Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2014, Article ID 672412, 12 pages
http://dx.doi.org/10.1155/2014/672412

Research Article

Novel Particle Swarm Optimization and Its Application in Calibrating the Underwater Transponder Coordinates

Zheping Yan, Chao Deng, Benyin Li, and Jiajia Zhou

College of Automation, Harbin Engineering University, Harbin 150001, China

Correspondence should be addressed to Chao Deng; soledc@163.com

Received 25 November 2013; Accepted 3 March 2014; Published 17 April 2014

Academic Editor: P. Karthigaikumar

A novel improved particle swarm algorithm named competition particle swarm optimization (CPSO) is proposed to calibrate the Underwater Transponder coordinates. To improve the performance of the algorithm, TVAC algorithm is introduced into CPSO to present an *extension competition particle swarm optimization* (ECPSO). The proposed method is tested with a set of 10 standard optimization benchmark problems and the results are compared with those obtained through existing PSO algorithms, *basic particle swarm optimization* (BPSO), *linear decreasing inertia weight particle swarm optimization* (LWPSO), *exponential inertia weight particle swarm optimization* (EPSO), and *time-varying acceleration coefficient* (TVAC). The results demonstrate that CPSO and ECPSO manifest faster searching speed, accuracy, and stability. The searching performance for multimodulus function of ECPSO is superior to CPSO. At last, calibration of the underwater transponder coordinates is present using particle swarm algorithm, and novel improved particle swarm algorithm shows better performance than other algorithms.

1. Introduction

Particle swarm optimization (PSO) technique is considered as one of the modern heuristic algorithms for optimization first proposed by Kennedy and Eberhart in 1995 [1]. The motivation for the development of this method was based on the simulation of simplified animal social behaviors [2]. The PSO algorithm works on the social behavior of particles in the swarm. In PSO, the population dynamics simulates a bird flock's behavior where social sharing of information takes place and individuals can profit from the discoveries and previous experience of all other companions during the search for food. That is, the global best solution is found by simply adjusting the trajectory of each individual towards its own best location and towards the best particle of the entire swarm at each time step [1–3]. Owing to its reduction on memory requirement and computational efficiency with convenient implementation, it has gained lots of attention in various optimal control system applications, compared to other evolutionary algorithms [4]. Several researches were carried out so far to analyze the performance of the PSO with different settings; for example, Shi and Eberhart [5] indicated that the optimal solution can be improved by varying the value of ω from 0.9 at the beginning of the search to 0.4 at the end of the search for most problems, and they introduced a method named TVIW with a linearly varying inertia weight over the generations. Chen et al. [6] introduced exponential inertia weight strategies, which is found to be very effective for TVIW. Ratnaweera et al. [2] propose time-varying acceleration coefficients as a parameter automation strategy for the PSO named TVAC, witch reduce the cognitive component and increase the social component, by changing the acceleration coefficients with time. Ni and Deng [7] analyze the performance of PSO with the proposed random topologies and explore the relationship between population topology and the performance of PSO from the perspective of graph theory characteristics in population topologies. Noel [8] presents a new hybrid optimization algorithm that combines the PSO algorithm and gradient-based local search algorithms to achieve faster convergence and better accuracy of final solution without getting trapped in local minima. Epitropakis et al. [9], motivated by the behavior and spatial characteristics of the social and cognitive experience of each particle in the swarm, develop a hybrid framework that combines the particle swarm optimization and the differential

evolution algorithm. In an attempt to efficiently guide the evolution and enhance the convergence, the author evolved the personal experience or memory of the particles with the differential evolution algorithm, without destroying the search capabilities of the algorithm. Mousa et al. [10] propose a hybrid multiobjective evolutionary algorithm combining genetic algorithm and particle swarm optimization; the local search scheme is implemented as a neighborhood search engine to improve the solution quality, where it intends to explore the less-crowded area in the current archive to possibly obtain more nondominated solutions.

As a kind of optimization algorithm, PSO is simple in structure, has good performance, and is easy to implement. It is widely applied in various engineering applications. Moradi and Abedini [11] combined genetic algorithm and particle swarm optimization for optimal location and sizing of distributed generation on distribution systems. The algorithm is to minimize network power losses, make better voltage regulation, and improve the voltage stability within the framework of system operation and security constraints in radial distribution systems. Chang et al. [12] apply the PSO algorithm to estimate the parameters of the Genesio-Tesi nonlinear chaotic systems, and the estimation of the PSO algorithm is verified by examining different sets of random initial populations under the presence of measurement noise. Soon and Low [13] proposed a new approach using particle swarm optimization with inverse barrier constraint to determine the unknown photovoltaic model parameters. The proposed method has been validated with three different photovoltaic technologies. Jiang et al. [14] proposed the barebone particle swarm optimization algorithm to determine the parameters of solid oxide fuel cell (SOPC). The cooperative coevolution strategy is applied to divide the output voltage function into four subfunctions based on the interdependence among variables. To the nonlinear characteristic of SOPC model, a hybrid learning strategy is proposed for BPSO to ensure a good balance between exploration and exploitation. Alfi [15] proposed novel particle swarm optimization, to cope with the online system parameter identification problem. The inertia weight for every particle is dynamically updated based on the feedback taken from the fitness of the best previous position found by the particle, and a novel methodology is incorporated into the novel particle swarm optimization to be able to effectively response and detect any parameter variations of system to be identified. Hu and Shi [16], to solve the premature convergence problem of PSO, improved algorithms with hybrid and mutation operators, leading to obtaining a high level of particle population diversity, decreasing the possibility of falling into local optima, and improving location accuracy. The novel algorithm is introduced in the range-based location for wireless sensor networks and simulation shows a better performance than basic PSO algorithm.

With the development of marine economy and technology, unmanned underwater vehicle (UUV) is an effective means for marine detection, resource exploitation, military interfere, and investigation [17–19]. Navigation of UUV has been and remains a substantial challenge to platforms. One of the main driving factors is the ability to carry out long-duration missions fully autonomously and without supervision from a surface ship [20, 21]. Combined with inertial navigation, the use of one or several transponders on the seabed is an accurate and cost-effective approach toward solving several of these challenges [22–24]. It is obvious that the exact position of the transponder is very important in the underwater transponder positioning system [25, 26]. However, in the practical operations, due to the influence of ocean currents and other factors, the practical coordinates of transponder will drift from the position where it launched into the water. So it requires the mother ship to calibrate the coordinate of the transponder; this paper proposed the particle swarm optimization algorithm solving the transponder coordinates.

The contribution of this paper is concluded as the following. Firstly, considering the competition particle swarm algorithm, each particle will evolve along two different directions to generate two homologous particles. The optimal one is kept through comparing the cost functions of two homologous particles, and the next generation particle will be obtained finally. Secondly, according to the advantage of TVAC, combining CPSO and TVAC, ECPSO algorithm is presented. With a large cognitive component and small social component at the beginning, on the other hand, a small cognitive component and a large social component allow the particles to converge to the global optima in the latter part of the optimization. Simultaneously, the evolution for each particle at any time is along two different inertia directions to generate two homologous particles and to obtain next generation particle. Lastly, ECPSO is introduced to calibrate the coordinate of the transponder.

The rest of this paper is organized as follows. In Section 2, the basic PSO and its previous developments are summarized. In Section 3, the competition particle swarm optimization algorithm and extension competition particle swarm optimization algorithm are introduced. The experimental settings for the benchmark functions and simulation strategies are explained, and the conclusion is drawn based on the comparison analysis. In Section 4, ECPSO is introduced to calibrate the coordinates of the transponder, and simulations are designed to verify the feasibility of the algorithm present.

