

# An adaptive particle swarm optimization method based on clustering

Xiaolei Liang · Wenfeng Li · Yu Zhang · MengChu Zhou

Published online: 17 April 2014  
© Springer-Verlag Berlin Heidelberg 2014

**Abstract** Particle swarm optimization (PSO) is an effective method for solving a wide range of problems. However, the most existing PSO algorithms easily trap into local optima when solving complex multimodal function optimization problems. This paper presents a variation, called adaptive PSO based on clustering (APSO-C), by considering the population topology and individual behavior control together to balance local and global search in an optimization process. APSO-C has two steps. First, via a K-means clustering operation, it divides the swarm dynamically in the whole process to construct variable subpopulation clusters and after that adopts a ring neighborhood topology for information sharing among these clusters. Then, an adaption mechanism is proposed to adjust the inertia weight of all individuals based on the evaluation results of the states of clusters and the swarm, thereby giving the individual suitable search power. The experimental results of fourteen benchmark functions show that APSO-C has better performance in the terms of convergence speed, solution accuracy and algorithm reliability than several other PSO algorithms.

Communicated by W. Pedrycz.

X. Liang · W. Li (✉) · Y. Zhang · M. Zhou  
School of Logistics Engineering, Wuhan University of Technology,  
Wuhan 430063, Hubei, China  
e-mail: liwf@whut.edu.cn

X. Liang  
e-mail: liangxiaolei@whut.edu.cn

Y. Zhang  
e-mail: sanli@whut.edu.cn

M. Zhou  
Department of Electrical and Computer Engineering, New Jersey  
Institute of Technology, Newark, NJ 07102-1982, USA  
e-mail: zhou@njit.edu

**Keywords** Particle swarm optimization (PSO) · Function optimization · Dynamic topology · Cluster evaluation · Adaptive particle swarm optimization

## 1 Introduction

A particle swarm optimization (PSO) algorithm is a kind of Swarm Intelligence (SI) technology, which was initially proposed (Eberhart and Kennedy 1995; Kennedy and Eberhart 1995) in 1995. Inspired by birds flocking, the core concept of PSO is to find out the optima or sub-optima of an objective function through the co-operation and information sharing among particles. Since PSO is efficient, simple and robust, it has been widely used in multi-objective optimization (Leong and Yen 2008), artificial neural network training (Grimaldi et al. 2004), image segmentation (Chander et al. 2011), vehicle routing problems (Khouadjia et al. 2012), job-shop scheduling problems (Ge et al. 2008), optimal power flow (Kang et al. 2012a, b) and other fields. However, as other evolutionary algorithms, how to balance between exploration and exploitation is still a key and unresolved problem for PSO. At the beginning of exploration, all the particles can enjoy fast moving and gather around the best positions found so far. At the late stage, the whole swarm becomes dense near several points and particles cannot escape from these local extreme areas. If the shortcoming can be overcome, PSO will be more powerful and thus applicable to more problems.

Most researchers have focused on the communication topology of a swarm or behavior of individuals separately and obtained some impressive results (Banks et al. 2007). From them the swarm benefits little from static single population structure. On one hand, a dynamic topology is a good choice. A simple and efficient dynamic neighborhood topology would be beneficial to improve the information shar-

ing and bio-diversity in a swarm. On the other hand, the behavior of individuals determines their search ability. Some researchers have proposed effective adaptive strategies, such as life time (Lanzarini et al. 2006), multiple stage learning (Brits et al. 2002) and quantum behavior (Sun et al. 2011; Li et al. 2012), to improve PSO. However, the population topology is closely associated with individual behavior. The intelligence of individuals is from learning from the environment. Hence, the swarm topology and individual behavior should be considered together comprehensively.

This work proposes a new approach based on the topological structures and self-adaptive control of individuals, called adaptive PSO based on clustering (APSO-C). The proposed algorithm takes a clustering method to divide the swarm into several subpopulations dynamically. These subpopulations share their information from the solution space with a ring topology. Compared with the topology of gbest and lbest versions (Kennedy and Eberhart 1995; Kennedy 1997), it shortens the information transmission distance and avoids excessive concentration of particles. Then a method is used to evaluate the performance of each cluster. Based on the evaluation results, each individual adjusts its inertia weight adaptively to balance the local and global searches. These operations are carried out automatically after individuals' renewal is completed. Thus each subpopulation can obtain different components at each generation. It yields positive effect on the diversity and activity of the swarm. Clustering, adaption and new learning mechanism together provide a quite different solution to improve PSO. Tests are performed to verify that the occurrence of prematurity is reduced and the search ability of the swarm is enhanced via APSO-C.

The rest of the paper is organized as follows. Section 2 reviews PSO and its developments. APSO-C is presented in Sect. 3 including the dynamic population decomposition and adaptation strategies. Section 4 experimentally compares APSO-C with various existing PSO algorithms taken from the literature through benchmark functions. Finally, conclusions are drawn in Sect. 5.

## 2 Background and related work

### 2.1 Particle swarm optimization

In the classic PSO, a group of particles are set in a multi-dimensional space and they can fly freely to search the best position. Each individual can be treated as a point in the search space and characterized by its position and velocity. Suppose that the search space is  $D$ -dimensional, then the position of the  $i$ -th particle can be represented by a  $D$ -dimensional vector  $\mathbf{x}_i = (x_{i1}, x_{i2}, \dots, x_{iD})$ , and the velocity by another vector  $\mathbf{v}_i = (v_{i1}, v_{i2}, \dots, v_{iD})$ . The best previously visited position of the  $i$ -th particle is

denoted as  $\mathbf{pbest}_i = (p_{i1}, p_{i2}, \dots, p_{iD})$ . Define  $\mathbf{gbest} = (g_1, g_2, \dots, g_D)$  as the best global position that the swarm has found so far. The velocity and position equations are given as follows:

$$v_{id}^{t+1} = v_{id}^t + c_1 r_1 (p_{id}^t - x_{id}^t) + c_2 r_2 (g_d^t - x_{id}^t) \quad (1)$$

$$x_{id}^{t+1} = x_{id}^t + v_{id}^{t+1} \quad (2)$$

where  $\forall i \in \mathbb{N}_N, \mathbb{N}_N = \{1, 2, \dots, N\}$ ,  $N$  is the size of the swarm;  $d$  is the index of the co-ordinate being updated;  $c_1$  and  $c_2$  are positive constants, called acceleration constants;  $r_1$  and  $r_2$  represent random numbers, following the uniform distribution over  $[0, 1]$ .

### 2.2 Existing particle swarm optimization

Since PSO was first produced, many developed versions have been presented in literature. We can classify them from two aspects: population topology and individual behavior control as briefly reviewed next.

#### 2.2.1 Population structure and topology

A population structure is the foundation of a swarm. Different structures may drive the swarm to behave differently. In the early research, the structure (Kennedy 1999; Eberhart and Kennedy 1995) is single in common, implying that all particles are in the same microenvironment and under the same control. In this model it is effective for particles to interact with others and the groups have a rapid speed of the convergence. Under it, Kennedy (1997), and Shi and Eberhart (1998a) analyze the effect of parameters on the PSO performance. Based on the evaluation of swarm distribution and particle fitness, an evaluation method for a swarm state is proposed in Zhan et al. (2009). It defines four different population evolutionary states: exploration, exploitation, convergence and jumping. The automatic control of parameters and execution of an elitist learning strategy are performed at runtime to improve the search efficiency and convergence speed. Through the analysis of particle behavior in a search process, dynamic population size strategies are used. In a kind of variable population size PSO (Lanzarini et al. 2006), life time and neighborhood are considered to allow the size of the population to vary. A variant (Montes de Oca et al. 2011a, b) adopts incremental social learning to realize time-varying population size. Particles are introduced to swarm gradually. Based on a kind of variable neighborhood search strategy and a path relinking strategy, Marinakis and Marinaki (2013) propose a new PSO model with expanding neighborhood. It starts from a small size neighborhood and increases the size of the neighborhood in each iteration. Chen and Zhao (2009) propose an adaptive variable population size by periodic partial

increasing or declining of individuals in the form of a ladder function. Like a living body, a particle has its life time.

Inspired from a biological society's structure, researchers have proposed varying population and multiple subpopulations to improve the performance of PSO. With the conception that a group means an environment for individuals to learn in swarm intelligence, Jie et al. (2008) propose knowledge-based co-operative PSO. In it, a learning mechanism based on billboard is established and three different states are adopted to measure sub-populations that contain growing, near mature or mature statuses. It is common that communities exist in a society. Kennedy (2000), Li (2004), and Passaro and Starita (2006) take a clustering operation to divide the society into several subpopulations. Madeiro et al. (2009) apply adaptive density-based clustering algorithm to create neighbors for particles. Competition is the main evolutionary power for biology to avoid extinction. Brits et al. (2002) mix the niching technology and PSO to develop a niching particle swarm optimizer. In each niche, a particle searches its local area based on its personal experience only. Subpopulations are then merged when they intersect. This model simulates the evolutionary process of a biocommunity. Similarly, a PSO with a simple yet effective niching algorithm is developed to improve the performance (Li 2010). GAPSO (Liu et al. 2012) is based on a greedy algorithm and niche structure. A ring topology is employed in each niche instead of a fully connected topology in it. Recently, a Gaussian classifier-based evolutionary strategy (Dong and Zhou 2014) is proposed to solve multimodal optimization problems under a multi-subpopulation structure in evolutionary algorithms (EAs). It employs Gaussian models to capture the landscape shapes of objective functions and a zoom factor to accelerate the search speed. Also, it can be applied in PSOs to guide subpopulations.

A topology is the description of information interaction patterns among individuals. Particles co-operate or compete with others through information exchanges to evolve the whole swarm. Some researchers (Passaro and Starita 2006; Mendes et al. 2003; Kennedy and Eberhart 2002) have proved that a topology is seriously important for PSO performance. The gbest and lbest topology are the classic examples of a static topology. In gbest, the best neighbor in the entire population influences the target particle and is a fully connected group. While, in lbest, each particle is only connected to several other members in its neighbor. Kennedy (1999) adopts a small world technology to analyze several different topology structures including Circles, Wheels, Stars and Random. Through the combined tests of a topology and benchmark function, the conclusion is drawn that the best selection of a topology is related to the characteristics of problems. After that, Kennedy and Eberhart (2002) study a group of classic topologies: gbest, lbest, pyramid, von Neumann and four clusters. The results show that lbest can keep a high rate of

success but with low convergence speed. Such researches focus on a static topology. However, the experiments show that dynamic and unstable topology favors information interaction for individuals. Thus some PSO versions with dynamic topology and adjustment are reported. Based on a dynamic neighborhood, a new learning model is proposed (Nasir et al. 2012). It uses a learning strategy in which all other particles' historical best information is used to update a particle's velocity. To adaptively generate the population topology, a scale-free network model is used as a self-organizing construction mechanism in Zhang and Yi (2011). Some researchers also study the effect of synchronicity in communications and neighborhood size (Rada-Vilela et al. 2013). They suggest that a random asynchronous model may be better than synchronous and asynchronous methods.

### 2.2.2 Individual behavior control

Similar to other real or virtual swarms, particles in PSO improve their knowledge and ability by learning from the environment. But high convergence speed makes the diversity decrease fast. This means when particles crowd in the same area, they express similar characteristics but show little difference. Thus the particles cannot absorb new knowledge from their neighborhood. Therefore, their behaviors are restricted and evolution is stagnant. The long-standing question is how to control and adjust their behavior to keep the swarm in evolution.

The PSO core is the updating formula of particles, which contain three components (Kennedy 1997). The first one is the effect of their present velocity; the second one is the cognition modal for learning from their own memory; and the last one is the social modal which indicates the influence from the swarm or neighbor. Without the first one, a swarm would contract to the global best individual within the initial area, like a local search. With the first one only, the particles would keep flying until the boundary is reached, resembling global search. Based on the original PSO, Shi and Eberhart (1998a) introduce a modified PSO with inertia weight  $\omega$  to balance between exploration and exploitation as:

$$v_{id}^{t+1} = \omega v_{id}^t + c_1 r_1 (p_{id}^t - x_{id}^t) + c_2 r_2 (g_d^t - x_{id}^t) \quad (3)$$

The experiments show that a larger value of  $\omega$  would drive the swarm to exploit new areas, thereby causing the swarm dispersed, and smaller  $\omega$  would make the swarm tend to explore local areas resulting in precocity. To adjust  $\omega$  to balance the global and local search, Shi and Eberhart (1998b) provide a dynamic strategy to reduce  $\omega$  from 0.9 to 0.4 linearly according to the iterations during running. But the PSO search process is a complex nonlinear process. This linear method fails to perform well due to the nonlinearity.

