iteration number; w is the inertia weight; c_1 and c_2 are the acceleration coefficients called as cognitive and social parameters, respectively; and r_1 and r_2 are two random numbers which are uniformly distributed over the interval $[0, 1]$.

In the past few years, a variety of improved PSO algorithms have been put forward to enhance the search ability of the PSO algorithm and reduce the possibility of getting trapped in the local optima, see e.g., [34], [36], [48], [51]. For example, as one of the most popular strategies, the PSO algorithm with a linearly decreased inertia weight w (PSO-LDIW) has been

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TEVC.2018.2878536, IEEE Transactions on Evolutionary Computation

proposed in [31], [32], where w is given as follows:

$$w = w_{\max} - (w_{\max} - w_{\min}) \times \frac{iter}{maxiter}, \tag{2}$$

where $w_{\max}$ and $w_{\min}$ represent the maximum and minimum value of the inertia weight, respectively; $iter$ denotes the number of current iteration, and $maxiter$ represents the maximum iteration number. Normally, a larger inertia weight will benefit the global exploration, and a smaller inertia weight will contribute to the local exploitation [32]. Moreover, the PSO algorithm with the constriction factor (PSO-CK) has been introduced in [4] to guarantee the convergence rate and the search ability, where w is set to be 0.729 and $c_1 = c_2 = 1.49$. In addition, the PSO algorithm with time-varying acceleration coefficients (PSO-TVAC) has been proposed in [27]. The cognitive acceleration coefficient c_1 is linearly decreased, and the social acceleration coefficient c_2 is linearly increased, which are shown as follows:

$$c_1 = (c_{1f} - c_{1i}) \times \frac{maxiter - iter}{maxiter} + c_{1i}, \tag{3}$$

$$c_2 = (c_{2f} - c_{2i}) \times \frac{maxiter - iter}{maxiter} + c_{2i}, \tag{4}$$

where c_{1i} and c_{2i} represent the initial values of the acceleration coefficients. c_{1f} and c_{2f} denote the final value of the cognitive acceleration coefficient c_1 and the social acceleration coefficient c_2, respectively. It should be mentioned that the parameters $c_{1i} = 2.5$, $c_{1f} = 0.5$, $c_{2i} = 0.5$, and $c_{2f} = 2.5$ are determined based on experiment experience.

Note that all of above variants of PSO algorithms have mainly focused on adjusting parameters of the PSO algorithms. Furthermore, by developing different topological structures and learning strategies, the search ability of the PSO algorithm can be further enhanced. Along this direction, the adaptive PSO (APSO) algorithm has been proposed in [51] which can automatically adjust the parameters according to the evolutionary factor. In the APSO algorithm, an evolutionary factor has been introduced to identify four evolutionary states, which are the exploration state, the exploitation state, the convergence state, and the jumping-out state. The parameters in the APSO algorithm (e.g. the inertia weight and the acceleration coefficients) are automatically controlled on the basis of the evolutionary state in each iteration. Recently, a switching PSO (SPSO) algorithm has been proposed in [36] to improve the search capability of the APSO algorithm. In the SPSO algorithm, the velocity updating model is switched from one mode to another depending on the evolutionary state predicted by a Markov chain. Furthermore, a switching delayed PSO (SDPSO) algorithm has been introduced in [48] where the delayed information (containing previous personal best and global best particles) has been used to further enhance the searching capability. Moreover, a multimodal delayed PSO (MDPSO) algorithm has been proposed in [34] where the multimodal time-delays (added in the velocity updating model) have helped reduce the possibility of getting trapped in the local optimum and also expand the search space. Nevertheless, there is still much room to further improve the performance of the aforementioned algorithms especially for high-dimensional optimization problems with a large number of local optima.

III. A Novel RODDPSO Algorithm

In this section, a novel RODDPSO algorithm is proposed to further improve the search ability of the traditional PSO algorithm. The main novelty of the proposed RODDPSO lies in the introduction of the randomly occurring distributed time-delays into the velocity updating model. More specifically, a certain number of historical personal best particles and global best particles are randomly selected according to the evolutionary state. Note that the delayed terms are selected by multiplying a random number which is 0 or 1. Compared to the traditional delayed PSO algorithms, the newly introduced randomly occurring distributed time-delays in the velocity updating model make it possible for us to 1) make better use of accumulated history about the population evolution with better accuracy; 2) pursue stronger capability of avoiding local optima trapping problems; and 3) keep an adequate balance between the convergence and the diversity. As such, the proposed RODDPSO could explore and exploit the search space more thoroughly than the traditional PSO algorithm.

A. Framework of the RODDPSO Algorithm

The velocity and position in the novel RODDPSO algorithm are updated as follows:

$$\begin{aligned} v_i(k+1) = {} & w v_i(k) + c_1 r_1 (p_i(k) - x_i(k)) \\ & + c_2 r_2 (p_g(k) - x_i(k)) \\ & + m_l(\xi) c_3 r_3 \sum_{\tau=1}^{N} \alpha_{(\tau)} (p_i(k-\tau) - x_i(k)) \\ & + m_g(\xi) c_4 r_4 \sum_{\tau=1}^{N} \alpha_{(\tau)} (p_g(k-\tau) - x_i(k)), \\ x_i(k+1) = {} & x_i(k) + v_i(k+1), \end{aligned} \tag{5}$$

where k denotes the current iteration number; w is the inertia weight defined in equation (2); acceleration coefficients c_1 and c_2 are updated according to equations (3) and (4), respectively; c_3 and c_4 are the acceleration coefficients for distributed time-delay terms, which are equal to c_1 and c_2, i.e., $c_1 = c_3$ and $c_2 = c_4$; N represents the upper bound of the distributed time-delays; $\alpha_{(\tau)}$ declares a N-dimensional vector where each element is randomly chosen from 0 or 1; $r_i (i = 1, 2, 3, 4)$ are random numbers which are uniformly distributed in $[0, 1]$; $m_l(\xi)$ and $m_g(\xi)$ represent the intensity factors of the distributed time-delay terms according to the evolutionary state ξ.

It is worth mentioning the relationship between the delayed iteration number τ and the current iteration number k. Note that the velocity updating model performs according to (5) when τ is smaller than k, and otherwise we set $\tau = 0$. On the other hand, the selections of the inertia weight and acceleration coefficients are very important in implementing PSO algorithms. The balance of the local and global searching performance is obtained by adjusting the inertia weight. In this paper, the selection of inertia weight adopts the linearly decreasing strategy proposed in [32] with equation (2). Due to the success in improving the search ability of conventional

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TEVC.2018.2878536, IEEE Transactions on Evolutionary Computation

PSO algorithms by employing time-varying acceleration coefficients in [27], we adopt the time-varying strategy to adjust acceleration coefficients with equations (3) and (4).

The flowchart of the novel RODDPSO algorithm is given in Fig. 1.