2. Some Previous Work

Introduced by Dr. Kennedy and Dr. Eberhart in 1995, PSO has ever since turned out to be a competitor in the field of numerical optimization, and there has been a considerable amount of work done in developing the original version of PSO. In this section, we summarize some entire significant previous developments.

2.1. Basic Particle Swarm Optimization (BPSO). In PSO, each solution called a "particle" flies in the search space searching for the optimal position to land. PSO system combines local search method (through individual experience) with global search methods (through neighboring experience), attempting to balance exploration and exploitation [27]. Each particle has a position vector $x_i(k)$, a velocity vector $v_i(k)$, the position with the best fitness encountered by the particle, and

the index of the best particle in the swarm. The position vector and the velocity vector of the ith particle in the d-dimensional search space can be represented as $x_i = (x_{i1}, x_{i2}, x_{i3}, \ldots, x_{id})$ and $v_i = (v_{i1}, v_{i2}, v_{i3}, \ldots, v_{id})$, respectively. The best position of each particle ($p$best) is $p_i = (p_{i1}, p_{i2}, p_{i3}, \ldots, p_{id})$, and the fitness particle found so far at generation k ($g$best) is $p_g = (p_{g1}, p_{g2}, \ldots, p_{gd})$. In each generation, each particle is updated by the following two equations:

$$v_{id}(k+1) = v_{id}(k) + c_1 \times r_1 \times \left(p_{id}(k) - x_{id}(k)\right) + c_2 \times r_2 \times \left(p_{gd}(k) - x_{id}(k)\right), \quad (1)$$

$$x_{id}(k+1) = x_{id}(k) + v_{id}(k+1). \quad (2)$$

The parameters c_1 and c_2 are constants known as acceleration coefficients. r_1 and r_2 are random values in the range from 0 to 1, and the value of r_1 and r_2 is not the same for every iteration. Kennedy and Eberhart [1] suggested setting either of the acceleration coefficients at 2, in order to make the mean of both stochastic factors in (1) unity, so that particles would over fly only half the time of search. The first equation shows that, in PSO, the search toward the optimum solution is guided by the previous velocity, the cognitive component, and the social component.

Since the introduction of the particle swarm optimization, numerous variations of the algorithm have been developed in the literature. Eberhart and Shi showed that PSO searches for wide areas effectively but tends to lack local search precision. They proposed in that work a solution by introducing ω, an inertia factor. In this paper, we name it as basic particle swarm optimization (BPSO):

$$v_{id}(k+1) = \omega \times v_{id}(k) + c_1 \times r_1 \times \left(p_{id}(k) - x_{id}(k)\right) + c_2 \times r_2 \times \left(p_{gd}(k) - x_{id}(k)\right), \quad (3)$$

$$x_{id}(k+1) = x_{id}(k) + v_{id}(k+1).$$

2.2. Time-Varying Inertia Weight (TVIW). The role of the inertia weight ω is considered very important in PSO convergence behavior. The inertia weight is applied to control the impact of the previous history of velocities on the current velocity. large inertia weight facilitates global exploration, while small one tends to facilitate local exploration. In order to assure that the particles converge to the best point in the course of the search, Shi and Eberhart [28] have found that *time-varying inertia weight* (TVIW) has a significant improvement in the performance of PSO and proposed *linear decreasing inertia weight PSO* (LWPSO) with a linear decreasing value of ω. This modification can increase the exploration of the parameter space during the initial search iterations and increase the exploitation of the parameter space during the final steps of the search [29]. The mathematical representation of inertia weight is given as follows:

$$\omega = (\omega_1 - \omega_2) \times \left(\frac{\text{MAXITER} - k}{\text{MAXITER}}\right) + \omega_2, \quad (4)$$

where ω_1 and ω_2 are the initial and final values of the inertia weight, respectively, k is the current iteration number, and MAXITER is the maximum number of allowable iterations. Shi and Eberhart [5] indicate that the optimal solution can be improved by varying the value of ω from 0.9 at the beginning of the search to 0.4 at the end of the search for most problems.

Chen et al. [6] proposed natural exponential (base e) inertia weight strategies, named EPSO and expressed as

$$\omega = \omega_2 + (\omega_1 - \omega_2) \times \exp\left[-\left(\frac{K}{(\text{MAXITER}/4)}\right)^2\right]. \quad (5)$$

2.3. Time-Varying Acceleration Coefficient (TVAC). In PSO, the particle was updated due to the cognitive component and the social component. Therefore, proper control of these two components is very important to find the optimum solution accurately and efficiently. Ratnaweera et al. [2] introduced a time-varying acceleration coefficient (TVAC), which reduces the cognitive component and increases the social component, by changing the acceleration coefficients c_1 and c_2 with the time evolution. The objective of this development is to enhance the global search in the early part of the optimization and to encourage the particles to converge toward the global optima at the end of the search. The TVAC is represented using the following equations:

$$c_1 = (c_{\max} - c_{\min})\frac{k}{\text{MAXITR}} + c_{\min}, \qquad c_2 = (c_{\min} - c_{\max})\frac{k}{\text{MAXITR}} + c_{\max}, \quad (6)$$

where $c_{\min}$ and $c_{\max}$ are constants, k is the current iteration number, and MAXITR is the maximum number of allowable iterations.

Simulations were carried out with numerical benchmarks, to find out the best ranges of values for c_1 and c_2. From the results it was observed that the best solutions were determined when changing c_1 from 2.5 to 0.5 and changing c_2 from 0.5 to 2.5 over the full search range.

3. Proposed New Developments

It is clarified from (1) that particle's new velocity is correlated with three terms: the particle's previous velocity, the value of the cognitive component, and the value of the social component. Therefore, proper control method with inertia weight factor and acceleration coefficients is significant to find the optimum solution accurately and efficiently.

The inertia weight is utilized to adjust the influence of the previous velocity on the current velocity and balance between global and local exploration abilities of the "flying particle" [30, 31]. A larger inertia weight implies stronger global exploration ability, advocating the particle to escape from a local minimum. A smaller inertia weight leads to stronger local exploration ability, confining the particle searching within a local range near its present position to guarantee the convergence.

Kennedy and Eberhart [1] indicated that a relatively high value of the cognitive component, compared with the social component, will result in excessive wandering of individuals

through the search space. In contrast, a relatively high value of the social component may lead particles to rush prematurely toward a local optimum.

Considering those concerns, we propose a new strategy for the PSO concept.

3.1. Competition Particle Swarm Optimization (CPSO). In the process of particle evolution, each particle is evolved along different directions with different inertia coefficients and acceleration coefficients. Two homologous particles are generated and the optimal one is kept through comparing cost functions of two homologous particles, eliminating the inferior one. Then, the next generation particle is updated finally. The evaluation function of each particle is described as

$$\begin{aligned} v_{id}^{t}(k+1) &= \omega^{t} v_{id}(k) + c_1^{t} r_1^{t}\left(p_{id}(k) - x_{id}(k)\right) \\ &\quad + c_2^{t} r_2^{t}\left(p_{gd}(k) - x_{id}(k)\right), \\ x_{id}^{t}(k+1) &= x_{id}^{t}(k) + v_{id}^{t}(k+1). \end{aligned} \tag{7}$$

And the final equations are shown as

$$\begin{aligned} v_{id}(k) &= \left\{ v_{id}^{t}(k) \mid \min\left(\text{fitness}\left(x_i^{t}(K)\right)\right),\ t = 1,2 \right\}, \\ x_{id}(k) &= \left\{ x_{id}^{t}(k) \mid \min\left(\text{fitness}\left(x_i^{t}(K)\right)\right),\ t = 1,2 \right\}, \end{aligned} \tag{8}$$

where n is the number of particles in the swarm, M is the maximum iteration frequency, r_1^t, r_2^t are the random numbers in the range of $[0, 1]$, $x_{id}(k)$ is the ith particle position in the dth dimension after k time iteration, $x_{id}^t(k)$ shows the dth dimension position for subparticle t of the ith particle after k time iteration, ω^t is the speed inertia weight of subparticle t, c_1^t and c_2^t are constants denoting acceleration coefficients, fitness $(x_i^t(K))$ is the fitting function of the subparticle t after k time iteration, $v_{id}^t(k)$ is the speed of subparticle t at dth dimension, $p_{id}(k)$ is the optimal position of the ith particle at dth dimension, and $p_{gd}(k)$ is the swarm optimal position at dth dimension after k time iteration.