Shi and Eberhart (2001) proposed a new method to adjust the inertia weight. Based on the present best solution and  $\omega$ , it calculates the changing rate of  $\omega$  by a defined fuzzy system to determine  $\omega$  in the next iteration.

Constriction is also a kind of method for controlling the behavior of particles. Clerc and Kennedy (2002) adopt a constriction factor  $\chi$  to modify the velocity as:

$$v_{id}^{t+1} = \chi [v_{id}^t + c_1 r_1 (p_{id}^t - x_{id}^t) + c_2 r_2 (g_d^t - x_{id}^t)] \quad (4)$$

$$\chi = \frac{2}{|2 - \varphi - \sqrt{\varphi^2 - 4\varphi}|} \quad (5)$$

For the choice between  $\omega$  and  $\chi$ , Eberhart and Shi (2000) compare the two models and conclude that they are algebraically equivalent.

In traditional PSO, each particle's behavior is affected by its previous success, best previous success of its neighbors, current positions and previous velocity. The information from remaining neighbors is not considered. Based on (Clerc and Kennedy 2002), researchers (Mendes et al. 2004; Kennedy and Mendes 2006) propose an algorithm called fully informed particle swarm optimization. In this model, a particle is affected by all its neighbors, not like in the original model in which each is affected by its own experience and the best success in its neighbors. The equation to update velocity is as follows:

$$v_{id}^{t+1} = \chi \left[ v_{id}^t + \sum_{n=1}^{K_i} \frac{U(0, \varphi) - (p_{nd}^t - x_{id}^t)}{K_i} \right] \quad (6)$$

To overcome the premature convergence and stagnation problems and enrich the PSO with better search capacity in a multidimensional search space, a method combining a constriction factor and inertia weight is proposed (Mandal et al. 2011).

Presently, an animated area is to combine desirable properties from different approaches to mitigate their weaknesses in SI algorithm research. Kang et al. (2012a) incorporate PSO with group search optimizer to provide a novel improved algorithm, in which the optimal information leading mechanism and computation mode of PSO are kept and the historical individual optimal information is introduced. In a chaos-embedded PSO (Alatas et al. 2009), sequences generated from a chaotic system to make a random choice for PSO to improve the global convergence and prevent premature termination. Similarly, ant colony optimization (ACO) is also an SI algorithm that simulates co-operational behaviors of ants for searching food to solve problems. Holden and Freitas (2005) provide a hybrid version of PSO and ACO and apply it to solve hierarchical classification problems successfully. A genetic algorithm (GA) is an evolutionary algorithm that uses mutation, crossover and selection operations to solve problems. It has better performance but low conver-

gence rate. A modified Broyden-Fletcher-Goldfarb-Shanno method is integrated into PSOs to improve a particle's local search behavior (Li et al. 2011). Culture algorithms have been frequently used to vary the parameters of an individual solution for optimization problems. Daneshyari and Yen (2011) introduce a cultural framework to adapt the personalized flight parameters of the mutated particles. Considerable work on hybridization of PSO with other algorithms is done as summarized in Banks et al. (2007).

### 3 Adaptive PSO based on clustering

By considering both swarm communication topology and individual behavior control, we propose an adaptive PSO method with dynamic division of population, based on the model of PSO with inertia weight. It focuses on two major aspects:

1. How to make a simple dynamic subpopulation structure? A community structure has the remarkable characteristics of social biology. In a human society active individuals would join different organizations to gain different knowledge. An individual can be absorbed into organizations passively or participate in some actively. This movement not only takes the individual to new developing scope but also brings different culture or information to the organizations. In this work, we incorporate clustering into the swarm evolutionary process in which a swarm is split according to the property of individuals, to form multiple subpopulations. This dynamic multi-subpopulation structure is used to improve the grouping behavior of the swarm and reduce the number of premature incidents. Based on the dynamic structure, a new velocity updating model is provided to adjust such learning way of an individual that the effects from the best ones in dynamic subpopulations are considered.
2. How to bring more intelligence to individuals in dynamic subpopulations? Obviously, within a dynamic subpopulation structure, heterogeneous sub-swarms with different sizes and ability will be formed. These particles in different clusters have different needs, e.g., some need to explore new areas and some want to exploit local areas. The issue is how to improve the search ability of the swarm based on heterogeneous multi-swarms. We propose a method to adaptively adjust parameters based on cluster evaluation that assess the state of subpopulations in the swarm after clustering, to adjust the behavior of particles.

We call the proposed algorithm as APSO-C. Its population decomposition and evaluation strategies will be further described next.

### 3.1 Dynamic population decomposition based on a clustering strategy (DPDC)

Similar to an organism society, the action of a single particle is mainly affected by the best ones in its neighborhood. Researches (Kennedy and Eberhart 2002; Mendes et al. 2003) point out that the swarm topology affects an individual’s behavior substantially. According to “Birds of a feather flock together”, we adopt a clustering method to divide a swarm into several subpopulations according to the similar properties dynamically. Different from some methods like exchanging individuals among subpopulations at regular intervals or taking a fixed size of subpopulations, this recombination of individuals is consistently operated within the optimization process, which is executed after swarm moving. After each division, we make several heterogeneous subpopulations with different number and characteristics of individuals. In this model, with the migration of the individuals among different subpopulations, the information exchanges are also enhanced.

A K-means algorithm is an effective method for clustering, and the core is to minimize the sum of distance between each particle and the centers. Some researchers (Kennedy 2000; Passaro and Starita 2006) have applied it to analyze the swarm, but here we modify the standard clustering operation and propose a novel updating equation for particles. Similarly, we select  $K$  members as the centers ( $\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_K$ ) and define the distance between two particles as  $\text{dis}(\mathbf{x}_i, \mathbf{x}_j) = \sum_{d=1}^D (x_{id} - x_{jd})^T (x_{id} - x_{jd})$ . Then the objective can be expressed as  $\text{Min } F = \sum_{k=1}^K \sum_{x_i \in C_k} \text{dis}(\mathbf{x}_i, \mathbf{p}_k)$ , where  $C_k$  is the  $k$ -th cluster. The operation can be described as follows:

- Step 1: Select  $K$  particles ( $\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_K$ ) randomly as the centers of clusters.
- Step 2: Calculate the distance between the rest of individuals and these centers. According to the minimum distance rule, assign the rest to the nearest cluster. Record the value of  $F = \sum_{k=1}^K \sum_{x_i \in C_k} \text{dis}(\mathbf{x}_i, \mathbf{p}_k)$ .
- Step 3: For each cluster, choose the average value of the particle positions as its new center.
- Step 4: If the termination condition is met (e.g. the maximum number of times is reached), output the result; and otherwise return to Step 2.

Actually, a swarm implicates an aggregative trend. The population will cover an area between  $\mathbf{p}_i^t$  and  $\mathbf{g}^t$ . Considering the self-aggregative behavior, we modify the clustering objective. The operation is designed as a progressive clustering mode associated with the swarm search process, which does not need to reach a steady state of swarm clustering at each step. We call it as dynamic population decomposition based on a clustering strategy (DPDC).

In this process, the best individual is selected as its cluster-best solution after clusters are formed. Considering the relations of all clusters, a ring structure is used to share and transmit the information among them in this work. Each cluster is connected to other two clusters in its own neighborhood in the cluster array under the ring topology, and in each cluster the particles communicate with each other by a structure like the *gbest* topology. Denote cluster  $j$ ’s best solution found so far by  $\mathbf{Cb}_j$  and best one in particle  $i$ ’s neighborhood is  $\mathbf{nbest}_i = (nb_{i1}, nb_{i2}, \dots, nb_{iD})$  where particle  $i$  belongs to cluster  $j$ . Then we make a comparison among  $\mathbf{Cb}_{j-1}$ ,  $\mathbf{Cb}_j$  and  $\mathbf{Cb}_{j+1}$ , and choose the better one as particle  $i$ ’s  $\mathbf{nbest}_i$ . Both  $\mathbf{g}$  and  $\mathbf{nbest}_i$  have important effect on the process of learning from the environment of a particle. Hence we rebuild its velocity updating mode.

$$v_{id}^{t+1} = \omega v_{id}^t + c_1 r_1 (p_{id}^t - x_{id}^t) + \frac{1}{2} c_2 r_2 [(g_{id}^t - x_{id}^t) + (nb_{id}^t - x_{id}^t)] \tag{7}$$

After the process, a new swarm structure is built dynamically under a ring topology among different clusters and an individual will update its position based on the information from its previous best, neighborhood best and global best found by a swarm. Thus, it can be seen that this strategy is different from other existing ones, (e.g., Kennedy 2000; Li 2010), which are mainly based on the classic topology and mode. Next, a novel adaptation strategy is presented.

### 3.2 Adaptation based on cluster evaluation strategy (ACE)

Each cluster formed following the above way has a different number of particles. Owing to the different ability of particles, clusters behave differently at the search level. For example, the average distance from the particles to the best particle may not be same. Thus we can classify all clusters based on the difference to find a new way to improve the search ability of particles. To describe ACE clearly, we give the following definitions first.

**Definition 1** The population of cluster  $j$  is denoted by  $|C_j|$ ; Define  $f_{ij}$  as particle  $i$ ’s fitness in cluster  $j$ . The average fitness of cluster  $j$  is  $a_j = \sum_{i=1}^{|C_j|} f_{ij} / |C_j|$ ; Define  $f_i$  as the particle  $i$ ’s fitness. The average fitness of a swarm is  $A = \sum_{i=1}^N f_i / N$  where  $N$  is population.

**Definition 2** The initial maximum and minimum inertia weights of particles are expressed as  $\omega_{\max}$  and  $\omega_{\min}$ . At generation  $t$  the inertia weight of particle  $i$  is  $\omega_i^{t+1}$ , which can be adjusted based on  $\omega_{\max}$ ,  $\omega_{\min}$  and  $\omega_i^t$  according to its state.

Since each individual is at a different position in a search process, it shows different characteristics and ability. The fitness can be used to show the difference. The particles at

a different stage need to strengthen their ability that they require when searching the space. For example, the particles in an inferior position need to escape from a poor area and speed up to explore other areas; while the particles in an advantageous position need to explore the area deeply to find a better solution. This novel algorithm improves the ability of particles based on a clustering operation. For each cluster, particles in it have different strengths and weaknesses. Thus the average fitness of the clusters is different. This difference can be used to evaluate the clusters and to adjust  $\omega$  for an individual.

Assume that the objective is to minimize the fitness. From the comparison, two different states are listed as follows:

*Case 1:*  $a_j \geq A$ : This state means that the subpopulation is in an inferior position which is far from the best one. Hence the search steps should be set bigger for improving the global search ability and avoiding local search. We adjust the searching step by increasing the inertia weight  $\omega$ ; and

*Case 2:*  $a_j < A$ : In this state, the subpopulation is located in a better area. For this subpopulation, they need to reduce the step length in search to improve local search ability and to avoid missing the optimal solutions nearby. Thus we adjust the step by reducing  $\omega$ .

According to the above analysis, each individual can realize the self-adaption adjustment of parameter  $\omega_i^t$  by Eqs. 8 and 9, respectively:

$$\omega_i^{t+1} = \omega_i^t + \max \{ \text{abs}(\omega_{\max} - \omega_i^t), \text{abs}(\omega_{\min} - \omega_i^t) \} \quad (8)$$

$$a_j \geq A$$

$$\omega_i^{t+1} = \omega_i^t - \min \{ \text{abs}(\omega_{\max} - \omega_i^t), \text{abs}(\omega_{\min} - \omega_i^t) \} \quad (9)$$

$$a_j < A$$

where particle  $i$  belongs to cluster  $j$ .