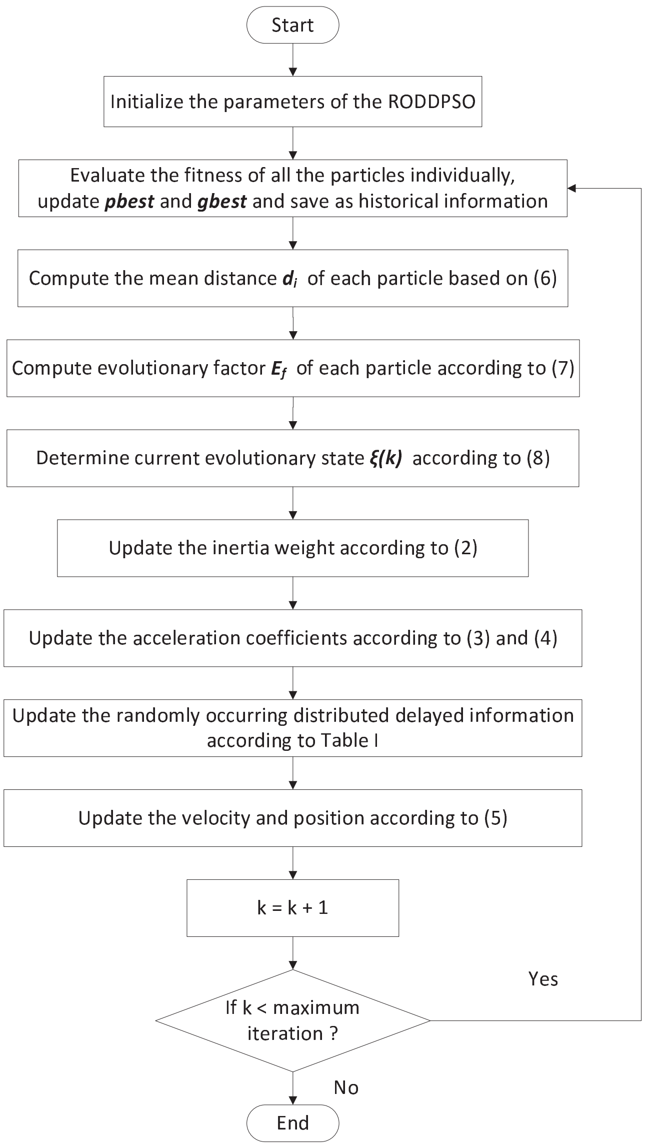


Fig. 1. Flowchart of the RODDPSO algorithm

B. Evolutionary State

In the proposed RODDPSO algorithm, the velocity and position equations are updated according to the evolutionary state depending on the evolutionary factor as mentioned in [36], [51]. The searching characteristics of the PSO algorithm are revealed through the four evolutionary states, i.e., the convergence state, the exploitation state, the exploration state, and the jumping-out state denoted by $\xi(k) = 1$, $\xi(k) = 2$, $\xi(k) = 3$ and $\xi(k) = 4$, respectively.

As mentioned in [51], the evolutionary factor is calculated based on the distance between the particles. The mean distance between the ith particle and other particles denoted by d_i is given as follows:

$$d_i = \frac{1}{S-1}\sum_{j=1,j\neq i}^{S}\sqrt{\sum_{k=1}^{D}(x_{ik}-x_{jk})^2}, \tag{6}$$

where S denotes the swarm size and D represents the dimension of the particle. The evolutionary factor denoted by E_f is shown as follows:

$$E_f = \frac{d_g - d_{\min}}{d_{\max} - d_{\min}}, \tag{7}$$

where d_g represents the global best particle among d_i; $d_{\min}$ and $d_{\max}$ represent the minimum and maximum of d_i in the swarm, respectively.

In this paper, the equal division strategy is employed to classify the four evolutionary states represented by $\xi(k)$ as follows:

$$\xi(k) = \begin{cases} 1, & 0.00 \le E_f < 0.25, \\ 2, & 0.25 \le E_f < 0.50, \\ 3, & 0.50 \le E_f < 0.75, \\ 4, & 0.75 \le E_f \le 1.00. \end{cases} \tag{8}$$

where $\xi(k) = 1, 2, 3, 4$ represent the convergence state, the exploitation state, the exploration state, and the jumping-out state, respectively. Detailed information about the four evolutionary states can be found in the literature [34], [36], [48], [51].

C. Velocity Updating Strategy Based on Randomly Occurring Distributed Time-delay

In this paper, a novel velocity updating strategy with randomly occurring distributed time-delays is demonstrated for four aforementioned evolutionary states as below:

- In the convergence state denoted by $\xi(k) = 1$, the particles are trying to fly into the globally optimal region as soon as possible. Therefore, the velocity updating model in the traditional PSO algorithm is employed, and the distributed time-delay terms are ignored by setting the intensity factor to be zero, i.e., $m_l(\xi) = 0$ and $m_g(\xi) = 0$, respectively.
- In the exploitation state denoted by $\xi(k) = 2$, the particles are supposed to exploit the region around personal best particles. To avoid premature convergence, randomly occurring distributed time-delays are added in the velocity updating model, and a certain number of historical personal best particles are randomly selected for a more thorough search. In this case, the intensity factors are set as $m_l(\xi) = 0.01$ and $m_g(\xi) = 0$.
- In the exploration state denoted by $\xi(k) = 3$, the particles are encouraged to explore the entire search space thoroughly. Hence, randomly occurring distributed time-delays are added in the velocity updating model, and a certain number of historical global best particles are randomly selected with the intensity factors $m_l(\xi) = 0$ and $m_g(\xi) = 0.01$.
- In the jumping-out state denoted by $\xi(k) = 4$, the particles are trying to escape from the region around the local optimum. Therefore, distributed time-delays are added in the velocity updating model where a certain

number of historical personal and global best particles are randomly selected with the intensity factors $m_l(\xi) = 0.01$ and $m_g(\xi) = 0.01$.

The discussion of the above strategy can be summarized in Table I, where the intensity factors $m_l(\xi)$ and $m_g(\xi)$ are determined by the evolutionary states; and k represents the number of current iteration.

TABLE I
VELOCITY UPDATING STRATEGY FOR DISTRIBUTED TIME-DELAYED INFORMATION

State	Mode	$m_l(\xi)$	$m_g(\xi)$
Convergence	$\xi(k) = 1$	0	0
Exploitation	$\xi(k) = 2$	0.01	0
Exploration	$\xi(k) = 3$	0	0.01
Jumping-out	$\xi(k) = 4$	0.01	0.01

IV. A NOVEL RODDPSO-BASED CLUSTERING ALGORITHM

In this section, a novel RODDPSO-based clustering algorithm is devised by employing the proposed RODDPSO algorithm to improve the basic K-means clustering algorithm. The K-means clustering algorithm is a popular clustering algorithm due to its low computation cost and simple implementation. In this paper, the RODDPSO algorithm is used to optimize the cluster centroids where each particle consists of N_c cluster centroids in a single vector. Moreover, the proposed RODDPSO-based clustering algorithm is applied to evaluate the patients' triage category using A&E attendance data.

A. Objective Function

In this paper, the goal of the objective function is to minimize the average distance between the data points to their own centroids, and the objective function is given as follows:

$$J = \frac{\sum_{j=1}^{N_c} \left[\sum_{\forall P_t \in C_{ij}} dist(P_t, M_{ij})/N_p\right]}{N_c} \tag{9}$$

where N_c represents the number of clusters; C_{ij} denotes the jth cluster of the ith particle; M_{ij} represents the jth cluster centroid of the ith particle; P_t denotes the tth data point; $dist(P_t, M_{ij})$ represents the Euclidean distance between the data point P_t and its cluster centroid M_{ij}; N_p represents the number of data points belonging to cluster C_{ij}; and N_c denotes the number of clusters.

B. Framework of the RODDPSO-Based Clustering Algorithm

The RODDPSO algorithm is used to optimize the cluster centroids in order to improve the clustering performance. It is worth mentioning that the powerful search ability of the proposed RODDPSO can reduce the possibility of getting trapped in local optima, and hence improve the clustering performance. The procedure of the proposed RODDPSO-based clustering algorithm is demonstrated as follows:

1) Initialize the parameters including the population size P, the velocity and position of the particles v_i, x_i, acceleration coefficients c_1, c_2, inertia weight w, maximum iteration, the number of clusters N_c, maximum velocity $V_{\max}$ and intensity factors m_l, m_g.
2) Randomly initialize every particle to contain N_c cluster centroids.
3) Calculate the Euclidean distance $dist(P_t, M_{ij})$ between the data point and its cluster centroid.
4) Assign the data points to the closest cluster.
5) Calculate the fitness of all particles based on the objective function (9).
6) Select the personal best particle and the global best particle.
7) Confirm the evolutionary state depending on the calculated evolutionary factor.
8) Update the velocity and position equations based on the evolutionary state according to equation (5).
9) Repeat Steps 3 to 8 till the algorithm reaches the maximal number of iterations.

V. SIMULATION AND DISCUSSION OF THE RODDPSO ALGORITHM

A. Selection of Benchmark Functions

In this paper, eight well-known benchmark functions are employed to evaluate the performance of the proposed RODDPSO algorithm. The benchmark functions are shown by (10) to (17). It should be pointed out that detailed information of the benchmark functions is displayed in Table II including the function number, the function name, the dimension, the search space of each dimension, the threshold, and the minimum of the benchmark functions.