Remark 1. In this paper, two subparticles are generated for each particle at one time; therefore, $t = 1, 2$.

The detailed steps are shown as follows.

Step 1. Initial.

Substep 1. Set the initial parameters n, M, w^t, c_1^t, c_2^t.

Substep 2. Take random initial $x_{id}(0)$.

Substep 3. Take random initial $v_{id}^t(1)$.

Substep 4. Calculate fitness$(x_i^t(0))$ and set $p_i(0) = x_i(0)$.

Substep 5. One has $p_g(0) = \{x_i(0) \mid x_i(0) \in \min(\text{fitness}(x_i(0)))\}$.

Step 2. If the criteria are satisfied, output the best solution; otherwise, go to Substep 6.

Substep 6. Update $v_{id}^t(k)$ and $x_{id}^t(k)$.

Substep 7. Calculate fitness$(x_i^t(k))$.

Substep 8. One has $x_{id}(k) = \{x_{id}^t(k) \mid \min(\text{fitness}(x_i^t(k))), t = 1, 2\}$.

Substep 9. If fitness$(x_i(k))$ < fitness$(p_i(k))$

$$p_i(k) = x_i(k). \tag{9}$$

If $\min\{\text{fitness}(x_i(k)), i = 1, \ldots, n\} < \text{fitness}(p_g(k))$

$$p_g(k) = \left\{ x_i(k) \mid \min\left[\text{fitness}\left(x_i(k)\right),\ i = 1, \ldots, n\right] \right\}. \tag{10}$$

Substep 10. Go back to Step 2.

3.2. Extension Competition Particle Swarm Optimization (ECPSO). competition particle swarm optimization (CPSO) helps adjust the search direction particles and improve the search speed and efficiency, but, due to rapid convergence, CPSO is easy to fall into local minima. According to benchmark functions simulation in Section 4, it is obvious that CPSO is superior to the searching effective of single-modulus function, and TVAC is superior to the searching effective of multimodulus function. The reason resulting in this phenomenon is due to the selection of acceleration coefficient. With a large cognitive component and a small social component at the beginning, particles are allowed to move around the search space, instead of moving toward the population best. On the other hand, a small cognitive component and a large social component allow the particles to converge to the global optima in the latter part of the optimization. Considering the advantage of TVAC, introduce TVAC into CPSO and the extension competition particle swarm optimization (ECPSO) with the acceleration coefficients proposed as above. The evolution equations can be mathematically represented as the following:

$$\begin{aligned} v_{id}^{t}(k+1) &= \omega^{t} v_{id}(k) + c_1 r_1^{t}\left(p_{id}(k) - x_{id}(k)\right) \\ &\quad + c_2 r_2^{t}\left(p_{gd}(k) - x_{id}(k)\right), \\ x_{id}^{t}(k+1) &= x_{id}^{t}(k) + v_{id}^{t}(k+1), \end{aligned} \tag{11}$$

where

$$\begin{aligned} c_1 &= \left(c_{\max} - c_{\min}\right)\frac{k}{\text{MAXITR}} + c_{\min}, \\ c_2 &= \left(c_{\min} - c_{\max}\right)\frac{k}{\text{MAXITR}} + c_{\max}. \end{aligned} \tag{12}$$

And it is obvious that

$$\begin{aligned} v_{id}(k) &= \left\{ v_{id}^{t}(k) \mid \min\left(\text{fitness}\left(x_i^{t}(K)\right)\right),\ t = 1,2 \right\}, \\ x_{id}(k) &= \left\{ x_{id}^{t}(k) \mid \min\left(\text{fitness}\left(x_i^{t}(K)\right)\right),\ t = 1,2 \right\}. \end{aligned} \tag{13}$$

4. Experimental Settings and Simulation Strategies for Benchmark Testing

Simulations were carried out to observe the rate of convergence and the quality of the optimum solution of the new methods introduced in this investigation by comparing with BPSO, EPSO, and TVAC. From the standard set of benchmark problems available in the literature, there are 5 important functions considered to test the efficacy of the proposed method. All of the benchmark functions reflect different degrees of complexity.

4.1. Functions Introduction. The functions are as follows.

(1) Sphere function: one has

$$f_1(x) = \sum_{i=1}^{D} x_i^2. \tag{14}$$

With the search space $\{x_i \mid -100 < x_i < 100\}$, the global minimum locates at $x = [0, \ldots, 0]^D$ with $f(x) = 0$. It is very simple, convex, and unimodal function with only one local optimum value.

(2) Axis parallel hyperellipsoid function: one has

$$f_2(x) = \sum_{i=1}^{D} i \cdot x_i^2. \tag{15}$$

It is known as the weighted sphere model. With the search space $\{x_i \mid -100 < x_i < 100\}$, the global minimum locates at $x = [0, \ldots, 0]^D$ with $f(x) = 0$. It is continuous, convex, and unimodal.

(3) Rotated hyperellipsoid function (Schwefel's problem 1.2): one has

$$f_3(x) = \sum_{i=1}^{D} \left(\sum_{j=1}^{i} x_j \right)^2. \tag{16}$$

With the search space $\{x_i \mid -100 < x_i < 100\}$, the global minimum locates at $x = [0, \ldots, 0]^D$ with $f(x) = 0$. It is continuous, convex, and unimodal. With respect to the coordinate axes, this function produces rotated hyperellipsoids.

(4) Moved axis parallel hyperellipsoid function: one has

$$f_4(x) = \sum_{i=1}^{D} 5i \cdot x_i^2. \tag{17}$$

With the search space $\{x_i \mid -100 < x_i < 100\}$, the global minimum locates at $x(i) = 5 * i, i = 1 : D$, with $f(x) = 0$.

(5) Rosenbrock function: one has

$$f_5(x) = \sum_{i=1}^{D-1} \left[100\left(x_{i+1} - x_i^2\right)^2 + \left(x_i - 1\right)^2 \right]. \tag{18}$$

Table 1: Parameters for simulation.

Method	Parameters	
CPSO	$\omega^1 = \omega_{\max} = 0.9, \omega^2 = \omega_{\min} = 0.4,$ $c_1^1 = 0.5, c_2^1 = 2.5, c_1^2 = 2.5,$ $c_2^2 = 0.5$	
ECPSO	$\omega^1 = \omega_{\max} = 0.9, \omega^2 = \omega_{\min} = 0.4,$ $c_{\min} = 0.5, c_{\max} = 2.5$	$n = 30,$ $D = 10,$ $x_i \in [-100, 100],$ $v_i \in [-100, 100]$
BPSO	$c_1 = c_2 = 2.0, \omega = 0.7$	
EPSO	$c_1 = c_2 = 2.0, \omega_{\max} = 0.9,$ $\omega_{\min} = 0.4$	
TVAC	$c_{\min} = 0.5, c_{\max} = 2.5, \omega = 0.7$	

With the search space $\{x_i \mid -100 < x_i < 100\}$, the global minimum locates at $x = [1, \ldots, 1]^D$ with $f(x) = 0$. It is a unimodal function and the global optimum is inside a long, narrow, parabolic shaped flat valley. To find the valley is trivial.