### 3.3 Step of APSO-C

Based on DPDC and ACE, the of APSO-C has two main parts. The first part is the swarm clustering in which a K-means method is adopted to cluster the swarm into several subpopulations, and the information interaction among subpopulations is performed with a ring-topology structure. The second one is parameter adjustment that aims to adjust the search behaviors of particles according to the states of subpopulations.

APSO-C adjusts the structure of a swarm and the behavior of a particle in every generation, as detailed in the following steps.

**Step 1:** Set randomly  $\mathbf{x}_i$  and  $\mathbf{v}_i$ ,  $i \in \mathbb{N}_N$  (where  $N$  is the size of the swarm). Calculate  $f_i$  and initialize  $\omega_{\max}$ ,  $\omega_{\min}$  and  $\omega_i$  for each individual;

**Step 2:** Select randomly  $K$  particles as the clusters' centers. Take a clustering method to divide the swarm and obtain subpopulations  $\mathbf{C}_j$ ,  $\forall j \in \mathbb{N}_K$ ;

**Step 3:** Calculate the fitness of each particle. Set the best position  $\mathbf{pbest}_i$  that is found so far by particle  $i$  where  $\forall i \in \mathbb{N}_{|C_j|}$  as the cluster  $j$ 's best position  $\mathbf{Cb}_j$  at present.

**Step 4:** Choose the best position in the neighborhoods as  $\mathbf{nbest}_i$  for each particle under the ring topology;

**Step 5:** Select the best position of the swarm found so far as  $\mathbf{gbest}$ ;

**Step 6:** If the termination condition is met, output the result; and otherwise, use Eqs. 8 and 9 to adjust  $\omega_i$ , and update  $\mathbf{v}_i$  and  $\mathbf{x}_i$  for each particle by Eqs. 7 and 2. Go to Step 2;

## 4 Experiments and discussion

### 4.1 Experiment and benchmark functions

To evaluate APSO-C, this work designs three experiments: Comparison under the same initial value, Comparison with the same iteration, and Sensitivity analysis of  $\omega_{\max}$  in APSO-C. We choose fourteen benchmark functions from (Liang et al. 2006; Hsieh et al. 2009; Suganthan et al. 2005). Note that  $f_1$  and  $f_2$  are unimodal,  $f_3$  (when the dimension is greater than three) and  $f_4$  are multimodal. They are four basic functions. Because some functions are separable and can be solved by using D 1-D searchers (Hsieh et al. 2009), where D is the dimension of the problem. Rotation, shifting and hybridization are introduced to overcome the known disadvantages of the functions. The new functions are more complex than the original ones. The properties of these functions are listed in Table 1.

**Table 1** Global optimum, search range, initialization range of the test functions

Fun	$f_{\min}$	Search Space	Initialization Range
$f_1$	0	$[-100, 100]^D$	$[-100, 75]^D$
$f_2$	0	$[-100, 100]^D$	$[-100, 75]^D$
$f_3$	0	$[-2.084, 2.084]^D$	$[-2.084, 2.084]^D$
$f_4$	0	$[-5.12, 5.12]^D$	$[-5.12, 3.5]^D$
$f_5$	-450	$[-100, 100]^D$	$[-100, 100]^D$
$f_6$	-450	$[-100, 100]^D$	$[-100, 100]^D$
$f_7$	-450	$[-100, 100]^D$	$[-100, 100]^D$
$f_8$	-450	$[-100, 100]^D$	$[-100, 100]^D$
$f_9$	390	$[-100, 100]^D$	$[-100, 100]^D$
$f_{10}$	-140	$[-32, 32]^D$	$[-32, 32]^D$
$f_{11}$	-330	$[-5, 5]^D$	$[-5, 5]^D$
$f_{12}$	-330	$[-5, 5]^D$	$[-5, 5]^D$
$f_{13}$	-130	$[-3, 1]^D$	$[-3, 1]^D$
$f_{14}$	-300	$[-100, 100]^D$	$[-100, 100]^D$

(1) Sphere function:

$$f_1 = \sum_{i=1}^D x_i^2$$

(2) Quadric function:

$$f_2 = \sum_{i=1}^D \left( \sum_{j=1}^i x_j \right)^2$$

(3) Rosenbrocks function:

$$f_3 = \sum_{i=1}^{D-1} \left( 100(x_{i+1}^2 - x_i)^2 + (x_i - 1)^2 \right)$$

(4) Rastrigin function:

$$f_4 = \sum_{i=1}^D (x_i^2 - 10 \cos(2\pi x_i) + 10)$$

(5) Shifted sphere function:

$$f_5 = \sum_{i=1}^D z_i^2 + f_{\text{bias}}$$

where  $z = x - o, o = [o_1, o_2, \dots, o_D]$

(6) Shifted schwefel's problem 1.2:

$$f_6 = \sum_{i=1}^D \left( \sum_{j=1}^i z_j \right)^2 + f_{\text{bias}}$$

where  $z = x - o, o = [o_1, o_2, \dots, o_D]$

(7) Shifted rotated high conditioned elliptic function:

$$f_7 = \sum_{i=1}^D (10^6)^{\frac{i-1}{D-1}} Z_i^2 + f_{\text{bias}}$$

where  $z = x - o * \mathbf{M}, o = [o_1, o_2, \dots, o_D]$

(8) Shifted schwefel's problem 1.2 with noise fitness:

$$f_9 = \left( \sum_{i=1}^D \left( \sum_{j=1}^i z_j^2 \right) \right) * (1 + 0.4|N(0, 1)|) + f_{\text{bias}}$$

where  $z = x - o, o = [o_1, o_2, \dots, o_D]$

(9) Shifted Rosenbrocks function:

$$f_8 = \sum_{i=1}^{D-1} \left( 100(z_i^2 - z_{i+1})^2 + (z_i - 1)^2 \right) + f_{\text{bias}}$$

where  $z = x - o, o = [o_1, o_2, \dots, o_D]$

(10) Shifted rotated Ackley's with global optimum on bounds:

$$f_{10} = -20 \exp \left( -0.2 \sqrt{\frac{1}{D} \sum_{i=1}^D x_i^2} \right) - \exp \left( \frac{1}{D} \sum_{i=1}^D \cos(2\pi x_i) \right) + 20 + e + f_{\text{bias}}$$

where  $z = (x - o) * \mathbf{M}, o = [o_1, o_2, \dots, o_D]$

(11) Shifted Rastrigin's function:

$$f_{11} = \sum_{i=1}^D \left( z_i^2 - 10 \cos(2\pi z_i) + 10 \right) + f_{\text{bias}}$$

where  $z = (x - o), o = [o_1, o_2, \dots, o_D]$

(12) Shifted rotated Rastrigin's function:

$$f_{12} = \sum_{j=1}^D \left( z_j^2 - 10 \cos(2\pi z_j) + 10 \right) + f_{\text{bias}}$$

where  $z = (x - o) * \mathbf{M}, o = [o_1, o_2, \dots, o_D]$

(13) Expanded extended Griewank's + Rosenbrock's:

$$f_{13} = f_s(f_3(z_1, z_2)) + f_s(f_3(z_2, z_3)) + \dots + f_s(f_3(z_{D-1}, z_D)) + f_{\text{bias}}$$

where  $f_s = \sum_{i=1}^D \frac{x_i^2}{400} - \prod_{i=1}^D \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$   
 $z = x - o + 1, o = [o_1, o_2, \dots, o_D]$

(14) Expanded rotated extended Scaffes's F6:

$$f_{14} = F(z_1, z_2) + F(z_2, z_3) + \dots + F(z_{D-1}, z_D) + F(z_D, z_1)$$

where  $z = x - o + 1, o = [o_1, o_2, \dots, o_D]$   
 $F(x, y) = 0.5 + \frac{\sin^2(\sqrt{x^2 + y^2} - 0.5)}{(1 + 0.001(x^2 + y^2))^2}$

Seven peer algorithms including FIPS (Mendes et al. 2004), PSO\_g and PSO\_l (Kennedy 1999), Biogeography-based optimization (BBO, Simon 2008), Firefly Algorithm (FA, Yang 2010), Artificial Bee Colony (ABC, Karaboga and

**Table 2** Comparison of APSO-C with different initialized  $\omega$  of 10-D function tests

Function	$\omega = 0.55$	0.60	0.65	0.70	0.75	0.80	0.85	0.90	0.95
<i>f</i> <sub>1</sub>									
Max	3.57E−68	1.28E−58	2.34E−58	9.57E−49	4.91E−52	5.65E−64	2.10E−66	1.11E−63	2.89E−66
Min	2.76E−78	2.60E−81	3.79E−82	3.56E−81	7.16E−81	3.55E−80	4.83E−77	1.11E−80	3.26E−80
Mean	1.80E−69	6.42E−60	1.17E−59	4.79E−50	2.45E−53	2.83E−65	1.27E−67	5.62E−65	1.55E−67
SD	7.97E−69	2.87E−59	5.23E−59	2.14E−49	1.10E−52	1.26E−64	4.68E−67	2.47E−64	6.44E−67
<i>f</i> <sub>2</sub>									
Max	8.53E−22	1.49E−21	2.37E−20	4.64E−19	9.70E−19	1.01E−21	2.33E−20	8.76E−23	8.04E−21
Min	6.89E−26	1.16E−27	6.00E−28	6.72E−27	2.33E−26	9.48E−29	8.22E−30	5.37E−27	3.80E−28
Mean	1.79E−22	1.58E−22	2.11E−21	2.33E−20	5.11E−20	7.48E−23	1.42E−21	1.09E−23	6.20E−22
SD	3.21E−22	4.28E−22	6.37E−21	1.04E−19	2.16E−19	2.28E−22	5.20E−21	1.95E−23	1.82E−21
<i>f</i> <sub>3</sub>									
Max	1.16E+02	2.53E+01	2.49E+02	7.37E+00	1.25E+02	1.13E+02	1.04E+02	1.80E+01	1.67E+02
Min	5.87E−01	8.61E−04	4.78E−04	9.92E−02	1.68E−03	3.81E−02	1.87E−04	7.58E−02	1.57E−01
Mean	9.13E+00	4.49E+00	2.14E+01	3.05E+00	1.60E+01	1.29E+01	7.62E+00	5.29E+00	1.13E+01
SD	2.54E+01	5.33E+00	5.78E+01	2.03E+00	3.37E+01	3.19E+01	2.27E+01	4.91E+00	3.66E+01
<i>f</i> <sub>4</sub>									
Max	6.96E+00	1.19E+01	5.97E+00	6.96E+00	6.96E+00	7.96E+00	5.97E+00	5.97E+00	5.97E+00
Min	9.95E−01	9.95E−01	9.95E−01	0.00E+00	9.95E−01	9.95E−01	9.95E−01	0.00E+00	9.95E−01
Mean	3.43E+00	3.83E+00	3.38E+00	3.38E+00	3.23E+00	3.58E+00	3.25E+00	2.74E+00	3.08E+00
SD	1.72E+00	2.76E+00	1.59E+00	1.95E+00	1.79E+00	2.08E+00	1.41E+00	1.48E+00	1.33E+00
<i>f</i> <sub>5</sub>									
Max	−450	−450	−450	−450	−450	−450	−450	−450	−450
Min	−450	−450	−450	−450	−450	−450	−450	−450	−450
Mean	−450	−450	−450	−450	−450	−450	−450	−450	−450
SD	0	0	0	0	0	0	0	0	0
<i>f</i> <sub>6</sub>									
Max	−4.50E+02	−4.50E+02	−4.50E+02	−4.50E+02	−4.50E+02	−4.50E+02	−4.50E+02	−4.50E+02	−4.50E+02
Min	−4.50E+02	−4.50E+02	−4.50E+02	−4.50E+02	−4.50E+02	−4.50E+02	−4.50E+02	−4.50E+02	−4.50E+02
Mean	−4.50E+02	−4.50E+02	−4.50E+02	−4.50E+02	−4.50E+02	−4.50E+02	−4.50E+02	−4.50E+02	−4.50E+02
SD	2.37E−05	8.82E−06	1.57E−04	2.63E−05	9.81E−06	9.44E−06	1.31E−05	2.69E−05	2.08E−05
<i>f</i> <sub>7</sub>									
Max	2.70E+05	3.92E+05	3.00E+05	3.88E+05	4.24E+05	2.78E+05	3.99E+05	2.88E+05	3.03E+05
Min	2.64E+04	1.62E+04	1.13E+04	6.14E+03	2.40E+04	3.23E+03	1.51E+04	1.38E+04	1.60E+04
Mean	9.74E+04	9.46E+04	1.03E+05	1.11E+05	1.05E+05	8.79E+04	9.97E+04	9.57E+04	9.86E+04
SD	6.16E+04	8.52E+04	7.89E+04	1.06E+05	1.01E+05	7.42E+04	1.05E+05	5.91E+04	8.28E+04
<i>f</i> <sub>8</sub>									
Max	−4.50E+02	−4.50E+02	−4.50E+02	−4.50E+02	−4.50E+02	−4.50E+02	−4.50E+02	−4.50E+02	−4.50E+02
Min	−4.50E+02	−4.50E+02	−4.50E+02	−4.50E+02	−4.50E+02	−4.50E+02	−4.50E+02	−4.50E+02	−4.50E+02
Mean	−4.50E+02	−4.50E+02	−4.50E+02	−4.50E+02	−4.50E+02	−4.50E+02	−4.50E+02	−4.50E+02	−4.50E+02
SD	4.44E−03	4.11E−03	1.23E−03	1.95E−03	2.42E−03	7.04E−03	1.49E−03	2.16E−03	3.13E−03
<i>f</i> <sub>9</sub>									
Max	6.03E+02	5.01E+02	5.74E+02	5.50E+02	3.99E+02	5.48E+02	5.54E+02	4.81E+02	6.37E+02
Min	3.91E+02	3.90E+02	3.90E+02	3.91E+02	3.92E+02	3.91E+02	3.90E+02	3.90E+02	3.90E+02
Mean	4.19E+02	4.00E+02	4.02E+02	4.31E+02	3.94E+02	4.01E+02	4.01E+02	3.97E+02	4.14E+02
SD	5.37E+01	2.45E+01	4.04E+01	6.10E+01	1.87E+00	3.45E+01	3.61E+01	1.99E+01	6.07E+01