Note that all the benchmark functions are high-dimensional problems. The Sphere function $f_1(x)$ is unimodal and is used to explore the convergence rate of the optimization problem. The Rosenbrock function $f_2(x)$ is a non-convex function which is also known as the Rosenbrock's banana function. The Ackley function $f_3(x)$ and the Rastrigin function $f_4(x)$ are very difficult to optimize because of a large number of local minima. The Schwefel 2.22 function $f_5(x)$ and the Schwefel 1.2 function $f_6(x)$ are classical unimodal and multimodal functions, which are hard to find the optimum. The Griewank function $f_7(x)$ is a popular benchmark function which is widely used to test the convergence of optimization algorithms. The step function $f_8(x)$ is also a typical benchmark function. Here, $x = (x_1, x_2, \cdots, x_D)$ where D is the dimension of the search space. In our simulation, D is taken as 50.

$$\text{Sphere}: f_1(x) = \sum_{i=1}^{D} x_i^2. \tag{10}$$

$$\text{Rosenbrock}: f_2(x) = \sum_{i=1}^{D-1} (100(x_{i+1} - x_i)^2 + (x_i - 1)^2). \tag{11}$$

$$\text{Ackley}: f_3(x) = -20e^{-0.2\sqrt{\frac{1}{D}\sum_{i=1}^{D} x_i^2}} + 20 + e - e^{\frac{1}{D}\sum_{i=1}^{D}\cos 2\pi x_i}. \tag{12}$$

$$\text{Rastrigin}: f_4(x) = \sum_{i=1}^{D} (x_i^2 - 10\cos 2\pi x_i + 10). \tag{13}$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TEVC.2018.2878536, IEEE Transactions on Evolutionary Computation

TABLE II
CONFIGURATION OF BENCHMARK FUNCTIONS

Functions	Name	Dimension	Search space	Threshold	Minimum
$f_1(x)$	Sphere	50	$[-100, 100]$	0.01	0
$f_2(x)$	Rosenbrock	50	$[-30, 30]$	100	0
$f_3(x)$	Ackley	50	$[-32, 32]$	0.01	0
$f_4(x)$	Rastrigin	50	$[-5.12, 5.12]$	50	0
$f_5(x)$	Schwefel 2.22	50	$[-10, 10]$	0.01	0
$f_6(x)$	Schwefel 1.2	50	$[-100, 100]$	0.01	0
$f_7(x)$	Griewank	50	$[-600, 600]$	0.01	0
$f_8(x)$	Step	50	$[-100, 100]$	0	0

$$\text{Schwefel 2.22}: f_5(x) = \sum_{i=1}^{D} | x_i | + \prod_{i=1}^{D} | x_i | . \quad (14)$$

$$\text{Schwefel 1.2}: f_6(x) = \sum_{i=1}^{D} (\sum_{j=1}^{i} x_j)^2. \quad (15)$$

$$\text{Griewank}: f_7(x) = 1 + \frac{1}{4000} \sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n} cos(\frac{x_i}{\sqrt{i}}). \quad (16)$$

$$\text{Step}: f_8(x) = \sum_{i=1}^{D} (\lfloor x_i + 0.5 \rfloor)^2. \quad (17)$$

B. Experiment Results of the RODDPSO Algorithm

As discussed above, eight benchmark functions are employed to evaluate the performance of the introduced RODDPSO algorithm. The superiority of the proposed RODDPSO algorithm is demonstrated over six popular PSO algorithms including the PSO-LDIW [31], PSO-TVAC [27], PSO-CK [4], SPSO [36] SDPSO [48] and MDPSO [34]. The parameters of the experiments are given as follows: the dimension of the search space is $D = 50$, and the population of the swarm is $S = 20$. It should be noted that each experiment has been repeated 20 times independently so as to avoid random influence. The setting of the distributed time-delay τ is determined based on the simulation results. The performance of the RODDPSO algorithm in the 20-dimensional search space with different settings of the upper bound of the distributed time-delay N is shown in Table III. It can be seen that the RODDPSO algorithm demonstrates competitive performance when $N = 100$.

The performance tests for the proposed RODDPSO algorithms are shown in Fig. 2 to Fig. 9. The vertical coordinate represents the logarithmic formation of the mean fitness value of all the tested PSO algorithms, and the horizontal coordinate denotes the number of iteration for Fig. 2 to Fig. 9. Additionally, detailed information of the optimization performance is listed in Table IV, where the mean, the minimum, and the standard deviation of the fitness value with respect to each benchmark function is presented to demonstrate the performance of various PSO algorithms as well as the successful convergence ratio.

It can be seen that the proposed RODDPSO algorithm demonstrates superiority over other PSO algorithms in terms of evaluation indices such as the mean, the minimum, and the standard deviation of the fitness values for function (10) to (17). Specifically, the mean fitness value of the RODDPSO

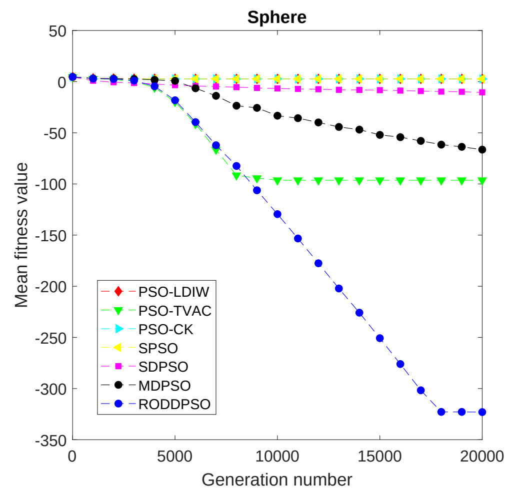


Fig. 2. Performance test for Sphere function $f_1(x)$

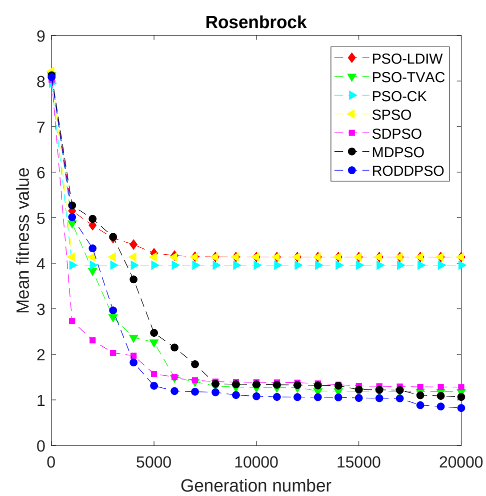


Fig. 3. Performance test for Rosenbrock function $f_2(x)$

algorithm is smaller than that of other PSO algorithms, which demonstrates the superiority of RODDPSO in reaching the global optimum. Moreover, although the RODDPSO algorithm cannot reach the best mean fitness for function (16), it presents competitive performance compared with the PSO-LDIW, PSO-TVAC, PSO-CK and SPSO algorithms. Similarly, the RODDPSO algorithm outperforms the PSO-LDIW, PSO-CK, SPSO, and SDPSO algorithms for function (17) as shown in Fig. 9.