(6) Rastrigin function: one has

$$f_6(x) = \sum_{i=1}^{D} \left[x_i^2 - 10\cos\left(2\pi x_i\right) + 10 \right]. \tag{19}$$

With the search space $\{x_i \mid -100 < x_i < 100\}$, the global minimum locates at $x = [0, \ldots, 0]^D$ with $f(x) = 0$. It is highly multimodal. However, the locations of the minima are regularly distributed.

(7) Griewank function: one has

$$f_7(x) = \frac{1}{4000} \sum_{i=1}^{D} x_i^2 - \prod_{i=1}^{D} \cos\left(\frac{x_i}{\sqrt{i}} \right) + 1. \tag{20}$$

With the search space $\{x_i \mid -100 < x_i < 100\}$, the global minimum locates at $x = [0, \ldots, 0]^D$ with $f(x) = 0$. It is a multimodal function and has many widespread local minima. However, the locations of the minima are regularly distributed.

(8) Sum of different power function: one has

$$f_8(x) = \sum_{i=1}^{D} \left| x_i \right|^{(i+1)}. \tag{21}$$

The sum of different powers is a commonly used unimodal test function. With the search space $\{x_i \mid -100 < x_i < 100\}$, the global minimum locates at $x = [0, \ldots, 0]^D$ with $f(x) = 0$.

(9) Ackley's path function: one has

$$f_9(x) = -a \cdot e^{-b \cdot \sqrt{\sum_{i=1}^{D} x_i^2 / D}} - e^{\sum_{i=1}^{D} \cos(c \cdot x_i)/D} + a + e^1. \tag{22}$$

Table 2: Comparison of BPSO, WPSO, EPSO, TVAC, and CPSO.

F	D	Average (standard deviation)					
		BPSO	LWPSO	EPSO	TVAC	CPSO	ECPSO
f_1	10	4.7979$e-11$ (1.1343$e-10$)	3.2219$e-24$ (1.0706$e-23$)	8.7747$e-51$ (3.408$e-50$)	1.4826$e-54$ (1.4770$e-53$)	4.1023$e-146$ (3.6388$e-145$)	4.1221$e-118$ (3.0916$e-117$)
	30	5.3768 (4.3459)	4.0137$e-05$ (5.926$e-05$)	2.3219$e-12$ (1.1495$e-11$)	1.1062$e-10$ (7.3302$e-10$)	1.4872$e-46$ (6.4922$e-46$)	1.6829$e-10$ (1.0243$e-09$)
	50	2.8412$e+01$ (4.5047)	0.2149 (0.1634)	2.2215$e-05$ (3.5696$e-05$)	0.0019 (0.0079)	9.3999$e-20$ (4.0558$e-19$)	0.0022 (0.0083)
	70	4.4686$e+01$ (4.3454)	1.1728$e+01$ (1.2113$e+01$)	0.0192 (0.0238)	0.0609 (0.1033)	2.0133$e-07$ (1.9424$e-06$)	0.0302 (0.0458)
f_2	10	3.7589$e-10$ (1.7479$e-09$)	8.5629$e-24$ (1.9031$e-23$)	5.2910$e-50$ (2.3037$e-49$)	1.5947$e-52$ (1.4618$e-51$)	8.9632$e-146$ (3.5734$e-145$)	3.0640$e-113$ (3.0622$e-112$)
	30	37.8269 (31.3869)	0.0005 (0.0007)	1.0278$e-11$ (2.0979$e-11$)	1.1937$e-09$ (6.5345$e-09$)	8.1119$e-45$ (4.4882$e-44$)	3.0047$e-10$ (2.1285$e-09$)
	50	6.8495$e+02$ (1.0535$e+02$)	2.8041 (2.2863)	0.0004 (0.0010)	0.1214 (0.6006)	2.4666$e-15$ (2.4614$e-14$)	0.0223 (0.0835)
	70	1.4952$e+03$ (1.8073$e+02$)	1.2961$e+02$ (1.4505$e+02$)	0.2601 (0.2405)	2.1146 (3.9189)	1.2547$e-05$ (0.0001)	1.1879 (2.0240)
f_3	10	0.0013 (0.0018)	4.0624$e-08$ (1.1066$e-07$)	3.1542$e-15$ (1.9686$e-14$)	3.6072$e-27$ (1.6376$e-26$)	6.0846$e-60$ (5.8552$e-59$)	4.4092$e-51$ (3.1687$e-50$)
	30	2.0647$e+01$ (7.9831)	3.7596 (2.0255)	0.5809 (0.3373)	0.0091 (0.0197)	2.7957$e-06$ (5.0455$e-06$)	0.0258 (0.0385)
	50	7.2983$e+01$ (2.8263$e+01$)	2.6189$e+01$ (1.3125$e+01$)	1.0295$e+01$ (5.1265)	0.4883 (0.3021)	0.0775 (0.0620)	0.5720 (0.3509)
	70	1.4821$e+02$ (4.6282$e+01$)	6.9337$e+01$ (3.3718$e+01$)	2.9336$e+01$ (1.0417$e+01$)	2.7039 (1.0853)	1.1214 (0.4917)	2.2342 (0.7423)
f_4	10	2.7409$e-09$ (1.5090$e-08$)	1.3708$e-22$ (6.5937$e-22$)	3.5012$e-49$ (2.8064$e-48$)	9.9156$e-52$ (6.9209$e-51$)	1.6140$e-144$ (1.4296$e-143$)	3.0561$e-115$ (2.1383$e-114$)
	30	1.9825$e+02$ (1.7202$e+02$)	0.0025 (0.0029)	7.3319$e-11$ (2.3165$e-10$)	6.6962$e-09$ (3.6744$e-08$)	1.0828$e-42$ (8.9052$e-42$)	5.6960$e-08$ (3.9998$e-07$)
	50	3.2922$e+03$ (5.6799$e+02$)	1.3397$e+01$ (9.4756)	0.0015 (0.0019)	0.3064 (1.2614)	2.2477$e-17$ (1.4523$e-16$)	0.2379 (1.1928)
	70	7.4756$e+03$ (7.5131$e+02$)	6.8348$e+02$ (6.1444$e+02$)	1.6799 (2.9749)	1.0304$e+01$ (1.4839$e+01$)	1.9407$e-06$ (9.6092$e-06$)	5.5072 (1.0286$e+01$)
f_5	10	5.6418 (1.092)	4.3037 (1.2401)	3.1701 (1.2637)	0.7262 (0.9490)	0.6503 (1.4786)	0.6445 (1.4706)
	30	1.2322$e+03$ (9.4694$e+02$)	4.2313$e+01$ (2.6596$e+01$)	3.1052$e+01$ (1.7699$e+01$)	2.3370$e+01$ (1.9238)	1.4702$e+01$ (3.1506)	1.9234$e+01$ (2.9573)
	50	6.5575$e+03$ (1.3919$e+03$)	2.9055$e+02$ (1.6229$e+02$)	7.3894$e+01$ (3.6633$e+01$)	4.6563$e+01$ (2.5861)	3.74782$e+01$ (2.7904)	4.7741$e+01$ (10.6763)
	70	1.1834$e+04$ (1.9184$e+03$)	5.2635$e+03$ (4.0246$e+03$)	2.0414$e+02$ (7.4873$e+01$)	7.5240$e+01$ (1.2063$e+01$)	6.5838$e+01$ (1.5847$e+01$)	8.4068$e+01$ (3.0477$e+01$)
f_6	10	5.1090 (3.0667)	3.6806 (1.7685)	3.8703 (1.7882)	0.9750 (0.9693)	2.2685 (1.3272)	0.6765 (0.8231)
	30	1.5696$e+02$ (3.9880$e+01$)	3.2291$e+01$ (1.0059$e+01$)	2.9949$e+01$ (7.1988)	1.7282$e+01$ (5.4892)	1.8148$e+01$ (6.2546)	1.3312$e+01$ (4.4942)
	50	3.8006$e+02$ (3.5258$e+01$)	7.5141$e+01$ (2.587$e+01$)	6.1896$e+01$ (1.5107$e+01$)	3.6575$e+01$ (9.4137)	3.6236$e+01$ (9.1136)	3.0739$e+01$ (8.5709)
	70	5.8512$e+02$ (3.6592$e+01$)	1.6259$e+02$ (6.2498$e+01$)	8.8994$e+01$ (1.7552$e+01$)	5.4198$e+01$ (1.1482$e+01$)	5.2752$e+01$ (1.2361$e+01$)	4.9043$e+01$ (1.1607$e+01$)
f_7	10	9.8646$e-05$ (0.0009)	0.0041 (0.0096)	0.0017 (0.0055)	0 (0)	0 (0)	0 (0)
	30	0.1366 (0.1090)	0.0008 (0.0045)	0.0013 (0.0071)	1.0210$e-10$ (1.0021$e-09$)	3.5527$e-17$ (1.4708$e-16$)	1.7067$e-11$ (1.3918$e-10$)
	50	0.5856 (0.0891)	0.0027 (0.0021)	9.9034$e-05$ (0.0009)	0.0001 (0.0007)	2.4547$e-15$ (8.8945$e-15$)	3.6456$e-05$ (0.0001)
	70	0.6941 (0.0646)	0.0329 (0.0190)	0.0002 (0.0010)	0.0011 (0.0017)	4.4137$e-11$ (2.7956$e-10$)	0.0007 (0.0012)