**Table 2** continued

Function	$\omega = 0.55$	0.60	0.65	0.70	0.75	0.80	0.85	0.90	0.95
<i>f</i> <sub>10</sub>									
Max	-1.19E+02	-1.19E+02	-1.19E+02	-1.19E+02	-1.19E+02	-1.20E+02	-1.19E+02	-1.19E+02	-1.19E+02
Min	-1.20E+02	-1.20E+02	-1.20E+02	-1.20E+02	-1.20E+02	-1.20E+02	-1.20E+02	-1.20E+02	-1.20E+02
Mean	-1.20E+02	-1.20E+02	-1.20E+02	-1.20E+02	-1.20E+02	-1.20E+02	-1.20E+02	-1.20E+02	-1.20E+02
SD	6.74E-02	1.10E-01	1.28E-01	1.17E-01	7.89E-02	7.11E-02	1.22E-01	9.91E-02	9.04E-02
<i>f</i> <sub>11</sub>									
Max	-3.16E+02	-3.23E+02	-3.14E+02	-3.21E+02	-3.22E+02	-3.22E+02	-3.22E+02	-3.23E+02	-3.23E+02
Min	-3.30E+02	-3.28E+02	-3.29E+02	-3.30E+02	-3.29E+02	-3.29E+02	-3.29E+02	-3.28E+02	-3.30E+02
Mean	-3.25E+02	-3.26E+02	-3.26E+02	-3.26E+02	-3.26E+02	-3.26E+02	-3.25E+02	-3.26E+02	-3.26E+02
SD	3.01E+00	1.40E+00	3.36E+00	2.43E+00	2.04E+00	1.93E+00	1.98E+00	1.27E+00	1.92E+00
<i>f</i> <sub>12</sub>									
Max	-2.91E+02	-2.92E+02	-2.90E+02	-2.85E+02	-2.85E+02	-2.80E+02	-2.82E+02	-2.88E+02	-2.95E+02
Min	-3.22E+02	-3.27E+02	-3.25E+02	-3.23E+02	-3.22E+02	-3.23E+02	-3.26E+02	-3.23E+02	-3.25E+02
Mean	-3.06E+02	-3.14E+02	-3.10E+02	-3.08E+02	-3.10E+02	-3.05E+02	-3.11E+02	-3.08E+02	-3.12E+02
SD	9.88E+00	8.68E+00	1.01E+01	1.04E+01	1.07E+01	1.19E+01	1.07E+01	1.17E+01	8.01E+00
<i>f</i> <sub>13</sub>									
Max	-1.29E+02	-1.29E+02	-1.28E+02	-1.29E+02	-1.29E+02	-1.29E+02	-1.28E+02	-1.28E+02	-1.29E+02
Min	-1.30E+02	0 -1.30E+02	-1.30E+02	-1.29E+02	-1.30E+02	-1.30E+02	-1.30E+02	-1.30E+02	-1.30E+02
Mean	-1.29E+02	-1.29E+02	-1.29E+02	-1.29E+02	-1.29E+02	-1.29E+02	-1.29E+02	-1.29E+02	-1.29E+02
SD	2.77E-01	2.99E-01	4.38E-01	2.43E-01	2.84E-01	2.99E-01	3.81E-01	2.97E-01	2.36E-01
<i>f</i> <sub>14</sub>									
Max	-2.96E+02	-2.96E+02	-2.95E+02	-2.96E+02	-2.96E+02	-2.96E+02	-2.96E+02	-2.96E+02	-2.96E+02
Min	-2.97E+02	-2.97E+02	-2.97E+02	-2.98E+02	-2.98E+02	-2.97E+02	-2.98E+02	-2.98E+02	-2.98E+02
Mean	-2.97E+02	-2.97E+02	-2.96E+02	-2.97E+02	-2.97E+02	-2.97E+02	-2.97E+02	-2.97E+02	-2.97E+02
SD	3.70E-01	4.46E-01	4.30E-01	4.82E-01	4.58E-01	3.94E-01	5.03E-01	5.04E-01	4.18E-01

**Table 3** Comparison of APSO-C with different initialized  $\omega$  of 30-D function tests

Function	$\omega = 0.55$	0.60	0.65	0.70	0.75	0.80	0.85	0.90	0.95
<i>f</i> <sub>1</sub>									
Max	8.67E-38	6.22E-40	9.48E-26	1.20E-40	1.25E-32	5.72E-41	1.79E-48	6.02E-32	3.23E-22
Min	8.26E-72	2.48E-68	3.56E-69	5.43E-77	2.73E-72	3.94E-68	1.80E-70	2.68E-66	2.85E-70
Mean	4.34E-39	3.11E-41	4.83E-27	6.24E-42	6.23E-34	2.88E-42	9.77E-50	3.01E-33	1.61E-23
SD	1.94E-38	1.39E-40	2.12E-26	2.68E-41	2.78E-33	1.28E-41	4.00E-49	1.35E-32	7.22E-23
<i>f</i> <sub>2</sub>									
Max	1.27E-01	2.72E-01	1.23E-01	1.69E-01	2.16E-01	2.33E-01	2.08E-01	3.33E-01	3.81E-01
Min	1.16E-02	5.12E-03	2.08E-03	2.01E-03	6.83E-03	3.68E-03	4.46E-03	4.31E-03	1.38E-03
Mean	5.50E-02	6.19E-02	4.13E-02	6.03E-02	7.77E-02	7.39E-02	6.25E-02	9.39E-02	6.98E-02
SD	3.00E-02	7.04E-02	3.43E-02	4.73E-02	6.82E-02	7.52E-02	6.18E-02	9.41E-02	8.74E-02
<i>f</i> <sub>3</sub>									
Max	7.63E+01	1.32E+02	9.02E+01	1.33E+02	1.46E+02	1.86E+02	1.71E+02	2.39E+02	9.79E+01
Min	7.07E+00	2.05E-01	2.09E-01	2.98E-02	1.92E+00	5.00E+00	9.66E-01	2.56E+00	1.20E-01
Mean	2.33E+01	4.55E+01	3.68E+01	2.72E+01	4.03E+01	4.14E+01	4.52E+01	4.42E+01	2.69E+01
SD	1.90E+01	3.83E+01	3.06E+01	3.19E+01	3.72E+01	4.57E+01	5.00E+01	6.16E+01	3.26E+01

**Table 3** continued

Function	$\omega = 0.55$	0.60	0.65	0.70	0.75	0.80	0.85	0.90	0.95
$f_4$									
Max	8.56E+01	7.66E+01	8.46E+01	6.47E+01	8.86E+01	8.86E+01	8.95E+01	6.96E+01	9.15E+01
Min	3.28E+01	1.99E+01	2.49E+01	1.59E+01	2.39E+01	1.99E+01	2.19E+01	2.19E+01	1.39E+01
Mean	5.02E+01	4.62E+01	4.48E+01	4.35E+01	4.90E+01	4.84E+01	3.98E+01	4.40E+01	4.85E+01
SD	1.41E+01	1.33E+01	1.53E+01	1.30E+01	1.91E+01	1.46E+01	1.59E+01	1.43E+01	1.89E+01
$f_5$									
Max	-4.50E+02	-4.50E+02	-4.50E+02	-4.50E+02	-4.50E+02	-4.50E+02	-4.50E+02	-4.50E+02	-4.50E+02
Min	-4.50E+02	-4.50E+02	-4.50E+02	-4.50E+02	-4.50E+02	-4.50E+02	-4.50E+02	-4.50E+02	-4.50E+02
Mean	-4.50E+02	-4.50E+02	-4.50E+02	-4.50E+02	-4.50E+02	-4.50E+02	-4.50E+02	-4.50E+02	-4.50E+02
SD	1.16E-10	9.98E-11	9.77E-12	6.63E-12	6.73E-11	1.64E-10	5.02E-12	3.16E-10	2.37E-11
$f_6$									
Max	-4.47E+02	-4.47E+02	-4.47E+02	-4.44E+02	-4.47E+02	-4.43E+02	-4.46E+02	-4.48E+02	-4.48E+02
Min	-4.50E+02	-4.50E+02	-4.50E+02	-4.50E+02	-4.50E+02	-4.50E+02	-4.50E+02	-4.50E+02	-4.50E+02
Mean	-4.49E+02	-4.49E+02	-4.49E+02	-4.49E+02	-4.49E+02	-4.49E+02	-4.49E+02	-4.49E+02	-4.49E+02
SD	8.03E-01	9.27E-01	6.72E-01	1.28E+00	7.50E-01	1.53E+00	9.66E-01	5.50E-01	5.24E-01
$f_7$									
Max	7.94E+06	4.91E+06	7.19E+06	5.89E+06	5.88E+06	4.76E+06	4.87E+06	4.72E+06	4.10E+06
Min	9.85E+05	1.19E+06	1.27E+06	1.07E+06	1.27E+06	8.74E+05	1.27E+06	1.05E+06	6.54E+05
Mean	2.93E+06	2.80E+06	3.00E+06	2.62E+06	3.09E+06	2.58E+06	2.53E+06	2.58E+06	2.39E+06
SD	1.61E+06	1.01E+06	1.44E+06	1.24E+06	1.24E+06	1.23E+06	1.01E+06	1.03E+06	9.99E+05
$f_8$									
Max	2.12E+03	9.14E+02	1.74E+03	2.75E+03	2.39E+03	2.17E+03	2.24E+03	2.02E+03	1.67E+03
Min	-9.86E+00	-1.26E+02	-1.01E+01	-4.55E+00	-1.04E+02	2.38E+01	-1.92E+02	-1.56E+02	6.07E+01
Mean	6.57E+02	4.73E+02	9.17E+02	7.21E+02	5.49E+02	8.44E+02	7.06E+02	6.26E+02	6.31E+02
SD	6.17E+02	3.33E+02	5.26E+02	7.60E+02	5.71E+02	6.42E+02	5.93E+02	6.64E+02	4.37E+02
$f_9$									
Max	6.36E+02	7.35E+02	.99E+02	6.40E+02	9.31E+02	5.39E+02	6.82E+02	6.23E+02	6.69E+02
Min	3.95E+02	3.94E+02	3.93E+02	3.95E+02	3.94E+02	3.91E+02	3.92E+02	3.90E+02	3.92E+02
Mean	4.41E+02	4.73E+02	4.58E+02	4.44E+02	4.76E+02	4.36E+02	4.34E+02	4.56E+02	4.52E+02
SD	6.01E+01	9.47E+01	6.61E+01	6.58E+01	1.20E+02	4.67E+01	6.96E+01	6.59E+01	7.49E+01
$f_{10}$									
Max	-1.19E+02	-1.19E+02	-1.19E+02	-1.19E+02	-1.19E+02	-1.19E+02	-1.19E+02	-1.19E+02	-1.19E+02
Min	-1.19E+02	-1.19E+02	-1.19E+02	-1.19E+02	-1.19E+02	-1.19E+02	-1.19E+02	-1.19E+02	-1.19E+02
Mean	-1.19E+02	-1.19E+02	-1.19E+02	-1.19E+02	-1.19E+02	-1.19E+02	-1.19E+02	-1.19E+02	-1.19E+02
SD	6.43E-02	7.47E-02	8.42E-02	5.81E-02	6.05E-02	9.07E-02	9.59E-02	7.21E-02	7.52E-02
$f_{11}$									
Max	-2.39E+02	-2.22E+02	-2.70E+02	-2.52E+02	-2.56E+02	-2.54E+02	-2.66E+02	-1.94E+02	-2.58E+02
Min	-3.07E+02	-3.13E+02	-3.07E+02	-3.15E+02	-3.15E+02	-3.00E+02	-3.13E+02	3.06E+02	-3.11E+02
Mean	-2.80E+02	-2.88E+02	-2.92E+02	-2.91E+02	-2.86E+02	-2.82E+02	-2.91E+02	-2.82E+02	-2.90E+02
SD	1.64E+01	2.39E+01	1.12E+01	1.43E+01	1.71E+01	1.25E+01	1.34E+01	2.52E+01	1.37E+01
$f_{12}$									
Max	-2.46E+01	-5.84E+01	-9.62E+01	-8.12E+01	-9.52E+01	2.34E+01	-8.23E+01	-1.86E+01	-2.75E+01
Min	-2.37E+02	-2.31E+02	-2.21E+02	-2.60E+02	-2.64E+02	-2.44E+02	-2.65E+02	-2.32E+02	-2.89E+02
Mean	-1.55E+02	-1.50E+02	-1.62E+02	-1.77E+02	-1.70E+02	-1.37E+02	-1.80E+02	-1.48E+02	-1.68E+02
SD	5.61E+01	4.78E+01	3.92E+01	5.04E+01	4.30E+01	6.96E+01	4.74E+01	6.66E+01	5.83E+01