In addition to the mean fitness, the successful convergence

TABLE III
PERFORMANCE EVALUATION OF RODDPSO ALGORITHM WITH DIFFERENT N

		N=25	N=50	N=75	N=100	N=125	N=150	N=175	N=200
$f_1(x)$	Minimum	7.33×10^{-179}	0.0000	0.0000	0.0000	0.0000	8.95×10^{-319}	2.26×10^{-305}	8.94×10^{-281}
	Mean	1.12×10^{-142}	1.48×10^{-323}	4.94×10^{-324}	4.94×10^{-324}	0.0000	1.69×10^{-300}	4.45×10^{-260}	3.52×10^{-236}
	Std. Dev.	4.80×10^{-142}	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	Ratio	100%	100%	100%	100%	100%	100%	100%	100%
$f_2(x)$	Minimum	2.03×10^{-2}	4.19×10^{-3}	2.28×10^{-2}	2.29×10^{-4}	1.16×10^{-3}	2.61×10^{-4}	1.60×10^{-5}	5.14×10^{-6}
	Mean	8.0533	5.1520	1.60×10^{2}	6.4467	1.02×10^{1}	1.45×10^{1}	1.03×10^{1}	7.5699
	Std. Dev.	4.5765	4.4331	6.75×10^{2}	6.2135	1.47×10^{1}	2.01×10^{1}	1.46×10^{1}	4.9529
	Ratio	100%	100%	95%	100%	100%	100%	100%	100%
$f_3(x)$	Minimum	2.66×10^{-15}	2.66×10^{-15}	2.66×10^{-15}	2.66×10^{-15}	2.66×10^{-15}	2.66×10^{-15}	2.66×10^{-15}	2.66×10^{-15}
	Mean	6.04×10^{-15}	5.15×10^{-15}	4.26×10^{-15}	4.80×10^{-15}	5.15×10^{-15}	4.26×10^{-15}	5.68×10^{-15}	4.80×10^{-15}
	Std. Dev.	7.94×10^{-16}	1.67×10^{-15}	1.81×10^{-15}	1.79×10^{-15}	1.67×10^{-15}	1.81×10^{-15}	1.30×10^{-15}	1.79×10^{-15}
	Ratio	100%	100%	100%	100%	100%	100%	100%	100%
$f_4(x)$	Minimum	4.9748	5.9698	6.9647	4.9748	4.9748	4.9748	4.9748	4.9748
	Mean	1.02×10^{1}	1.04×10^{1}	1.05×10^{1}	1.12×10^{1}	1.02×10^{1}	1.06×10^{1}	8.9049	9.8998
	Std. Dev.	2.8345	2.9316	3.0874	3.7350	3.2422	3.8096	2.3384	2.9844
	Ratio	100%	100%	100%	100%	100%	100%	100%	100%
$f_5(x)$	Minimum	5.40×10^{-48}	4.34×10^{-60}	9.26×10^{-76}	3.17×10^{-77}	4.56×10^{-86}	1.28×10^{-86}	2.22×10^{-102}	1.04×10^{-86}
	Mean	1.15×10^{-31}	1.83×10^{-33}	6.83×10^{-44}	1.63×10^{-58}	6.02×10^{-59}	4.28×10^{-52}	4.04×10^{-59}	8.72×10^{-56}
	Std. Dev.	3.43×10^{-31}	7.91×10^{-33}	2.10×10^{-43}	4.94×10^{-58}	2.07×10^{-58}	1.48×10^{-51}	1.75×10^{-58}	3.89×10^{-55}
	Ratio	100%	100%	100%	100%	100%	100%	100%	100%
$f_6(x)$	Minimum	1.47×10^{-32}	1.39×10^{-45}	1.84×10^{-60}	1.78×10^{-66}	2.50×10^{-67}	2.35×10^{-67}	1.18×10^{-62}	9.19×10^{-59}
	Mean	1.23×10^{-22}	2.28×10^{-34}	1.88×10^{-45}	2.37×10^{-55}	2.39×10^{-53}	5.43×10^{-53}	6.71×10^{-52}	1.64×10^{-47}
	Std. Dev.	2.57×10^{-22}	6.83×10^{-34}	8.35×10^{-45}	5.03×10^{-55}	1.07×10^{-52}	2.12×10^{-52}	3.00×10^{-51}	6.23×10^{-47}
	Ratio	100%	100%	100%	100%	100%	100%	100%	100%
$f_7(x)$	Minimum	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	Mean	3.77×10^{-2}	3.10×10^{-2}	4.47×10^{-2}	4.90×10^{-2}	3.32×10^{-2}	1.83×10^{-2}	1.96×10^{-2}	2.86×10^{-2}
	Std. Dev.	3.24×10^{-2}	2.38×10^{-2}	3.47×10^{-2}	4.34×10^{-2}	3.22×10^{-2}	1.96×10^{-2}	2.68×10^{-2}	2.11×10^{-2}
	Ratio	25%	15%	10%	15%	15%	45%	50%	20%
$f_8(x)$	Minimum	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	Mean	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	Std. Dev.	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	Ratio	100%	100%	100%	100%	100%	100%	100%	100%

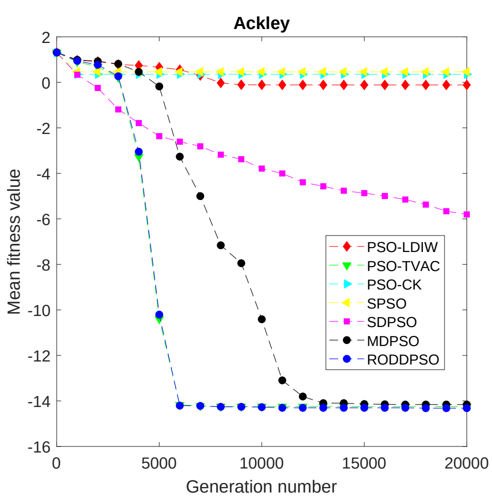


Fig. 4. Performance test for Ackley function $f_3(x)$

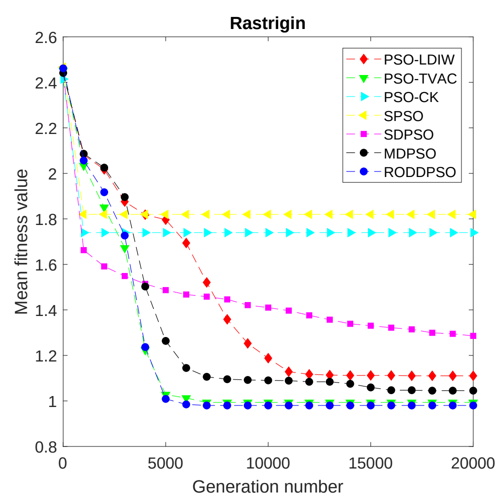


Fig. 5. Performance test for Rastrigin function $f_4(x)$

ratio is a very important index to justify the convergence performance of optimization algorithms. The successful convergence ratio is not always 100% because the testing algorithms cannot always reach the global optimum for all the benchmark functions as shown in Table IV. Note that the RODDPSO algorithm demonstrates competitive performance over other PSO algorithms for function (10) to function (15) and function (17). Note that the Griewank function has a very large number of local minima, therefore, it is difficult to detect the global minimum which leads to a low successful convergence ratio. We can see that all the testing PSO algorithms have low successful convergence ratio for function (16) which are 5%, 25%, 40%, 20%, 30%, 40% and 30%, respectively. Nevertheless, the proposed RODDPSO algorithm can still reach the global minimum with a satisfactory mean fitness value, which demonstrates its competitive performance than other PSO algorithms.

The plots of the convergence rate for testing algorithms are depicted in Fig. 2 to Fig. 9. It is clear that the convergence rate of the RODDPSO algorithm is not as fast as the PSO-TVAC algorithm and the SDPSO algorithm at the beginning for function (10), however, the RODDPSO algorithm reaches

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TEVC.2018.2878536, IEEE Transactions on Evolutionary Computation