Table 2: Continued.

F	D	Average (standard deviation)					
		BPSO	LWPSO	EPSO	TVAC	CPSO	ECPSO
f_8	10	5.8516$e-18$ (2.5897$e-17$)	2.2299$e-40$ (1.6366$e-39$)	2.0832$e-84$ (1.6585$e-83$)	2.5229$e-87$ (1.1045$e-86$)	5.8546$e-237$ (0)	2.2665$e-154$ (2.2664$e-153$)
	30	1.9895$e+02$ (6.2708$e+02$)	1.0418$e-06$ (3.7845$e-06$)	7.3163$e-18$ (2.7837$e-17$)	1.0483$e-39$ (7.2123$e-39$)	3.0968$e-99$ (3.0467$e-98$)	1.0029$e-41$ (1.0029$e-40$)
	50	1.9927$e+07$ (7.1603$e+07$)	1.8348$e+01$ (4.8704$e+01$)	0.0017 (0.0067)	6.5668$e-25$ (3.1648$e-24$)	1.5985$e-47$ (1.4843$e-46$)	3.7443$e-23$ (3.7143$e-22$)
	70	1.5637$e+13$ (8.3656$e+13$)	2.6886$e+08$ (1.6837$e+09$)	1.4973$e+04$ (1.0807$e+05$)	4.9509$e-19$ (2.6306$e-18$)	3.2094$e-29$ (2.5130$e-28$)	1.9384$e-19$ (1.0414$e-18$)
f_9	10	0.0472 (0.1618)	0.1115 (0.2369)	0.1773 (0.2354)	5.0448$e-15$ (1.3412$e-15$)	0.1269 (0.2045)	0.0132 (0.0770)
	30	1.5006 (0.2835)	0.5798 (0.0888)	0.5469 (0.0765)	0.3709 (0.0736)	0.4187 (0.0747)	0.3822 (0.0767)
	50	1.8625 (0.1067)	0.6039 (0.1209)	0.5348 (0.0655)	0.3691 (0.0544)	0.3848 (0.0541)	0.3631 (0.0478)
	70	1.8765 (0.0728)	0.8228 (0.2349)	0.4790 (0.0512)	0.3439 (0.0463)	0.3369 (0.0469)	0.3393 (0.0413)
f_{10}	10	9.4158$e-13$ (4.9298$e-12$)	1.4531$e-26$ (3.9844$e-26$)	1.4997$e-33$ (3.4384$e-48$)	1.4997$e-33$ (3.4384$e-48$)	1.4997$e-33$ (3.4384$e-48$)	2.5554$e-33$ (1.0557$e-32$)
	30	0.0527 (0.0668)	3.4402$e-07$ (5.4202$e-07$)	9.9564$e-15$ (2.0677$e-14$)	6.7912$e-12$ (4.6051$e-11$)	3.3310$e-31$ (1.4574$e-30$)	8.3251$e-13$ (4.2490$e-12$)
	50	1.3060 (0.4239)	0.0010 (0.0008)	1.8317$e-07$ (3.4350$e-07$)	0.0003 (0.0015)	0.0018 (0.0127)	0.0011 (0.0091)
	70	2.6961 (0.4601)	0.0350 (0.0393)	0.0011 (0.0091)	0.0059 (0.0115)	0.0009 (0.0090)	0.0023 (0.0094)

With the search space $\{x_i \mid -100 < x_i < 100\}$, the global minimum locates at $x = [0, \ldots, 0]^D$ with $f(x) = 0$.

Set the coefficients as $a = 20$, $b = 0.2$, $c = 2 \cdot \pi$.

(10) Penalised function: one has

$$f_{10}(x) = 0.1 \left\{ \sin^2(3\pi x_1) + \sum_{i=1}^{D-1} (x_i - 1)^2 \left[1 + \sin^2(3\pi x_1)\right] + (x_D - 1)^2 \left[1 + \sin^2(2\pi x_D)\right] \right\} + \sum_{i=1}^{D} u(x_i, 5, 100, 4), \tag{23}$$

where

$$u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m, & x_i > a, \\ 0, & -a \le x_i \le a, \\ k(-x_i - a)^m, & x_i < -a, \end{cases} \tag{24}$$

$$y_i = 1 + \frac{1}{4}(x_i + 1).$$

With the search space $\{x_i \mid -100 < x_i < 100\}$, the global minimum locates at $x = [1, \ldots, 1]^D$ with $f(x) = 0$.

4.2. The Coefficients Setting. The parameters for simulation are listed in Table 1.

In this table, $c_{(\cdot)}$ expresses the accelerations coefficients, ω denotes inertia weight, the dimension is D, and the range of the search space and the velocity space are $x_{(\cdot)}$ and $v_{(\cdot)}$. If the current position is out of the search space, the position of the particle is taken to be the value of the boundary, and the velocity is taken to be zero. If the velocity of the particle is outside of the boundary, its value is set to be the boundary value. The maximum number of iterations is set to 1000. For each function, 100 trials were carried out and the average optimal value and the standard deviation are presented. To verify the performance of the algorithm at different dimensions, variable dimension D increases from 10 to 100, and the optimal mean and variance of benchmark functions are calculated. The results are presented in Table 2.