**Table 3** continued

Function	$\omega = 0.55$	0.60	0.65	0.70	0.75	0.80	0.85	0.90	0.95
<i>f</i> <sub>13</sub>									
Max	-1.13E+02	-1.18E+02	-1.16E+02	-1.13E+02	-1.06E+02	-1.15E+02	-1.17E+02	-1.18E+02	-1.18E+02
Min	-1.27E+02	-1.27E+02	-1.27E+02	-1.27E+02	-1.27E+02	-1.27E+02	-1.27E+02	-1.27E+02	-1.27E+02
Mean	-1.24E+02	-1.24E+02	-1.23E+02	-1.23E+02	-1.23E+02	-1.23E+02	-1.24E+02	-1.24E+02	-1.24E+02
SD	2.98E+00	2.59E+00	2.77E+00	4.18E+00	4.57E+00	2.85E+00	2.60E+00	2.13E+00	2.67E+00
<i>f</i> <sub>14</sub>									
Max	-2.86E+02	-2.87E+02	-2.86E+02	-2.87E+02	-2.86E+02	-2.87E+02	-2.87E+02	-2.86E+02	-2.87E+02
Min	-2.88E+02	-2.88E+02	-2.88E+02	-2.88E+02	-2.89E+02	-2.88E+02	-2.88E+02	-2.89E+02	-2.88E+02
Mean	-2.87E+02	-2.87E+02	-2.87E+02	-2.87E+02	-2.87E+02	-2.87E+02	-2.87E+02	-2.87E+02	-2.87E+02
SD	4.55E-01	3.73E-01	3.87E-01	3.42E-01	5.56E-01	3.14E-01	3.69E-01	4.86E-01	3.93E-01

Basturk 2007) and Comprehensive Learning PSO (CLPSO, Liang et al. 2006) are compared with APSO-C. In CLPSO, the individual would update the position based on probability in each dimension. Liner inertia weight is taken.  $\omega_{\max} = 0.95$ ,  $\omega_{\min} = 0.4$ ,  $c = 1.5$ , and the refreshing gap  $m = 7$ . FIPS uses all the neighbors to influence the velocity of particles. The constriction coefficient  $\chi = 0.729$ , and acceleration coefficient  $\varphi = 4.1$ . PSO<sub>g</sub> and PSO<sub>l</sub> are the canonical PSO with gbest topology and lbest (ring) topology. Both algorithms set the linear control of inertia weight.  $\omega_{\max} = 0.95$ ,  $\omega_{\min} = 0.4$ , and  $c_1 = c_2 = 1.5$ . In APSO-C,  $c_1 = c_2 = 2$ ,  $\omega_{\max} = 0.95$ ,  $\omega_{\min} = 0.4$ , initialized  $\omega = 0.65$  (exception in experiment 1) and  $K = 5$ . The parameters in BBO, FA and ABC are set as suggested. All the sizes of populations are set as 30 in each algorithm. These algorithms are executed on same machine with an Intel Core Duo CPU 2.10 GHz, 2G memory and Windows 7 OS.

#### 4.2 Experiment 1: sensitivity analysis of APSO-C

In APSO-C,  $\omega$  is a vital parameter that affects the search performance. We try to adjust it adaptively to reduce the dependence of an initial value. Here we define different values of  $\omega$  from 0.55 to 0.95 and the interval is 0.05. According to the analysis (Shi and Eberhart 1998a) of  $\omega$  in swarm search,  $\omega_{\min}$  is 0.4 and  $\omega_{\max}$  is 0.95. The maximum iteration counts are same as before. The results are shown in Tables 2 and 3. No matter of 10-D or 30-D tests, for the unimodal problems, APSO-C obtains good solutions with different  $\omega$  under the same number of iterations. From the mean and SD, the solutions are relatively excellent. For most multimodal problems, the proposed algorithm also performs well and obtains the best values. Various values influence the results but slightly.

From the analysis of results we can see that different initial values of  $\omega$  have little effect on the performance of APSO-C. In conclusion, the adaptive operation can reduce its depen-

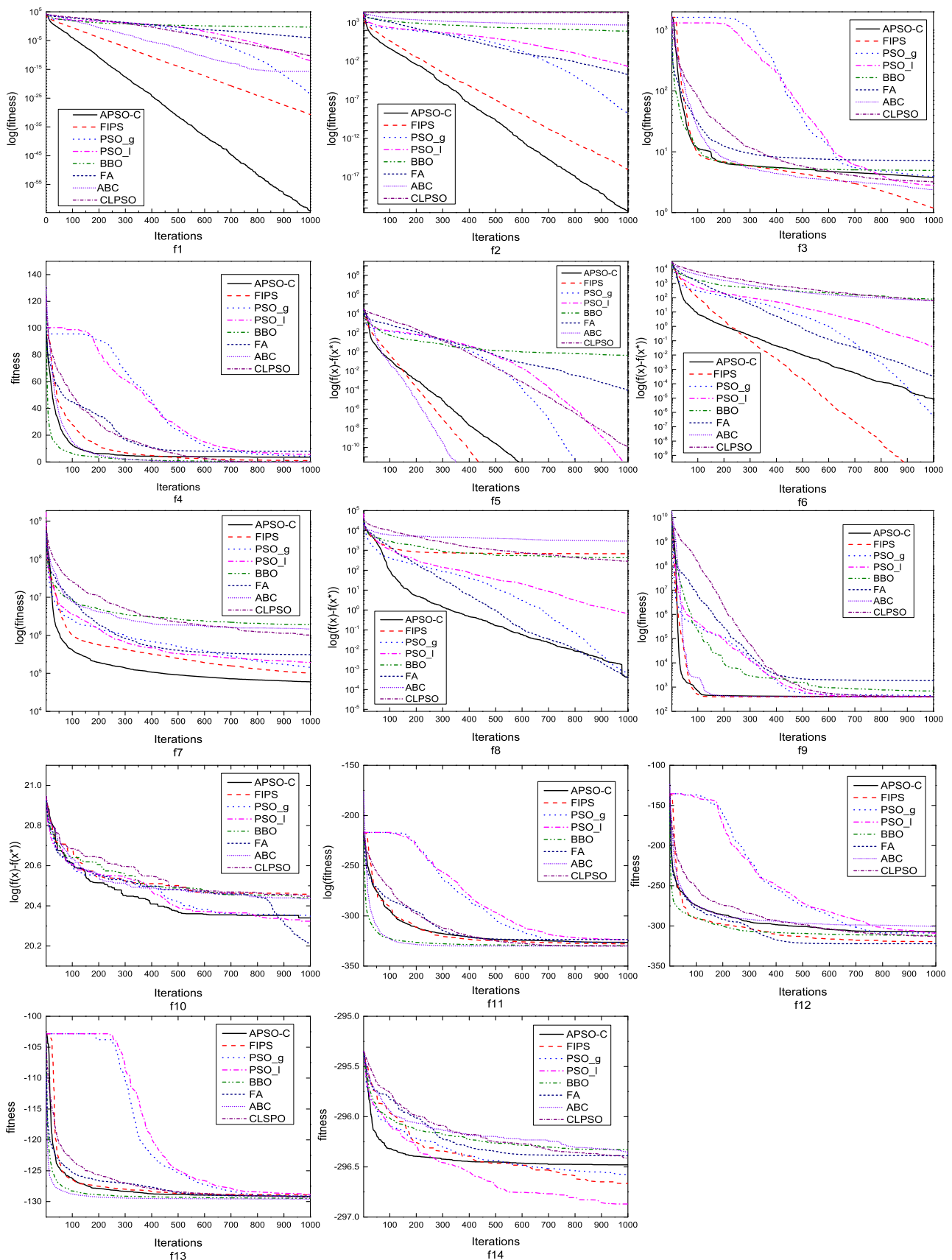
dence on initial values of  $\omega$ . That means the initial value can be set as a random value between  $\omega_{\min}$  and  $\omega_{\max}$ .

#### 4.3 Experiment 2: comparison with same value

In this experiment, APSO-C is compared with other PSOs and swarm intelligence optimization algorithms on their search coverage. All the algorithms are executed with the same initial values for each benchmark function. The maximum iteration count is set 1,000 for 10-D problems, and 3,000 for 30-D problems. Each algorithm is executed 20 times. The curves are obtained based on the average performances.

Figures 1 and 2 reveal the results of the selected algorithms in 10-D and 30-D problems. From the comparison, it can be seen that, APSO-C offers, more or less, better performance, no matter unimodal or multimodal functions. Because unimodal benchmark functions  $f_1$  and  $f_2$  are changed mildly in the scope. Most test algorithms have slow search, but APSO-C has a fast convergence speed and obtains the highest precisions solution.