TABLE IV
COMPARISONS OF VARIOUS PSO ALGORITHMS ON EIGHT OPTIMIZATION BENCHMARK FUNCTIONS

		PSO-LDIW	PSO-TVAC	PSO-CK	SPSO	SDPSO	MDPSO	RODDPSO
$f_1(x)$	Minimum	1.83×10^{-201}	5.19×10^{-159}	0.0000	6.35×10^{-177}	8.37×10^{-18}	3.59×10^{-102}	0.0000
	Mean	5.00×10^{2}	4.76×10^{-97}	5.00×10^{2}	5.00×10^{2}	3.85×10^{-11}	3.59×10^{-67}	9.88×10^{-324}
	Std. Dev.	2.24×10^{3}	1.76×10^{-96}	2.24×10^{3}	2.24×10^{3}	7.35×10^{-11}	1.60×10^{-66}	0.0000
	Ratio	95%	100%	95%	95%	100%	100%	100%
$f_2(x)$	Minimum	5.38×10^{-3}	1.2375	6.81×10^{-9}	2.70×10^{-6}	9.41×10^{-1}	1.53×10^{-2}	2.43×10^{-2}
	Mean	1.37×10^{4}	1.51×10^{1}	9.07×10^{3}	1.37×10^{4}	1.91×10^{1}	1.16×10^{1}	6.6373
	Std. Dev.	3.29×10^{4}	1.74×10^{1}	2.77×10^{4}	3.29×10^{4}	1.78×10^{1}	1.39×10^{1}	4.7374
	Ratio	75%	100%	75%	75%	100%	100%	100%
$f_3(x)$	Minimum	2.66×10^{-15}	2.66×10^{-15}	6.22×10^{-15}	6.22×10^{-15}	2.93×10^{-8}	6.22×10^{-15}	2.66×10^{-15}
	Mean	7.68×10^{-1}	5.68×10^{-15}	2.2544	2.9002	1.57×10^{-6}	6.93×10^{-15}	4.80×10^{-15}
	Std. Dev.	3.4329	1.30×10^{-15}	3.2030	3.3136	1.91×10^{-6}	2.19×10^{-15}	1.79×10^{-15}
	Ratio	95%	100%	15%	10%	100%	100%	100%
$f_4(x)$	Minimum	4.9748	3.9798	1.89×10^{1}	3.08×10^{1}	2.9850	6.9647	3.9798
	Mean	1.29×10^{1}	9.8501	5.49×10^{1}	6.60×10^{1}	1.93×10^{1}	1.11×10^{1}	9.5516
	Std. Dev.	1.34×10^{1}	4.0435	2.28×10^{1}	2.12×10^{1}	1.13×10^{1}	3.6845	3.0692
	Ratio	95%	100%	50%	25%	100%	100%	100%
$f_5(x)$	Minimum	8.46×10^{-121}	2.67×10^{-32}	5.74×10^{-34}	2.08×10^{-66}	4.22×10^{-9}	1.02×10^{-44}	8.79×10^{-80}
	Mean	1.65×10^{1}	1.01×10^{-19}	6.0000	9.0000	5.00×10^{-1}	1.69×10^{-28}	1.61×10^{-52}
	Std. Dev.	1.18×10^{1}	4.50×10^{-19}	6.8056	1.21×10^{1}	2.2361	7.18×10^{-28}	7.18×10^{-52}
	Ratio	20%	100%	50%	45%	95%	100%	100%
$f_6(x)$	Minimum	5.69×10^{-26}	1.92×10^{-29}	8.30×10^{-103}	5.11×10^{-53}	2.12×10^{-1}	5.30×10^{-28}	6.62×10^{-70}
	Mean	1.00×10^{3}	4.40×10^{-15}	1.25×10^{3}	2.33×10^{3}	1.8968	1.31×10^{-18}	2.42×10^{-48}
	Std. Dev.	2.05×10^{3}	1.97×10^{-14}	2.22×10^{3}	3.88×10^{3}	1.4415	4.23×10^{-18}	1.08×10^{-47}
	Ratio	80%	100%	75%	65%	0%	100%	100%
$f_7(x)$	Minimum	9.86×10^{-3}	0.0000	0.0000	0.0000	2.72×10^{-13}	0.0000	0.0000
	Mean	4.83×10^{-2}	3.22×10^{-2}	5.29×10^{-2}	4.5869	1.88×10^{-2}	2.05×10^{-2}	2.80×10^{-2}
	Std. Dev.	2.79×10^{-2}	3.51×10^{-2}	1.19×10^{-1}	2.03×10^{1}	1.55×10^{-2}	2.43×10^{-2}	2.56×10^{-2}
	Ratio	5%	25%	40%	20%	30%	40%	30%
$f_8(x)$	Minimum	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	Mean	0.0000	0.0000	5.01×10^{2}	6.5500	0.0000	0.0000	0.0000
	Std. Dev.	0.0000	0.0000	2.24×10^{3}	2.70×10^{1}	0.0000	0.0000	0.0000
	Ratio	100%	100%	80%	85%	100%	100%	100%

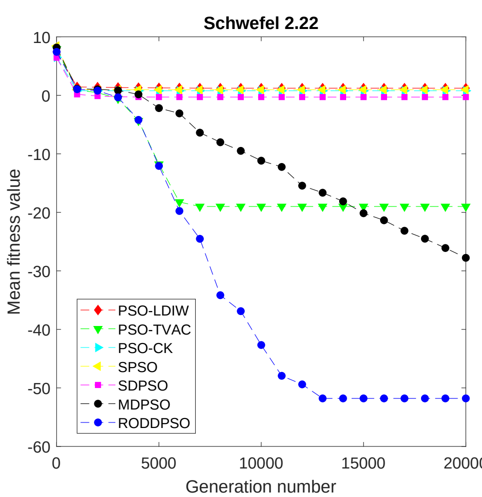


Fig. 6. Performance test for Schwefel 2.22 function $f_5(x)$

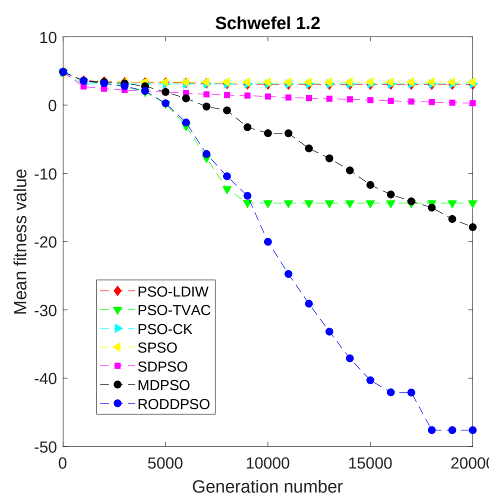


Fig. 7. Performance test for Schwefel 1.2 function $f_6(x)$

the global optimum with better mean fitness value than other PSO algorithms. Moreover, it can be seen that the RODDPSO algorithm tends to reach the global optimum robustly for all the benchmark functions according to the low mean fitness value and high successful convergence ratio. The proposed RODDPSO algorithm outperforms six popular PSO algorithms in both unimodal and multimodal optimization benchmark functions, which indicate that the RODDPSO algorithm is capable of getting rid of local optima. As such, the RODDPSO algorithm can solve the optimization problem with satisfactory convergence speed and convergence accuracy.

It should be mentioned that the RODDPSO-based clustering algorithm is developed for patient clustering from emergency departments. For the purpose of optimising the cluster centroids, the dimension of the search space is 50. In this case, a series of experiments have been conducted to evaluate the effectiveness of the proposed RODDPSO algorithm where the search space is $D = 50$, the upper bound of the distributed