5. The Results in Comparison with the Previous Developments

The simulation results are given in Table 2. The comparison results elucidate that the searching accuracy and stability ranging from low to high are listed as BPSO, LWPSO, EPSO, TVAC, ECPSO, and CPSO for unimodal function. It is obvious that the performances of ECPSO and CPSO are superior due to their advantage of obtaining the optimal speed direction and the searching efficiency, while, in the multimodal function, the CPSO algorithm is easy to trap into local minimum, and TVAC shows better performance than CPSO. Combining the advantages of CPSO and TVAC, the EPSO algorithm is applied to enhance the global search in

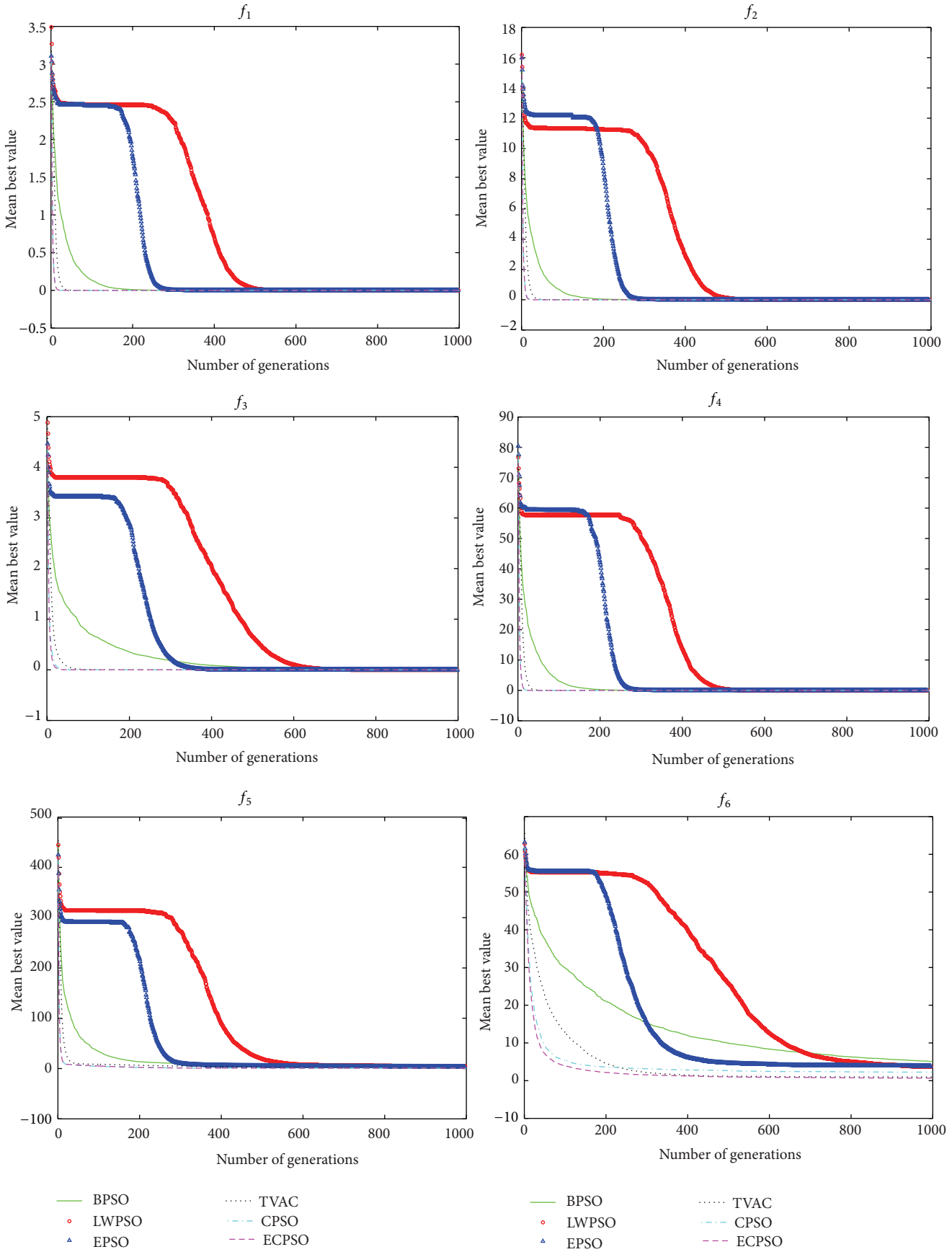

Figure 1: Continued.

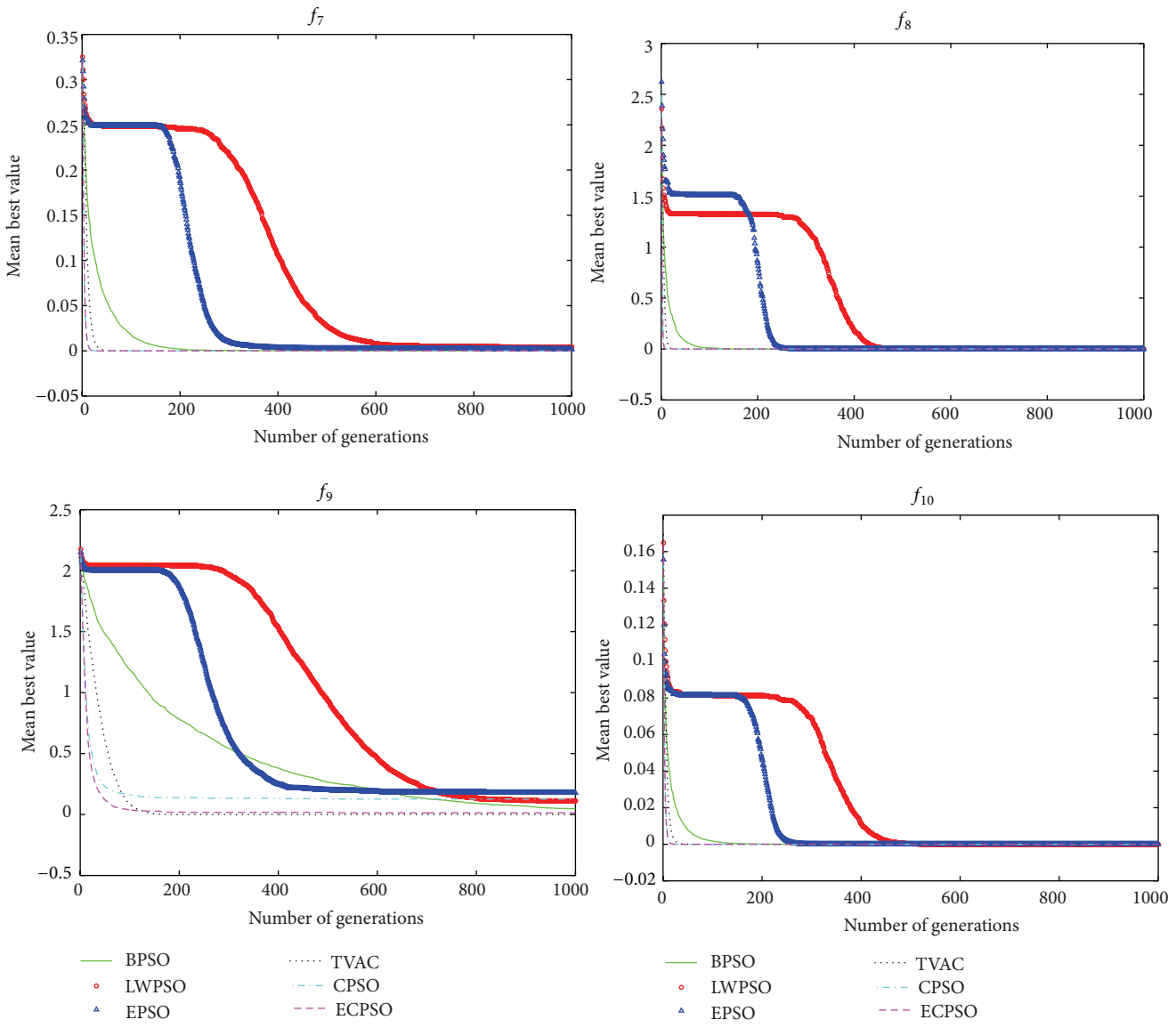

Figure 1: Variation of the average optimum value with time.

the early part of the optimization and encourage the particles to converge toward the global optima at the end of the search. Comparing to TVAC and CPSO, the proposed new algorithm ECPSO is appropriate for multimodal function search.