For multimodal problems, no algorithm has obvious advantages than others. For example, BBO is a little better than APSO-C in  $f_{10}$  and  $f_{12}$ , but inferior in  $f_5$  and  $f_9$ , and their performances on other problems are similar. The results suggest that the different performance of algorithms may depend on the characteristics of the problems. They also suggest that APSO-C has some minor advantage on average. No matter how multimodal functions differ, APSO-C can obtain better performance or one close to the best ones found by other algorithms. It is more stable than others. This implies that the proposed algorithm has good search ability and fast convergence. With a dynamic multi-swarm structure and adaptation, APSO-C eases the conflicts between convergence speed and premature termination, and balances global



**Fig. 1** Median convergence characteristics of 10-dimensional problems

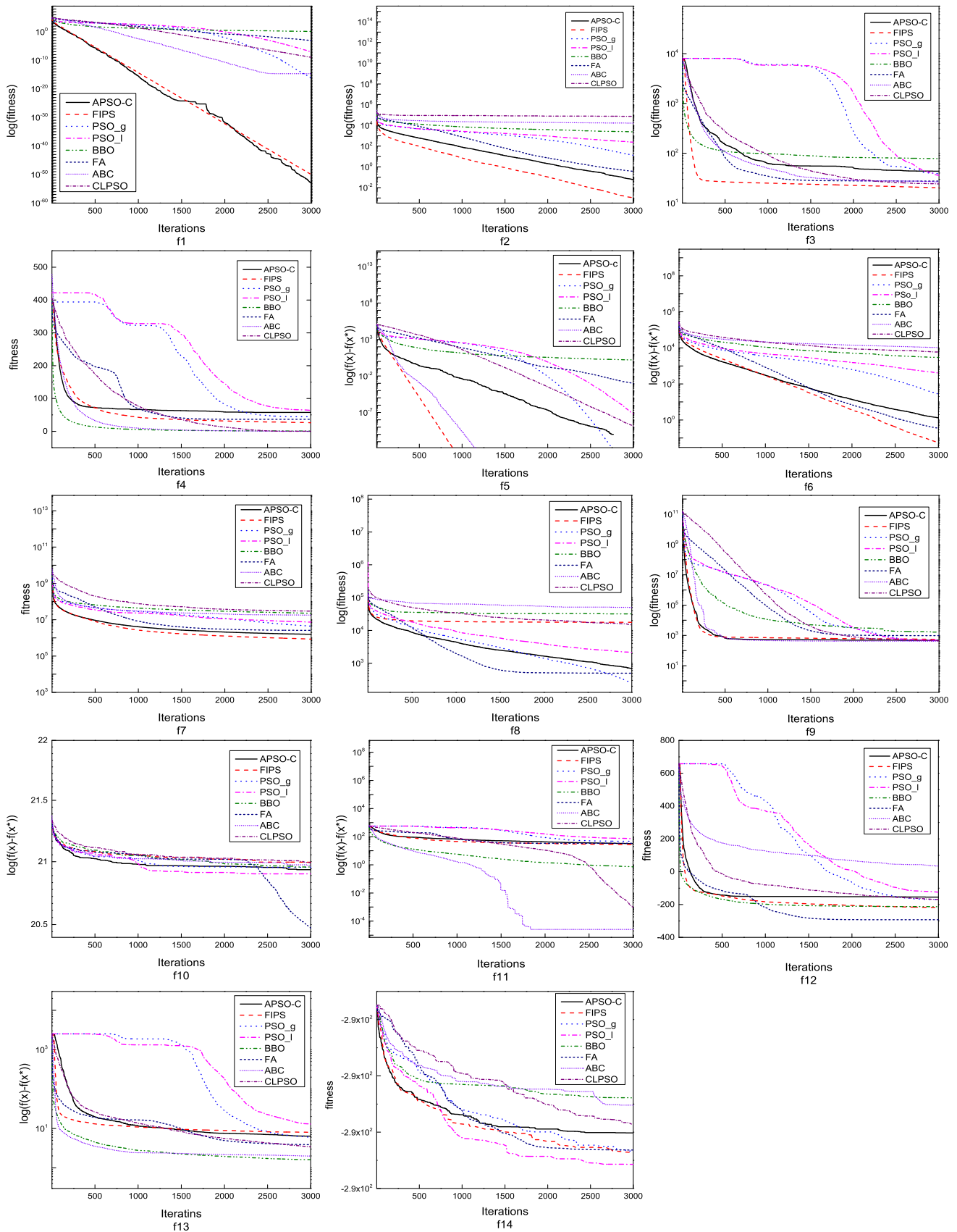


Fig. 2 Median convergence characteristics of 30-dimension problems

**Table 4** Comparison results of min, mean, max and SD in 10 dimensions

Algorithm	Min	Mean	Max	SD	Min	Mean	Max	SD
	$f_1$				$f_2$			
APSO-C	1.82E-77	2.58E-63	5.14E-62	1.15E-62	1.54E-27	8.74E-23	9.10E-22	2.07E-22
FIPS	1.33E-32	2.42E-31	7.36E-31	2.30E-31	4.44E-18	1.30E-15	7.68E-15	2.16E-15
PSO_g	1.14E-25	2.48E-21	3.17E-20	7.23E-21	1.81E-09	5.82E-08	3.55E-07	9.62E-08
PSO_l	7.69E-15	2.78E-12	1.86E-11	4.35E-12	2.22E-04	3.19E-03	1.74E-02	4.43E-03
BBO	2.34E-01	5.56E-01	1.39E+00	3.22E-01	8.85E+00	7.53E+01	2.02E+02	6.33E+01
FA	4.04E-05	9.83E-05	2.36E-04	4.29E-05	8.19E-05	3.06E-04	9.75E-04	2.16E-04
ABC	9.48E-17	2.12E-16	2.97E-16	6.94E-17	3.40E+02	6.03E+02	9.63E+02	1.76E+02
CLPSO	4.05E-12	5.99E-11	1.63E-10	4.17E-11	9.64E+03	1.78E+04	3.42E+04	6.65E+03
	$f_3$				$f_4$			
APSO-C	1.40E-04	2.91E+00	7.29E+00	1.98E+00	9.95E-01	1.94E+00	2.98E+00	6.83E-01
FIPS	9.58E-01	1.52E+00	2.10E+00	2.85E-01	1.74E-13	1.53E+00	3.98E+00	1.19E+00
PSO_g	1.38E+00	4.73E+00	8.95E+00	1.75E+00	1.77E-08	3.85E+00	6.96E+00	1.84E+00
PSO_l	2.47E+00	5.14E+00	6.68E+00	1.44E+00	1.99E+00	5.49E+00	1.06E+01	2.49E+00
BBO	4.48E-01	7.09E+00	6.76E+01	1.46E+01	1.10E-01	3.01E-01	8.37E-01	2.10E-01
FA	5.47E+00	7.36E+00	9.59E+00	1.18E+00	3.98E+00	7.71E+00	1.89E+01	3.43E+00
ABC	3.60E-01	2.03E+00	5.36E+00	1.39E+00	1.78E-15	4.20E-07	8.29E-06	1.85E-06
CLPSO	1.56E+00	3.33E+00	4.76E+00	8.58E-01	1.33E-06	3.04E-05	1.93E-04	5.22E-05
	$f_5$				$f_6$			
APSO-C	-4.50E+02	-4.50E+02	-4.50E+02	0.00E+00	-4.50E+02	-4.50E+02	-4.50E+02	6.22E-06
FIPS	-4.50E+02	-4.50E+02	-4.50E+02	0.00E+00	-4.50E+02	-4.50E+02	-4.50E+02	1.39E-11
PSO_g	-4.50E+02	-4.50E+02	-4.50E+02	0.00E+00	-4.50E+02	-4.50E+02	-4.50E+02	6.35E-07
PSO_l	-4.50E+02	-4.50E+02	-4.50E+02	1.76E-13	-4.50E+02	-4.50E+02	-4.50E+02	2.28E-02
BBO	-4.50E+02	-4.49E+02	-4.48E+02	4.22E-01	-4.33E+02	-3.45E+02	-1.60E+02	7.83E+01
FA	-4.50E+02	-4.50E+02	-4.50E+02	3.30E-05	-4.50E+02	-4.50E+02	-4.50E+02	1.33E-04
ABC	-4.50E+02	-4.50E+02	-4.50E+02	0.00E+00	-4.42E+02	-3.88E+02	-3.08E+02	3.65E+01
CLPSO	-4.50E+02	-4.50E+02	-4.50E+02	1.48E-10	-4.29E+02	-3.92E+02	-3.30E+02	2.60E+01
	$f_7$				$f_8$			
APSO-C	5.73E+03	9.27E+04	2.36E+05	6.78E+04	-4.50E+02	-4.50E+02	-4.50E+02	1.97E-03
FIPS	2.85E+04	1.19E+05	4.98E+05	1.09E+05	-4.34E+02	3.83E+02	2.47E+03	9.38E+02
PSO_g	2.54E+04	2.00E+05	1.34E+06	2.85E+05	-4.50E+02	-4.50E+02	-4.50E+02	5.53E-04
PSO_l	4.86E+04	2.77E+05	7.23E+05	1.78E+05	-4.50E+02	-4.49E+02	-4.44E+02	1.49E+00
BBO	5.21E+05	3.00E+06	7.89E+06	2.14E+06	-3.91E+02	2.63E+02	1.69E+03	4.85E+02
FA	2.02E+04	1.80E+05	6.71E+05	1.69E+05	-4.50E+02	-4.50E+02	-4.50E+02	2.36E-04
ABC	3.04E+05	1.60E+06	3.72E+06	9.24E+05	2.66E+02	2.77E+03	6.02E+03	1.53E+03
CLPSO	1.88E+05	9.68E+05	2.94E+06	6.59E+05	-4.09E+02	-2.14E+02	2.37E+01	1.06E+02
	$f_9$				$f_{10}$			
APSO-C	3.90E+02	3.93E+02	3.97E+02	1.83E+00	-4.50E+02	-4.50E+02	-4.50E+02	1.97E-03
FIPS	3.91E+02	4.39E+02	9.36E+02	1.33E+02	-4.34E+02	3.83E+02	2.47E+03	9.38E+02
PSO_g	3.91E+02	4.21E+02	6.15E+02	5.68E+01	-4.50E+02	-4.50E+02	-4.50E+02	5.53E-04
PSO_l	3.90E+02	3.94E+02	3.99E+02	2.70E+00	-4.50E+02	-4.49E+02	-4.44E+02	1.49E+00
BBO	4.52E+02	7.77E+02	1.67E+03	3.29E+02	-1.20E+02	-1.20E+02	-1.19E+02	8.68E-02
FA	3.96E+02	1.21E+03	8.91E+03	1.92E+03	-1.20E+02	-1.20E+02	-1.20E+02	8.68E-02
ABC	3.90E+02	3.99E+02	4.29E+02	1.01E+01	-1.20E+02	-1.20E+02	-1.20E+02	5.92E-02
CLPSO	3.93E+02	4.05E+02	4.24E+02	9.39E+00	-1.20E+02	-1.20E+02	-1.19E+02	1.05E-01

**Table 4** continued

Algorithm	Min	Mean	Max	SD	Min	Mean	Max	SD
	$f_{11}$				$f_{12}$			
APSO-C	3.90E+02	3.93E+02	3.97E+02	1.83E+00	-3.21E+02	-3.11E+02	-3.02E+02	5.49E+00
FIPS	3.91E+02	4.39E+02	9.36E+02	1.33E+02	-3.27E+02	-3.20E+02	-3.09E+02	4.44E+00
PSO <sub>g</sub>	3.91E+02	4.21E+02	6.15E+02	5.68E+01	-3.19E+02	-3.04E+02	-2.88E+02	7.78E+00
PSO <sub>l</sub>	3.90E+02	3.94E+02	3.99E+02	2.70E+00	-3.22E+02	-3.06E+02	-2.94E+02	7.28E+00
BBO	-3.30E+02	-3.30E+02	-3.29E+02	1.51E-01	-3.19E+02	-3.01E+02	-2.76E+02	1.08E+01
FA	-3.27E+02	-3.23E+02	-3.12E+02	3.61E+00	-3.27E+02	-3.20E+02	-3.09E+02	4.12E+00
ABC	-3.30E+02	-3.30E+02	-3.30E+02	0.00E+00	-3.11E+02	-2.97E+02	-2.77E+02	8.58E+00
CLPSO	-3.30E+02	-3.30E+02	-3.30E+02	1.74E-05	-3.21E+02	3.12E+02	-3.02E+02	4.66E+00
	$f_{13}$				$f_{14}$			
APSO-C	-1.30E+02	-1.29E+02	-1.28E+02	3.16E-01	-2.97E+02	-2.97E+02	-2.96E+02	4.04E-01
FIPS	-1.29E+02	-1.29E+02	-1.28E+02	2.28E-01	-2.97E+02	-2.97E+02	-2.96E+02	2.60E-01
PSO <sub>g</sub>	-1.29E+02	-1.29E+02	-1.29E+02	2.78E-01	-2.98E+02	-2.97E+02	-2.96E+02	4.70E-01
PSO <sub>l</sub>	-1.30E+02	-1.29E+02	-1.28E+02	3.16E-01	-2.98E+02	-2.97E+02	-2.96E+02	2.91E-01
BBO	-1.30E+02	7- 1.29E+02	- 1.29E+02	1.86E-01	-2.97E+02	-2.96E+02	-2.96E+02	3.08E-01
FA	-1.30E+02	-1.29E+02	-1.29E+02	3.25E-01	-2.98E+02	-2.96E+02	-2.96E+02	3.84E-01
ABC	-1.30E+02	-1.30E+02	-1.29E+02	7.87E-02	-2.97E+02	-2.96E+02	-2.96E+02	2.13E-01
CLPSO	-1.30E+02	-1.29E+02	-1.29E+02	1.60E-01	-2.97E+02	-2.96E+02	-2.96E+02	1.12E-01