TABLE V
COMPARISONS OF VARIOUS PSO ALGORITHMS IN 50-DIMENSIONAL SEARCH SPACE

		PSO-LDIW	PSO-TVAC	PSO-CK	SPSO	SDPSO	MDPSO	RODDPSO
$f_1(x)$	Minimum	4.03×10^{-51}	1.58×10^{-14}	1.55×10^{-69}	2.63×10^{-140}	1.51×10^{-1}	4.24×10^{-27}	2.35×10^{-118}
	Mean	3.00×10^{3}	7.29×10^{-10}	5.50×10^{3}	2.50×10^{3}	9.3336	2.63×10^{-19}	7.29×10^{-102}
	Std. Dev.	4.70×10^{3}	1.49×10^{-9}	7.59×10^{3}	4.44×10^{3}	9.8519	1.11×10^{-18}	2.87×10^{-101}
	Ratio	70%	100%	60%	75%	0%	100%	100%
$f_2(x)$	Minimum	5.0985	1.65×10^{1}	1.23×10^{-5}	4.6525	1.83×10^{2}	4.8925	4.66×10^{-5}
	Mean	4.76×10^{3}	1.40×10^{2}	4.00×10^{6}	4.91×10^{3}	1.02×10^{3}	5.95×10^{1}	1.91×10^{2}
	Std. Dev.	2.01×10^{4}	1.41×10^{2}	1.79×10^{7}	2.01×10^{4}	9.55×10^{2}	4.41×10^{1}	6.67×10^{2}
	Ratio	70%	50%	90%	60%	0%	85%	95%
$f_3(x)$	Minimum	1.33×10^{-14}	3.85×10^{-6}	4.1669	1.33×10^{-14}	7.44×10^{-1}	1.72×10^{-12}	2.04×10^{-14}
	Mean	5.2116	1.4500	1.11×10^{1}	4.6816	1.8167	1.32×10^{-6}	5.84×10^{-1}
	Std. Dev.	6.6150	2.6486	4.3265	4.8179	5.48×10^{-1}	5.74×10^{-6}	7.42×10^{-1}
	Ratio	60%	35%	0%	10%	0%	100%	60%
$f_4(x)$	Minimum	4.08×10^{1}	5.27×10^{1}	1.38×10^{2}	1.07×10^{2}	8.98×10^{1}	3.48×10^{1}	4.68×10^{1}
	Mean	1.21×10^{2}	8.86×10^{1}	2.23×10^{2}	1.76×10^{2}	1.38×10^{2}	7.62×10^{1}	7.45×10^{1}
	Std. Dev.	5.57×10^{1}	2.18×10^{1}	4.28×10^{1}	3.19×10^{1}	3.93×10^{1}	2.10×10^{1}	1.45×10^{1}
	Ratio	5%	0%	0%	0%	0%	10%	5%
$f_5(x)$	Minimum	2.00×10^{1}	2.34×10^{-6}	2.76×10^{-2}	1.97×10^{-60}	2.23×10^{-1}	1.28×10^{-12}	3.12×10^{-17}
	Mean	5.85×10^{1}	2.5012	2.86×10^{1}	3.70×10^{1}	2.49×10^{1}	1.0000	1.0000
	Std. Dev.	2.54×10^{1}	4.4421	1.85×10^{1}	1.56×10^{1}	2.14×10^{1}	3.0779	3.0779
	Ratio	0%	70%	0%	10%	0%	90%	90%
$f_6(x)$	Minimum	5.01×10^{3}	3.2281	6.59×10^{-11}	1.58×10^{-4}	2.84×10^{3}	3.00×10^{-1}	4.49×10^{-4}
	Mean	2.58×10^{4}	3.94×10^{3}	1.22×10^{4}	1.67×10^{4}	1.30×10^{4}	1.11×10^{3}	5.01×10^{2}
	Std. Dev.	1.41×10^{4}	3.80×10^{3}	1.01×10^{4}	1.92×10^{4}	7.15×10^{3}	2.30×10^{3}	1.54×10^{3}
	Ratio	0%	0%	15%	25%	0%	0%	30%
$f_7(x)$	Minimum	0.0000	1.76×10^{-12}	1.01×10^{-14}	0.0000	3.92×10^{-1}	0.0000	1.11×10^{-16}
	Mean	4.52×10^{1}	2.37×10^{-2}	4.54×10^{1}	9.1116	8.81×10^{-1}	3.30×10^{-2}	2.13×10^{-2}
	Std. Dev.	6.22×10^{1}	2.96×10^{-2}	6.88×10^{1}	2.78×10^{1}	2.28×10^{-1}	3.43×10^{-2}	2.80×10^{-2}
	Ratio	35%	50%	15%	40%	0%	35%	45%
$f_8(x)$	Minimum	0.0000	0.0000	1.30×10^{1}	0.0000	6.0000	0.0000	0.0000
	Mean	1.50×10^{3}	2.50×10^{-1}	7.38×10^{3}	2.01×10^{3}	1.41×10^{1}	0.0000	2.50×10^{-1}
	Std. Dev.	3.66×10^{3}	5.50×10^{-1}	6.59×10^{3}	4.10×10^{3}	6.2146	0.0000	4.44×10^{-1}
	Ratio	85%	80%	0%	15%	0%	100%	75%

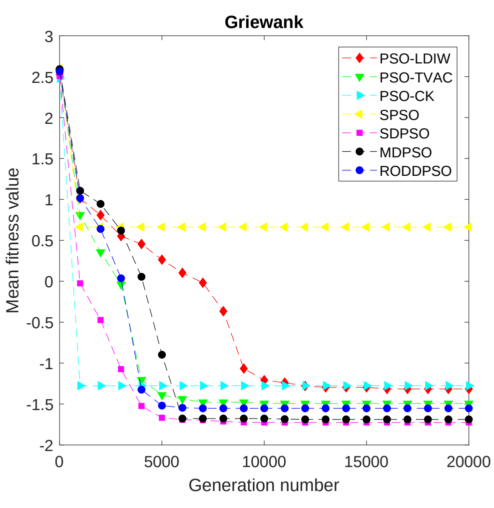


Fig. 8. Performance test for Griewank function $f_7(x)$

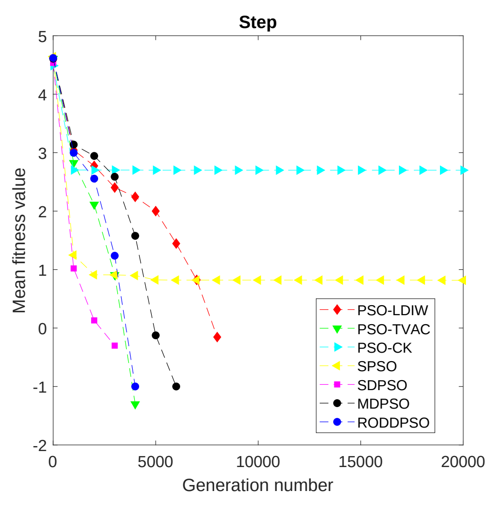


Fig. 9. Performance test for Step function $f_8(x)$

time-delay is $N = 110$, and other parameter settings remain the same. The corresponding experiment results are displayed in Table V. We can see that the successful convergence ratio of the RODDPSO algorithm is satisfactory. Moreover, the proposed RODDPSO algorithm demonstrates competitive performance over other PSO algorithms in terms of the mean, the minimum, and the standard deviation of the fitness value via the selected benchmark functions. Therefore, the proposed RODDPSO algorithm exhibits satisfactory performance on the convergence, accuracy and the diversity in the 50-dimensional search space, which indicates that the reliability of the RODDPSO-based clustering algorithm.

VI. RESULTS AND ANALYSIS OF THE RODDPSO BASED CLUSTERING ALGORITHM

In A&E departments, an obvious challenge is that patients requiring urgent treatment can go straight to the A&E at any time, thereby causing substantial strain on limited medical

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TEVC.2018.2878536, IEEE Transactions on Evolutionary Computation

resources. Hence, the number of emergency cases increases rapidly in recent years which leads to overcrowding in many A&E departments. In response to the revolution of data mining and machine learning techniques, it becomes more and more convenient for A&E staff to manage medical resources and arrange work schedules, thereby meeting the 4-hour requirement in emergency departments [3]. For instance, computer simulation models have been widely used for simulating real-world systems. Mathematical models have been introduced in [5], [9] to simulate the patient flow of emergency departments. A discrete-event simulation model has been introduced in [21] to simulate the patient flows in A&E departments, and multi-objective optimization analysis has been conducted for bed management. The relationship between ambient air pollution and patients' attendance at emergency departments has been studied in [17].

Moreover, overcrowding in A&E departments brings many adverse effects such as lower treatment quality, increased working burden and increased patient waiting time. Notably, an efficient and accurate identification of patients' severity is of vital importance to improve the efficiency of medical treatment and relieve the burden on the human and medical resources. Consequently, it is of significance to investigate a proper triage category of the patients. Importantly, an appropriate triage category enables patients with serious illness or injury to be treated. Non-emergency cases can also be re-routed to other services in the health system. In addition, the management of medical resources can be allocated in an appropriate manner so as to reduce the financial cost. As such, the generation of an accurate triage category is important for A&E departments.

In this section, the clustering performance of the introduced RODDPSO-based clustering algorithm is evaluated by adopting the silhouette clustering validation method. The triage category is defined to include 5 groups in [24] which are immediate resuscitation, very urgent, urgent, standard and non-urgent. Therefore, the number of clusters is five, and the clustering performance is evaluated by comparing the silhouette coefficients obtained by the K-means clustering algorithm and the FCM clustering algorithm with the RODDPSO-based clustering algorithms.

A. Data Pre-processing

The data is provided by a hospital in West London including the urgent care center, the minor injury unit and the A&E department. The overall number of patient attendances at the emergency departments is $126,986$ over the period examined. Patient attendances at the A&E department, the urgent care center and the minor injury unit are $51,713$, $15,151$ and $60,122$, respectively.

Each record represents an incident in a single row and each column indicates an attribute with respect to the patient. Note that there are totally 25 attributes in the data consisting of the pseudo NHS number, general practitioner (GP) practice code, patient age, arrival time, departure time, provider code, provider name, date time for treatment, fiscal year label, arrival month, arrival date, modal of arrival, mode of arrival description, attendance disposal, attendance disposal description, core healthcare resource group (HRG), HRG description, referral source, referral source description, A&E department description, clinical commissioning groups (CCGs), first diagnosis, diagnosis description, and postcode sector of usual address.