With the increase of benchmark functions dimension, the searching accuracy and stability for each algorithm are decreased. The performances of CPSO and ECPSO are superior to the other algorithms. With regard to multimodal function, the performances of ECPSO and TVAC are superior to CPSO. The average fitness is varied as in Table 2. From Figure 1, it shows that the order of searching speed from high to low is BPSO (green solid curve), LWPSO (red circle), EPSO (blue triangle), TVAC (black dot curve), CPSO (blue dash-dot), and ECPSO (purples dash). It is obvious that the performances of EPSO and CPSO are more effective than the other algorithms.

6. Calibrate the Underwater Transponder Coordinates

After two decades of dedicated research and development, unmanned underwater vehicles (UUV) have been accepted by an increasing number of users in both military and civilian institutions. The design and implementation of navigation systems stand out as one of the most critical steps towards the successful operation of autonomous vehicles. The quality of the overall estimates of the navigation system dramatically influences the capability of the vehicles to perform precision-demanding tasks [32]. From a navigation point of view, a single range transponder may be regarded as an underwater lighthouse providing UUV with the ranges relative to its fixed geographical location. Single transponder navigation is not a new concept. As mentioned, the first at-sea demonstration of

Table 3: The calibration of the underwater transponder coordinates.

	BPSO	LWPSO	EPSO	TVAC	CPSO	ECPSO
$e1$	$3.5371e-15$	$-2.3573e-15$	$2.6542e-15$	$-1.9813e-15$	$-1.1700e-15$	$1.4526e-15$
$e2$	$1.5510e-15$	$2.3280e-15$	$-4.8804e-16$	$-1.4289e-15$	$1.3702e-15$	$2.2549e-15$
$e3$	0	0	0	0	0	0
sq	$4.1948e-15$	$2.7057e-15$	$2.6986e-15$	$2.4428e-15$	$1.8018e-15$	$2.6823e-15$

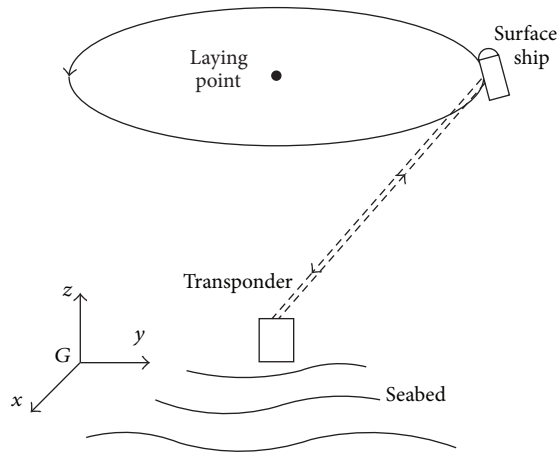

Figure 2: The calibration geometry for calibration of the transponder position.

single transponder UTP aided inertial navigation was carried out in 2003, as described in [33].

For ranging techniques like UTP to work, the geographical location of the transponders must be known. The preferred method is to measure the position directly using USBL on a surface ship. When the transponder deployment is completed, a surface ship with USBL and GPS sails around the transponder in a circular motion, collecting surface ship position and distance information. The calibration geometry is illustrated in Figure 2.

In order to set the design framework, let $\{G\}$ denote the global coordinate frame, and let $\{B\}$ denote a coordinate frame attached to the vehicle, usually denominated as body-fixed coordinate frame. The frame $\{O\}$ is the transceiver coordinates. The position of transponder in the global coordinate frame is given by

$$^{G}X_{T} = {}^{G}X_{D} - {}^{G}_{B}R\,{}^{B}X_{D} + {}^{G}_{B}R\left({}^{B}_{O}R\,{}^{O}X_{T} + {}^{B}X_{O}\right), \quad (25)$$

where $^{G}X_{D}$ is the position of the GPS in global coordinates, $^{B}X_{D}$ is the position of the GPS in vehicle coordinates, $^{O}X_{T}$ is the position of the transponder in transceiver coordinates, $^{B}X_{O}$ is the position of the transceiver in vehicle coordinates, $^{G}_{B}R$ is the rotation matrix from $\{B\}$ to $\{G\}$, and $^{B}_{O}R$ is the rotation matrix from $\{O\}$ to $\{B\}$.

After the data collection, we can get the distance between the transponder and transceiver and the corresponding GPS position in the global coordinates. The attitude of the vehicle can be measured by the heading sensors and the pitch/roll sensor, so $^{G}_{B}R$ is known. We also know $^{B}X_{D}$, $^{B}X_{O}$, and $^{B}_{O}R$.

According to (25) we have

$$^{O}X_{T} = {}^{B}_{O}R^{-1}\left({}^{G}_{B}R^{-1}\left({}^{G}X_{T} - {}^{G}X_{D}\right) + {}^{B}X_{D} - {}^{B}X_{O}\right). \quad (26)$$

We denote $^{O}X_{Ti} = \left[{}^{O}x_{Ti} \;\; {}^{O}y_{Ti} \;\; {}^{O}z_{Ti}\right]^{T}$ $(i = 1, 2, \ldots, N)$ as the position of the transponder in the transceiver coordinates in the ith measurement point. The cost function becomes

$$F = \sum_{i=1}^{N} \frac{\left(\sqrt{{}^{O}x_{Ti}^{2} + {}^{O}y_{Ti}^{2} + {}^{O}z_{Ti}^{2}} - L_{i}\right)^{2}}{N}. \quad (27)$$

Particles optimization algorithms have been introduced. Defining each particle as a coordinate of the transponder. The parameters of particle swarm optimization for simulation are the same as in Section 4. And a surface ship moves in a circular motion with a radius of 100; the real coordinates of transponder are $^{G}X_{rT} = \left[0 \;\; 0 \;\; 100\right]^{T}$. To simplify the problem, $^{G}_{B}R$ is unit matrix, and $^{B}X_{D} - {}^{B}X_{O} = \left[0 \;\; 0 \;\; 0\right]^{T}$, and ignore the sensor measurement error. The simulation data is shown in Table 3. Obviously, the particle swarm algorithm can search for the transponder coordinates and obtain accurate results. At the same time, CPSO shows the best performance.

Remark 2. One has

$$\begin{aligned} e1 &= {}^{G}x_{T} - {}^{G}x_{rT}, \qquad e2 = {}^{G}y_{T} - {}^{G}y_{rT}, \\ e3 &= {}^{G}z_{T} - {}^{G}z_{rT}, \qquad sq = \sqrt{e1^{2} + e2^{2} + e3^{2}}. \end{aligned} \quad (28)$$

7. Conclusion

In this paper, a novel strategy to improve the performance of particle swarm optimization is proposed to apply in calibration of the underwater transponder coordinates. To improve the population-based search optimization algorithm, each particle is evolved along two different directions to generate two homologous particles. The cost functions of two homologous particles are calculated to keep the optimal one and to eliminate the poor one. Then the next generation particle is updated. It is regarded as CPSO. Ten classify benchmark functions are introduced to reflect the effectiveness of the proposed algorithm. The simulation results demonstrate that the unimodal function, CPSO algorithm, is superior to BPSO, LWPSO, EPSO, and TVAC on the searching accuracy, stability, and convergence speed. However, considering the multimodal function, the performance of TVAC is superior to CPSO.