**Table 5** Comparison Results of Min, Mean, Max and SD in 30 Dimensions

Algorithm	Min	Mean	Max	SD	Min	Mean	Max	SD
	$f_1$				$f_2$			
APSO-C	2.35E-68	4.37E-53	6.41E-52	1.44E-52	1.94E-06	5.19E-05	1.65E-04	4.72E-05
FIPS	2.44E-51	6.44E-51	1.32E-50	2.77E-51	8.43E-05	5.84E-04	2.17E-03	5.09E-04
PSO <sub>g</sub>	1.63E-22	1.14E-18	3.90E-18	1.35E-18	2.54E+00	1.23E+01	2.80E+01	7.68E+00
PSO <sub>l</sub>	5.86E-09	3.75E-08	8.07E-08	1.86E-08	1.25E+02	2.17E+02	3.36E+02	6.26E+01
BBO	5.71E-01	1.39E+00	2.19E+00	5.11E-01	1.39E+03	2.63E+03	4.13E+03	7.04E+02
FA	6.01E-04	8.96E-04	1.21E-03	1.84E-04	3.29E-02	4.65E-01	3.04E+00	6.49E-01
ABC	7.17E-16	3.23E-15	1.55E-14	3.66E-15	1.27E+04	1.69E+04	1.99E+04	2.15E+03
CLPSO	3.91E-10	1.08E-09	2.93E-09	5.61E-10	1.39E+04	4.17E+04	9.67E+04	2.14E+04
	$f_3$				$f_4$			
APSO-C	3.88E-03	4.81E+01	1.96E+02	4.97E+01	2.19E+01	5.66E+01	1.27E+02	2.43E+01
FIPS	1.90E+01	1.96E+01	2.00E+01	3.13E-01	1.50E+01	2.63E+01	5.40E+01	9.48E+00
PSO <sub>g</sub>	1.54E+01	3.97E+01	7.92E+01	2.44E+01	2.59E+01	3.91E+01	5.27E+01	7.66E+00
PSO <sub>l</sub>	2.06E+01	3.68E+01	8.14E+01	2.00E+01	4.51E+01	6.37E+01	9.37E+01	1.27E+01
BBO	1.06E+01	5.52E+01	8.21E+01	2.75E+01	2.82E-01	7.39E-01	1.73E+00	3.11E-01
FA	2.55E+01	2.75E+01	2.91E+01	1.00E+00	2.09E+01	3.85E+01	6.37E+01	1.18E+01
ABC	1.84E+01	2.26E+01	2.84E+01	2.89E+00	2.61E-10	5.32E-02	9.95E-01	2.22E-01
CLPSO	1.75E+01	2.21E+01	2.69E+01	2.65E+00	5.20E-04	1.89E-03	7.61E-03	1.53E-03
	$f_5$				$f_6$			
APSO-C	-4.50E+02	-4.50E+02	-4.50E+02	4.45E-11	-4.50E+02	-4.49E+02	-4.47E+02	5.30E-01
FIPS	-4.50E+02	-4.50E+02	-4.50E+02	2.26E-14	-4.50E+02	-4.50E+02	-4.50E+02	4.15E-02
PSO <sub>g</sub>	-4.50E+02	-4.50E+02	-4.50E+02	3.45E-14	-4.40E+02	-4.20E+02	-3.64E+02	1.95E+01
PSO <sub>l</sub>	-4.50E+02	-4.50E+02	-4.50E+02	9.91E-08	-2.36E+02	-1.05E+02	9.42E+01	9.57E+01
BBO	-4.49E+02	-4.48E+02	-4.47E+02	6.48E-01	7.86E+02	2.51E+03	4.26E+03	9.74E+02

Table 5 continued

Algorithm	Min	Mean	Max	SD	Min	Mean	Max	SD
FA	-4.50E+02	-4.50E+02	-4.50E+02	1.35E-04	-4.50E+02	-4.50E+02	-4.49E+02	1.94E-01
ABC	-4.50E+02	-4.50E+02	-4.50E+02	0.00E+00	5.02E+03	9.55E+03	1.52E+04	2.78E+03
CLPSO	-4.50E+02	-4.50E+02	-4.50E+02	6.41E-10	4.67E+03	5.74E+03	7.68E+03	8.91E+02
	$f_7$				$f_8$			
APSO-C	1.15E+06	3.22E+06	5.97E+06	1.42E+06	-2.57E+02	2.45E+02	7.19E+02	3.22E+02
FIPS	4.29E+05	8.52E+05	1.48E+06	2.68E+05	1.27E+04	1.83E+04	2.84E+04	3.84E+03
PSO_g	9.81E+05	6.09E+06	1.33E+07	3.63E+06	-2.38E+02	4.47E+02	3.18E+03	7.77E+02
PSO_l	2.08E+06	8.79E+06	1.68E+07	3.73E+06	8.78E+02	2.57E+03	5.06E+03	1.17E+03
BBO	1.18E+07	2.03E+07	2.91E+07	4.99E+06	1.78E+04	3.00E+04	4.52E+04	8.01E+03
FA	4.49E+05	2.05E+06	3.71E+06	8.20E+05	-4.46E+02	1.17E+03	4.93E+03	1.63E+03
ABC	6.11E+06	1.59E+07	2.63E+07	5.67E+06	2.77E+04	5.07E+04	6.28E+04	7.36E+03
CLPSO	1.66E+07	3.16E+07	4.96E+07	9.28E+06	1.15E+04	1.75E+04	2.21E+04	3.30E+03
	$f_9$				$f_{10}$			
APSO-C	3.93E+02	4.07E+02	4.18E+02	9.14E+00	-1.19E+02	-1.19E+02	-1.19E+02	7.00E-02
FIPS	4.08E+02	5.10E+02	7.65E+02	1.05E+02	-1.19E+02	-1.19E+02	-1.19E+02	6.78E-02
PSO_g	4.01E+02	6.27E+02	1.75E+03	3.95E+02	-1.19E+02	-1.19E+02	-1.19E+02	6.43E-02
PSO_l	3.94E+02	4.79E+02	5.89E+02	6.33E+01	-1.19E+02	-1.19E+02	-1.19E+02	6.28E-02
BBO	6.76E+02	1.89E+03	6.53E+03	1.46E+03	-1.19E+02	-1.19E+02	-1.19E+02	6.28E-02
FA	4.17E+02	2.84E+03	1.27E+04	4.27E+03	-1.20E+02	-1.19E+02	-1.19E+02	8.11E-02
ABC	3.90E+02	4.01E+02	4.59E+02	1.55E+01	-1.19E+02	-1.19E+02	-1.19E+02	6.99E-02
CLPSO	4.08E+02	4.42E+02	4.92E+02	2.45E+01	-1.19E+02	-1.19E+02	-1.19E+02	5.37E-02
	$f_{11}$				$f_{12}$			
APSO-C	-3.13E+02	-2.98E+02	-2.80E+02	9.66E+00	-2.39E+02	-1.88E+02	-1.54E+01	6.76E+01
FIPS	-3.13E+02	-3.01E+02	-2.85E+02	7.31E+00	-2.67E+02	-2.24E+02	-1.86E+02	2.21E+01
PSO_g	-3.12E+02	-2.83E+02	-2.62E+02	1.24E+01	-2.49E+02	-1.74E+02	-1.15E+02	4.04E+01
PSO_l	-2.86E+02	-2.57E+02	-2.13E+02	1.92E+01	-1.97E+02	-1.29E+02	-3.36E+01	4.42E+01
BBO	-3.30E+02	-3.29E+02	-3.29E+02	3.40E-01	-2.58E+02	-2.02E+02	-1.13E+02	3.37E+01
FA	-3.10E+02	-2.92E+02	-2.62E+02	1.27E+01	-3.09E+02	-2.94E+02	-2.81E+02	8.66E+00
ABC	-3.30E+02	-3.30E+02	-3.30E+02	1.03E-06	-1.18E+02	-4.33E+00	7.60E+01	4.61E+01
CLPSO	-3.30E+02	-3.30E+02	-3.30E+02	8.10E-04	-2.13E+02	-1.70E+02	-1.36E+02	1.94E+01
	$f_{13}$				$f_{14}$			
APSO-C	-1.27E+02	-1.24E+02	-1.20E+02	1.73E+00	-2.88E+02	-2.87E+02	-2.87E+02	3.06E-01
FIPS	-1.26E+02	-1.21E+02	-1.17E+02	2.32E+00	-2.88E+02	-2.87E+02	-2.87E+02	2.57E-01
PSO_g	-1.27E+02	-1.24E+02	-1.17E+02	2.04E+00	-2.89E+02	-2.88E+02	-2.87E+02	4.87E-01
PSO_l	-1.23E+02	-1.18E+02	-1.12E+02	2.78E+00	-2.88E+02	-2.88E+02	-2.87E+02	3.49E-01
BBO	-1.29E+02	-1.28E+02	-1.28E+02	3.53E-01	-2.88E+02	-2.87E+02	-2.86E+02	2.92E-01
FA	-1.28E+02	-1.25E+02	-1.21E+02	1.65E+00	-2.88E+02	-2.87E+02	-2.86E+02	4.36E-01
ABC	-1.28E+02	-1.28E+02	-1.27E+02	2.75E-01	-2.87E+02	-2.87E+02	-2.86E+02	1.55E-01
CLPSO	-1.27E+02	-1.27E+02	-1.26E+02	2.73E-01	-2.87E+02	-2.87E+02	-2.87E+02	1.82E-01

search and local search. Regardless unimodal or multimodal problems, APSO-C can obtain highly satisfied solutions.

#### 4.4 Experiment 3: comparison with the same iteration count

In the second experiment, APSO-C with other four algorithms is compared with random initial value for each function. All algorithms are also executed 20 times independently.

As in the first experiment, the maximum iteration counts for 10-D and 30-D problems are, respectively, 1,000 and 3,000. The statistics includes the minimum value (min), maximum value (max), mean value (mean) and standard deviation (SD). The results can be seen in Tables 4 and 5.

It is clear from the results that APSO-C obtains much better solution than other algorithms in  $f_1$ ,  $f_2$  and  $f_3$  under the same number of iterations. However, though APSO-C



meets the accuracy requirements, it is inferior to others in the test of  $f_4$  with 30 dimensions. For  $f_5$ – $f_{14}$ , APSO-C offers good performance and obtains the global best solutions for some of them. Through the operations of dynamical clustering and adaptive adjusting, the proposed APSO-C improves the performance of local search, global search and robustness. The ability of particles is strengthened by considering comprehensive attractors of the best solution from the swarm and subpopulations. The solution efficiency and accuracy are increased for most benchmark problems.

## 5 Conclusion

This paper presents an adaptive PSO method (APSO-C) based on swarm structure topology and individual behavior control. Two different strategies are proposed, i.e., dynamic population decomposition based on a clustering strategy and adaptation based on a cluster evaluation strategy. In the first strategy, the population is divided via a K-means method into several clusters that contain different numbers of particles with different ability. Then the clusters take a ring topology to exchange information. At the second strategy, through the cluster ability evaluation on searching status and parameter adaption for each individual, particles adjust their parameters to balance their local and global search.

Results from the experiments show that APSO-C obtains a better result than some other typical PSO algorithms in literature and reaches the expectations in terms of convergence speed, solution accuracy, and parameter sensitivity given the same benchmark functions.