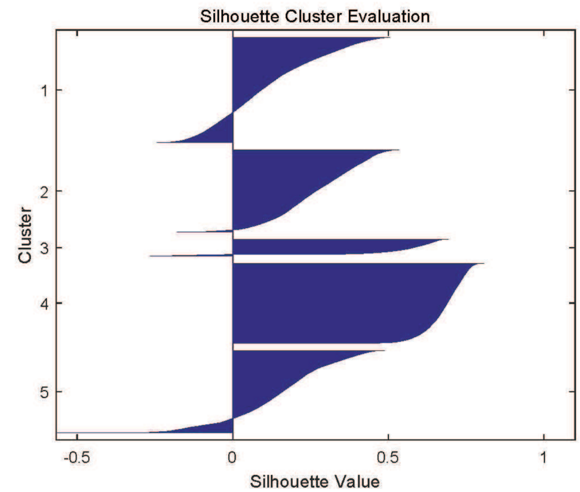


Fig. 10. Silhouette coefficient of K-means clustering algorithm

The data is recorded in real-time, especially the arrival date time, conclusion date time and date time seen for treatment. Hence, we compute the time interval of treatment time and waiting time in A&E departments for later analysis. Moreover, the computation cost is effectively reduced by normalizing the data. It should be mentioned that the data includes missing values and redundant information. Hence, $3,778$ incidents are deleted because their treatment date time is null or missing. Furthermore, redundant information is also removed, e.g., healthcare resource group (HRG) and HRG description, where the latter only represents the description of previous attribute. In addition, the irrelevant attributes such as the provider code and the GP practice code are also abandoned by employing statistical analysis.

B. Experiments Results of the RODDPSO Based Clustering Algorithm

Silhouette is a popular cluster validation method proposed in [28]. To evaluate the clustering performance of the proposed RODDPSO-based clustering algorithm, we compare the silhouette coefficients of the RODDPSO-based clustering algorithm with the K-means and FCM clustering algorithms. In this paper, the squared Euclidean (sqeuclidean) distance metric is adopted due to its simple implementation. The MATLAB plots of the silhouette coefficients of the K-means, the FCM and the RODDPSO-based clustering algorithms are depicted in Fig. 10, Fig. 11 and Fig. 12, respectively.

The mean silhouette coefficients of the K-means, the FCM and the RODDPSO-based clustering algorithms are 0.2970, 0.1253 and 0.3166, respectively. We can see that in Fig. 10, most of the silhouette values of the K-means clustering algorithm are positive, which indicates that most of the data points are assigned to the proper clusters. In Fig. 11, more than half of the data points obtain negative values of the silhouette

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TEVC.2018.2878536, IEEE Transactions on Evolutionary Computation

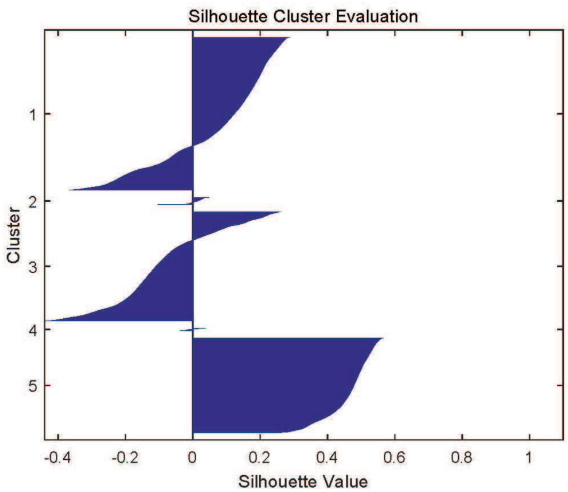


Fig. 11. Silhouette coefficient of Fuzzy C-means clustering algorithm

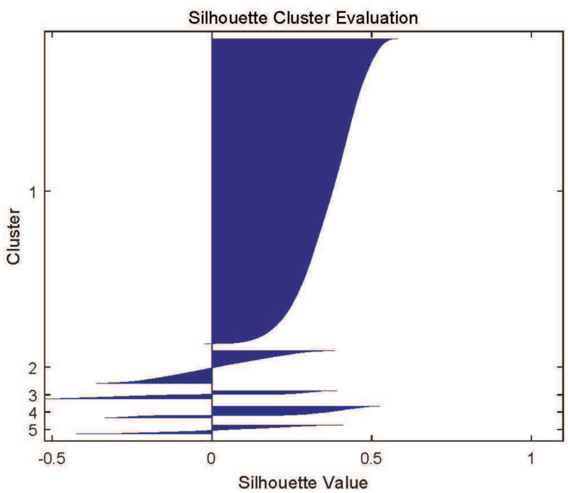


Fig. 12. Silhouette coefficient of RODDPSO-based clustering algorithm

coefficients, and the mean silhouette coefficient is much smaller than that of the K-means and the RODDPSO-based clustering algorithms. As such, the clustering performance of FCM algorithm is not satisfactory. It has been shown in Fig. 12 that the mean silhouette value of the RODDPSO-based clustering algorithm is 0.3166 which is higher than the results of the K-means and the FCM clustering techniques. Furthermore, it is clear that there are fewer negative silhouette values using the RODDPSO-based clustering algorithm than the K-means and the FCM clustering techniques, which indicate fewer data points are assigned to the inappropriate clusters. Thus, the superiority and feasibility of the proposed RODDPSO-based clustering algorithm is demonstrated and the generated triage category is reasonable.

VII. Conclusion

In this paper, a novel RODDPSO algorithm is proposed and successfully applied to improve the standard K-means clustering algorithm on A&E attendance data. The velocity updating model of the RODDPSO algorithm is adaptively adjusted depending on the evolutionary state. It is worth mentioning that the distributed time-delay terms containing historical information of previous personal and global best particles are added in the velocity updating model. As such, the RODDPSO algorithm is capable of escaping from local optima, and the search space is explored and exploited more thoroughly than the classic PSO algorithm. The superiority of the proposed RODDPSO algorithm is demonstrated over six well-known PSO algorithms on eight popular benchmark functions including both unimodal and multimodal cases. Finally, the novel RODDPSO algorithm has been successfully employed to improve the standard K-means clustering algorithm on A&E attendance data. The effectiveness of the proposed RODDPSO-based clustering algorithm is demonstrated by comparing the mean silhouette value with the K-means and FCM clustering algorithms. Future work can be summarized into three aspects: (1) how to further improve the convergence speed of the proposed RODDPSO algorithm; (2) how to apply the proposed RODDPSO algorithm to other complex systems such as deep neural networks [49], [50], genetic regulatory networks [42], and telecommunication systems [15], [45], [46]; and (3) how to extend our results to other data mining problems in A&E departments and the wider health system [43].

References

[1] Centre for Change and Innovation, "Decide to admit v admit to decide - working together to improve unscheduled care in NHS scotland," *National Conference Unplanned Care for Medical Conditions*, Feb. 2004.

[2] M. Alswaitti, M. Albughdadi, and N. A. M. Isa, "Density-based particle swarm optimization algorithm for data clustering," *Expert Systems with Applications*, vol. 91, pp. 170-186, Jan. 2018.

[3] P. Bhattacharjee and P. K. Ray, "Patient flow modelling and performance analysis of healthcare delivery processes in hospitals: a review and reflections," *Computers and Industrial Engineering*, vol. 78, pp. 299-312, Dec. 2014.

[4] 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.

[5] T. J. Coats and S. Michalis, "Mathematical modelling of patient flow through an accident and emergency department," *Emergency Medicine Journal*, vol. 18, no. 3, pp. 190-192, May. 2001.

[6] S. Das, A. Abraham, and A. Konar, "Automatic kernel clustering with a multi-elitist particle swarm optimization algorithm," *Pattern Recognition Letters*, vol. 29, no. 5, pp. 688-699, Apr. 2008.

[7] I. S. Dhillon and D. S. Modha, "Concept decompositions for large sparse text data using clustering," *Machine Learning*, vol. 42, no. 1, pp. 143-175, Jan. 2001.

[8] K. A. De Jong, *Evolutionary Computation: A Unified Approach*, Cambridge, MA: MIT Press, 2006.

[9] J. Eatock, M. Clarke, C. Picton, and T. Young, "Meeting the four-hour deadline in an A&E department," *Journal of Health Organization and Management*, vol. 25, no. 6, pp. 606-624, 2011.

[10] 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.

[11] 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.

[12] 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.

[13] W. A. Feess and S. G. Stephens, "Evaluation of GPS ionospheric time-delay model," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 3, pp. 332-338, May. 1987.

[14] B. J. Frey and D. Dueck, "Clustering by passing messages between data points," *Science*, vol. 315, no. 5814, pp. 972-976, Jan. 2007.