In the future work, we will focus on the following issues:

1. To evaluate the search ability of an individual particle, we need to consider the comparison value with the average fitness of dynamic clusters and the swarm. One of the important questions is whether there are other effective evaluation methods;
2. The effects of best positions in neighborhood and in a swarm are treated the same for an individual in this work. In the future their different effect will be investigated;
3. The clustering process is essential to obtain subpopulations, and is also the key influencing factor of algorithm efficiency. From some tests, the swarm can obtain a good state of subpopulation division in APSO-C after executing a K-means method for two to four times. But how to choose the optimal number of execution times and number of clusters remains open;
4. Because the swarm has a bias towards the single objective in the proposed algorithm, multiple global optima may confuse the individuals. Thus, more study is needed on “multi-optima issue” (Dong and Zhou 2014) in our future

research. Furthermore, the application of the proposed method to some industrial optimization problems (Fang et al. 2012; Kang et al. 2013; Wu and Zhou 2007; Xing et al. 2012; Xiong et al. 2009; Yu et al. 2009) should be pursued.

**Acknowledgments** This work was supported in part by the Key Program of Natural Science Foundation of Hubei Province (No. 2010CDA022), the international cooperation project of Hubei province (No. 2011BFA012), National Natural Science Foundation of China (No. 71372202), the national key Technology Research and Development Program (Nos. 2012BAJ05B07 and 2014BAH24F03), the PRC Ministry of Science and Technology under Contract No. 2013DFM10100, the Fundamental Research Funds for the Central Universities (WUT: 2013-IV-057), and the Opening Project of Traffic Transport Industry Key Laboratory of Port Handling Technology (No. 2013-gkzx-k-01).

## References

- Alatas B, Akin E, Ozer AB (2009) Chaos embedded particle swarm optimization algorithms. *Chaos Soliton Fract* 40(4):1715–1734
- Banks A, Vincent J, Anyakoha C (2007) A review of particle swarm optimization. Part I: background and development. *Nat Comp Ser* 6(4):467–484
- Brits R, Engelbrecht AP, Van den Bergh F (2002) A niching particle swarm optimizer. In: *Proceedings of the 4th Asia-Pacific conference on simulated evolution and learning*, Orchid Country Club, Singapore, vol 2, pp 692–696
- Chander A, Chatterjee A, Siarry P (2011) A new social and momentum component adaptive PSO algorithm for image segmentation. *Expert Syst Appl* 38(5):4998–5004
- Chen D, Zhao C (2009) Particle swarm optimization with adaptive population size and its application. *Appl Soft Comput* 9(1):39–48
- Clerc M, Kennedy J (2002) The particle swarm-explosion, stability, and convergence in a multidimensional complex space. *IEEE Trans Evol Comput* 6(1):58–73
- Daneshyari M, Yen GG (2011) Cultural-based multiobjective particle swarm optimization. *IEEE Trans Syst Man Cybern B* 41(2):553–567
- Dong W, Zhou M (2014) Gaussian classifier-based evolutionary strategy for multimodal optimization. to appear in *IEEE Trans Neural Netw Learn Syst*
- Eberhart RC, Shi Y (2000) Comparing inertia weights and constriction factors in particle swarm optimization. In: *Proceedings of the 2000 Congress on Evolutionary Computation*, IEEE, vol 1, pp 84–88
- Eberhart R, Kennedy J (1995) A new optimizer using particle swarm theory. In: *Proceedings of the Sixth International Symposium on Micro Machine and Human Science*, 1995, IEEE, pp 39–43
- Fang Y, Chu F, Mammar S, Zhou M (2012) Optimal lane reservation in transportation network. *IEEE Trans Intell Trans Syst* 13(2):482–491
- Ge HW, Sun L, Liang YC, Qian F (2008) An effective PSO and AIS-based hybrid intelligent algorithm for job-shop scheduling. *IEEE Trans Syst Man Cybern A* 38(2):358–368
- Grimaldi EA, Grimaccia F, Mussetta M, Zich R (2004) PSO as an effective learning algorithm for neural network applications. In: *Proceedings of 2004 3rd International Conference on Computational Electromagnetics and Its Applications*, IEEE, pp 557–560
- Holden N, Freitas AA (2005) A hybrid particle swarm/ant colony algorithm for the classification of hierarchical biological data. In: *Proceedings 2005 IEEE Swarm Intelligence Symposium*, IEEE, pp 100–107
- Hsieh ST, Sun TY, Liu CC, Tsai SJ (2009) Efficient population utilization strategy for particle swarm optimizer. *IEEE Trans Syst Man Cybern B* 39(2):444–456

- Jie J, Zeng J, Han C, Wang Q (2008) Knowledge-based cooperative particle swarm optimization. *Appl Math Comput* 205(2):861–873
- Kang Q, Lan T, Yan Y, Wang L, Wu Q (2012a) Group search optimizer based optimal location and capacity of distributed generations. *Neurocomputing* 78(1):55–63
- Kang Q, Zhou M, Xu C (2012b) Solving optimal power flow problems subject to distributed generator failures via particle swarm intelligence. In: 2012 International Conference on Advanced Mechatronic Systems (ICAMechS), IEEE, pp 418–423
- Kang Q, Zhou M, An J, Wu Q (2013) Swarm intelligence approaches to optimal power flow problem with distributed generator failures in power networks. *IEEE Trans Autom Sci Eng* 10(2):343–353. doi:10.1109/TASE.2012.2204980
- Karaboga D, Basturk B (2007) A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. *J Global Optim* 39(3):459–471
- Kennedy J (1997) The particle swarm: social adaptation of knowledge. In: IEEE International Conference on Evolutionary Computation, 1997, IEEE, pp 303–308
- Kennedy J (1999) Small worlds and mega-minds: effects of neighborhood topology on particle swarm performance. In: Proceedings of the 1999 Congress on Evolutionary Computation, CEC 99, IEEE, vol 3
- Kennedy J (2000) Stereotyping: improving particle swarm performance with cluster analysis. In: Proceedings of the 2000 Congress on Evolutionary Computation, IEEE, vol 2, pp 1507–1512
- Kennedy J, Mendes R (2006) Neighborhood topologies in fully informed and best-of-neighborhood particle swarms. *IEEE Trans Syst Man Cybern C* 36(4):515–519
- Kennedy J, Eberhart R (1995) Particle swarm optimization. In: Proceedings of IEEE International Conference on Neural Networks, 1995, IEEE, vol 4, pp 1942–1948
- Kennedy J, Mendes R (2002) Population structure and particle swarm performance. In: Proceedings of the 2002 Congress on Evolutionary Computation, IEEE, vol 2, pp 1671–1676
- Khouadjia MR, Sarasola B, Alba E, Jourdan L, Talbi EG (2012) A comparative study between dynamic adapted PSO and VNS for the vehicle routing problem with dynamic requests. *Appl Soft Comput* 12(4):1426–1439
- Lanzarini L, Leza V, Giusti A (2006) Particle swarm optimization with variable population size. In: Proceedings of the 9th international conference on Artificial Intelligence and Soft Computing, Springer-Verlag, Berlin, pp 438–449
- Leong WF, Yen GG (2008) PSO-based multiobjective optimization with dynamic population size and adaptive local archives. *IEEE Trans Syst Man Cybern B* 38(5):1270–1293
- Li X (2004) Adaptively choosing neighbourhood bests using species in a particle swarm optimizer for multimodal function optimization. In: Genetic and Evolutionary Computation-GECCO 2004, Springer, Berlin, pp 105–116
- Li X (2010) Niching without niching parameters: particle swarm optimization using a ring topology. *IEEE Trans Evol Comput* 14(1):150–169
- Li S, Tan M, Tsang IW, Kwok JY (2011) A hybrid PSO-BFGS strategy for global optimization of multimodal functions. *IEEE Trans Syst Man Cybern B* 41(4):1003–1014
- Li Y, Xiang R, Jiao L, Liu R (2012) An improved cooperative quantum-behaved particle swarm optimization. *Soft Comput* 16(6):1061–1069
- Liang JJ, Qin AK, Suganthan PN, Baskar S (2006) Comprehensive learning particle swarm optimizer for global optimization of multimodal functions. *IEEE Trans Evol Comput* 10(3):281–295
- Liu Y, Lv M, Zuo W (2012) A new multimodal particle swarm optimization algorithm based on greedy algorithm. *Int J Comput Intell Appl* 11(03)
- Madeiro SS, Bastos-Filho CJA, Neto FBL, Figueiredo EMN (2009) Adaptive clustering particle swarm optimization. In: IEEE International Symposium on Parallel & Distributed Processing, pp 1–8
- Mandal S, Kar R, Mandal D, Ghoshal SP (2011) Swarm intelligence based optimal linear phase FIR high pass filter design using particle swarm optimization with constriction factor and inertia weight approach. *Int J Electr Electron Eng* 5(4):296–301
- Marinakis Y, Marinaki M (2013) Particle swarm optimization with expanding neighborhood topology for the permutation flowshop scheduling problem. *Soft Comput* 17(7):1159–1173
- Mendes R, Kennedy J, Neves J (2004) The fully informed particle swarm: simpler, maybe better. *IEEE Trans Evol Comput* 8(3):204–210
- Mendes R, Kennedy J, Neves J (2003) Watch thy neighbor or how the swarm can learn from its environment. In: Proceedings of the 2003 IEEE Swarm Intelligence Symposium, IEEE, pp 88–94
- Montes de Oca M, Aydn D, Stutzle T (2011a) An incremental particle swarm for large-scale continuous optimization problems: an example of tuning-in-the-loop (re)design of optimization algorithms. *Soft Comput* 15(11):2233–2255. doi:10.1007/s00500-010-0649-0
- Montes de Oca M, Stutzle T, Van den Enden K, Dorigo M (2011b) Incremental social learning in particle swarms. *IEEE Trans Syst Man Cybern B* 41(2):368–384
- Nasir M, Das S, Maity D, Sengupta S, Halder U, Suganthan PN (2012) A dynamic neighborhood learning based particle swarm optimizer for global numerical optimization. *Inform Sci* 209:16–36
- Passaro A, Starita A (2006) Clustering particles for multimodal function optimization. In: Proceedings of ECAI Workshop on Evolutionary Computation, pp 124–131
- Rada-Vilela J, Zhang M, Seah W (2013) A performance study on synchronicity and neighborhood size in particle swarm optimization. *Soft Comput* pp 1–12
- Shi Y, Eberhart R (1998a) A modified particle swarm optimizer. In: Evolutionary Computation Proceedings, 1998. IEEE World Congress on Computational Intelligence, IEEE, pp 69–73
- Shi Y, Eberhart RC (1998b) Parameter selection in particle swarm optimization. In: Evolutionary Programming VII, Springer, pp 591–600
- Shi Y, Eberhart RC (2001) Fuzzy adaptive particle swarm optimization. In: Proceedings of the IEEE Congress on Evolutionary Computation 2001, vol 1. IEEE, pp 101–106
- Simon D (2008) Biogeography-based optimization. *IEEE Trans Evol Comput* 12(6):702–713
- Suganthan PN, Hansen N, Liang JJ, Deb K, Chen YP, Auger A, Tiwari S (2005) Problem definitions and evaluation criteria for the CEC 2005 special session on real-parameter optimization. KanGAL, Report 2005005
- Sun J, Fang W, Palade V, Wu X, Xu W (2011) Quantum-behaved particle swarm optimization with gaussian distributed local attractor point. *Appl Math Comput* 218(7):3763–3775
- Wu NQ, Zhou MC (2007) Shortest routing of bidirectional automated guided vehicles avoiding deadlock and blocking. *IEEE/ASME Trans Mechatron* 12(1):63–72
- Xing K, Han L, Zhou M (2012) Deadlock-free genetic scheduling algorithm for automated manufacturing systems based on deadlock control policy. *IEEE Trans Syst Man Cybern B* 42(3):603–615
- Xiong PC, Fan Y, Zhou MC (2009) Web service configuration under multiple quality-of-service attributes. *IEEE Trans Autom Sci Eng* 6(2):311–321
- Yang XS (2010) Firefly algorithm, lévy flights and global optimization. *Research and development in intelligent systems XXVI* pp 209–218
- Yu M, Zhou MC, Su W (2009) A secure routing protocol against Byzantine attacks for MANETs in adversarial environments. *IEEE Trans Veh Technol* 58(1):449–460
- Zhan ZH, Zhang J, Li Y, Chung HH (2009) Adaptive particle swarm optimization. *IEEE Trans Syst Man Cybern B* 39(6):1362–1381
- Zhang C, Yi Z (2011) Scale-free fully informed particle swarm optimization algorithm. *Inform Sci* 181(20):4550–4568