[15] H. Geng, Y. Liang, Y. Liu, and F. E. Alsaadi, "Bias estimation for multi-rate sensor fusion with unknown inputs," *Information Fusion*, vol. 39, pp. 139-153, Jan. 2018.

[16] M. Gong, Q. Cai, X. Chen, and L. Ma, "Complex network clustering by multiobjective discrete particle swarm optimization based on decomposition," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 1, pp. 82-97, Feb. 2014.

[17] P. Guo, W. Feng, M. Zheng, J. Lv, L. Wang, J. Liu, Y. Zhang, G. Luo, Y. Zhang, C. Deng, and T. Shi, "Short-term associations of ambient air pollution and cause-specific emergency department visits in Guangzhou, China," *Science of the Total Environment*, vol. 613, pp. 306-313, Sept. 2018.

[18] H. Izakian and A. Abraham, "Fuzzy C-means and fuzzy swarm for fuzzy clustering problem," *Expert Systems with Applications*, vol. 38, no. 3, pp. 1835-1838, Mar. 2011.

[19] A. K. Jain, "Data clustering: 50 years beyond K-means," *Pattern Recognition Letters*, vol. 31, no. 8, pp. 651-666, 2010.

[20] K. Krishna and M. N. Murty, "Genetic K-means algorithm," *IEEE Transactions on Systems, Man, and Cybernetics–Part B: Cybernetics*, vol. 29, no. 3, pp. 433-439, Jun. 1999.

[21] P. Landa, M. Sonnessa, E. Tanfani, and A. Testi, "Multiobjective bed management considering emergency and elective patient flows," *International Transactions in Operational Research*, vol. 25, no. 1, pp. 91-110, Jan. 2018.

[22] M. Li, S. Yang, and X. Liu, "A performance comparison indicator for Pareto front approximations in many-objective optimization," In: *Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation*, Madrid, Spain, Jul. 2015, pp. 703-710.

[23] 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.

[24] J. Marrow, "Triage and casemix in accident and emergency medicine," *European Journal of Emergency Medicine*, vol. 5, no. 1, pp. 53-58, Mar. 1998.

[25] T. Niknam and B. Amiri, "An efficient hybrid approach based on PSO, ACO and K-means for cluster analysis," *Applied Soft Computing*, vol. 10, no. 1, pp. 183-197, Jan. 2010.

[26] C. C. Queirolo, L. Silva, O. R. Bellon, and M. P. Segundo, "3D face recognition using simulated annealing and the surface interpenetration measure," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 2, pp. 206-219, Feb. 2010.

[27] 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.

[28] P. J. Rousseeuw, "Silhouettes: a graphical aid to the interpretation and validation of cluster analysis," *Journal of Computational and Applied Mathematics*, vol. 20, pp. 53-65, Nov. 1987.

[29] S. Z. Selim and K. Alsultan, "A simulated annealing algorithm for the clustering problem," *Pattern Recognition*, vol. 24, no. 10, pp. 1003-1008, Mar. 1991.

[30] M. Steinbach, G. Karypis, and V. Kumar, "A comparison of document clustering techniques," In: *Proceedings of the KDD Workshop on Text Mining*, Boston, USA, Aug. 2000, vol. 400, no. 1, pp. 525-526.

[31] 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.

[32] 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.

[33] Q. Song and Z. Wang, "Neural networks with discrete and distributed time-varying delays: a general stability analysis," *Chaos, Solitons and Fractals*, vol. 37, no. 5, pp. 1538-1547, Sept. 2008.

[34] 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.

[35] P. N. Tan, M. Steinbach, and V. Kumar, *Introduction to Data Mining*, Boston: Addison Wesley, 2005.

[36] 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.

[37] R. W. Tothill, A. V. Tinker, J. George, R. Brown, S. B. Fox, S. Lade, D. S. Johnson, M. K. Trivett, D. Etemadmoghadam, B. Locandro, and N. Traficante, "Novel molecular subtypes of serous and endometrioid ovarian cancer linked to clinical outcome," *Clinical Cancer Research*, vol. 14, no. 16, pp. 5198-5208, Aug. 2008.

[38] D. W. Van der Merwe and A. P. Engelbrecht, "Data clustering using particle swarm optimization," In: *Proceedings of the IEEE Congress on Evolutionary Computation*, Canberra, Australia, Dec. 2003, vol. 1, pp. 215-220.

[39] F. Van den Bergh and A. P. Engelbrecht, "A cooperative approach to particle swarm optimization," *IEEE Transactions on Evolutionary Computation*, vol. 8, no. 3, pp. 225-239, Jun. 2004.

[40] 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.

[41] J. Vissers and R. Beech, *Health Operations Management: Patient Flow Logistics in Health Care*, London: Routledge, 2005.

[42] 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.

[43] H. Wang, Y. Jin, and J. O. Jansen, "Data-driven surrogate-assisted multiobjective evolutionary optimization of a trauma system," *IEEE Transactions on Evolutionary Computation*, vol. 20, no. 6, pp. 939-952, Apr. 2016.

[44] Z. Wang, Y. Liu, and X. Liu, "On global asymptotic stability of neural networks with discrete and distributed delays," *Physics Letters A*, vol. 345, no. 4, pp. 299-308, Oct. 2005.

[45] Z. Wang, L. Wang, S. Liu, and G. Wei, "Encoding-decoding-based control and filtering of networked systems: insights, developments and opportunities," *IEEE/CAA Journal of Automatica Sinica*, vol. 5, no. 1, pp. 3-18, Jan. 2018.

[46] L. Wang, Z. Wang, Q.-L. Han, and G. Wei, "Synchronization control for a class of discrete-time dynamical networks with packet dropouts: a coding-decoding-based approach," *IEEE Transactions on Cybernetics*, vol. 48, no. 8, pp. 2437-2448, Aug. 2018.

[47] X. Xiao, E. R. Dow, R. Eberhart, Z. B. Miled, and R. J. Oppelt, "Gene clustering using self-organizing maps and particle swarm optimization," In: *Proceedings of the International Parallel and Distributed Processing Symposium*, Nice, France, April. 2003.

[48] 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.

[49] N. Zeng, Z. Wang, H. Zhang, W. Liu, and F. E. Alsaadi, "Deep belief networks for quantitative analysis of gold immunochromatographic strip," *Cognitive Computation*, vol. 8, no. 4, pp. 684-692, Aug. 2016.

[50] N. Zeng, H. Zhang, B. Song, W. Liu, Y. Li, and A. M. Dobaie, "Facial expression recognition via learning deep sparse autoencoders," *Neurocomputing*, vol. 273, pp. 643-649, Jan. 2018.

[51] Z. Zhan, J. Zhang, Y. Li, and 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.

[52] C. Zhang, D. Ouyang, and J. Ning, "An artificial bee colony approach for clustering," *Expert Systems with Applications*, vol. 37, no. 7, pp. 4761-4767, Jul. 2010.

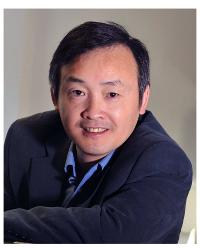
Weibo Liu received his B.S. degree in electrical engineering from the Department of Electrical Engineering & Electronics, University of Liverpool, Liverpool, UK, in 2015. He is currently pursuing the Ph.D. degree in Computer Science at Brunel University London, Uxbridge, UK. His research interests include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

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TEVC.2018.2878536, IEEE Transactions on Evolutionary Computation

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.



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, Edinburg, 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.

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.

David Bell received a BSc (Hons) degree in Computer Science with Electronics from Swansea University of the UK in 1988, the MBA degree in Business Administration from Warwick Business School of the UK in 2003, and the PhD degree in Computing from Brunel University London of the UK in 2006.

Dr. Bell is currently a Senior Lecturer in Computer Science at Brunel University London. He has authored over 80 papers in refereed international conferences and journals. His current research interests include hybrid simulation, healthcare systems, data-driven modelling and simulation. He is an investigator on a number of funded research projects including the simulation of healthcare devices & systems, cloud computing services and the migration of displaced individuals using high performance computing. Prior to returning to academia, he worked in the IT industry, latterly as Technology Directory at a global investment bank. He is the Education Chair of the ACM Special Interest group on Simulation and Modelling (ACM SIGSIM) and reviewer for several international journals.