Secondly, to further improve the performance of CPSO, the ECPSO is proposed by combining the CPSO and the TVAC. In the initial period of the evolution, the individual experience is a significant aspect with larger acceleration coefficient, and in the final period, the swarm experience is superior with a greater acceleration coefficient. Simultaneously, the evolution for each particle at any time is towards two different inertia directions to generate two homologous particles and to obtain its next generation particles. The simulations show the effectiveness of multimodal function by using ECPSO. With the incensement of benchmark functions dimensions, the accuracy and stability of each algorithm will decrease, but CPSO and EPSO display the best performance.

At last, the strategy to calibrate the underwater transponder coordinates using particle swarm algorithm is introduced. As the cost function for transponder coordinates is a unimodal function, CPSO shows better performance than the other algorithms.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This work is partially supported by the Natural Science Foundation of China (51179038, 1109043, 51309067/E091002) and the Program of New Century Excellent Talents in University (NCET-10-0053).

References

[1] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of the 1995 IEEE International Conference on Neural Networks*, vol. 4, no. 2, pp. 1942–1948, December 1995.

[2] 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, 2004.

[3] M. Clerc and J. Kennedy, "The particle swarm-explosion, stability, and convergence in a multidimensional complex space," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 1, pp. 58–73, 2002.

[4] M. S. Arumugam, M. V. C. Rao, and A. W. C. Tan, "A novel and effective particle swarm optimization like algorithm with extrapolation technique," *Applied Soft Computing Journal*, vol. 9, no. 1, pp. 308–320, 2009.

[5] Y. Shi and R. C. Eberhart, "Empirical study of particle swarm optimization," in *Proceedings of the Congress on Evolutionary Computation (CEC '99)*, vol. 3, 1999.

[6] G. Chen, X. Huang, J. Jia et al., "Natural exponential inertia weight strategy in particle swarm optimization," in *Proceedings of the 6th World Congress on Intelligent Control and Automation (WCICA '06)*, vol. 1, pp. 3672–3675, IEEE, June 2006.

[7] Q. Ni and J. Deng, "A new logistic dynamic particle swarm optimization algorithm based on random topology," *The Scientific World Journal*, vol. 2013, Article ID 409167, 8 pages, 2013.

[8] M. M. Noel, "A new gradient based particle swarm optimization algorithm for accurate computation of global minimum," *Applied Soft Computing Journal*, vol. 12, no. 1, pp. 353–359, 2012.

[9] M. G. Epitropakis, V. P. Plagianakos, and M. N. Vrahatis, "Evolving cognitive and social experience in particle swarm optimization through differential evolution," in *Proceedings of the IEEE Congress on Evolutionary Computation (CEC '10)*, pp. 1–8, IEEE, July 2010.

[10] A. A. Mousa, M. A. El-Shorbagy, and W. F. Abd-El-Wahed, "Local search based hybrid particle swarm optimization algorithm for multiobjective optimization," *Swarm and Evolutionary Computation*, vol. 3, pp. 1–14, 2012.

[11] M. H. Moradi and M. Abedini, "A combination of genetic algorithm and particle swarm optimization for optimal DG location and sizing in distribution systems," *International Journal of Electrical Power and Energy Systems*, vol. 34, no. 1, pp. 66–74, 2012.

[12] W. D. Chang, J. P. Cheng, M. C. Hsu et al., "Parameter identification of nonlinear systems using a particle swarm optimization approach," in *Proceedings of the 3rd International Conference on Networking and Computing (ICNC '12)*, pp. 113–117, IEEE, 2012.

[13] J. J. Soon and K. S. Low, "Photovoltaic model identification using particle swarm optimization with inverse barrier constraint," *IEEE Transactions on Power Electronics*, vol. 27, no. 9, pp. 3975–3983, 2012.

[14] B. Jiang, N. Wang, and L. Wang, "Parameter identification for solid oxide fuel cells using cooperative barebone particle swarm optimization with hybrid learning," *International Journal of Hydrogen Energy*, vol. 39, no. 1, pp. 532–542, 2013.

[15] A. Alfi, "Particle swarm optimization algorithm with Dynamic Inertia Weight for online parameter identification applied to Lorenz chaotic system," *International Journal of Innovative Computing, Information and Control*, vol. 8, no. 2, pp. 1191–1204, 2012.

[16] X. Hu, S. Shi, and X. Gu, "An improved particle swarm optimization algorithm for wireless sensor networks localization," in *Proceedings of the 8th International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM '12)*, pp. 1–4, Shanghai, China, September 2012.

[17] F. Sun, J. Yu, and D. Xu, "Visual measurement and control for underwater robots: a survey," in *Proceedings of the 25th Chinese Control and Decision Conference (CCDC '13)*, pp. 333–338, IEEE, 2013.

[18] V. Djapic, D. Nad, G. Ferri et al., "Novel method for underwater navigation aiding using a companion underwater robot as a guiding platforms," in *Proceedings of the 2013 MTS/IEEE OCEANS-Bergen*, pp. 1–10, IEEE, June 2013.

[19] A. Caiti, V. Calabrò, D. Meucci et al., "Underwater Robots: Past, Present and Future," 422–437.

[20] P. Krishnamurthy and F. Khorrami, "A self-aligning underwater navigation system based on fusion of multiple sensors including DVL and IMU," in *Proceedings of the 9th Asian Control Conference (ASCC '13)*, pp. 1–6, IEEE, 2013.

[21] S. Wang and F. Kang, "Research and design of UUV navigation and control integrative simulation system based on component," *Intelligent Information Management*, vol. 4, no. 5, pp. 181–187, 2012.

[22] T. Kashima, A. Asada, and T. Ura, "The positioning system integrated LBL and SSBL using seafloor acoustic mirror transponder," in *Proceedings of the IEEE International Underwater Technology Symposium (UT '13)*, March 2013.

[23] P. Batista, C. Silvestre, and P. Oliveira, "GAS tightly coupled LBL/USBL position and velocity filter for underwater vehicles," in *Proceedings of the European Control Conference (ECC '13)*, pp. 2982–2987, Zurich, Switzerland, 2013.

[24] M. Morgado, P. Oliveira, and C. Silvestre, "Tightly coupled ultrashort baseline and inertial navigation system for underwater vehicles: an experimental validation," *Journal of Field Robotics*, vol. 30, no. 1, pp. 142–170, 2013.

[25] Y. U. Min and H. Junyin, "The calibration of the USBL transducer array for Long-range precision underwater positioning," in *Proceedings of the IEEE 10th International Conference on Signal Processing (ICSP '10)*, pp. 2357–2360, IEEE, October 2010.

[26] D. Ji, Y. Li, and J. Liu, "Seafloor transponder calibration using improved perpendiculars intersection," *Applied Ocean Research*, vol. 32, no. 3, pp. 261–266, 2010.

[27] B. Jiao, Z. Lian, and Q. Chen, "A dynamic global and local combined particle swarm optimization algorithm," *Chaos, Solitons and Fractals*, vol. 42, no. 5, pp. 2688–2695, 2009.

[28] Y. Shi and R. Eberhart, "Modified particle swarm optimizer," in *Proceedings of the IEEE International Conference on Evolutionary Computation (ICEC '98)*, pp. 69–73, IEEE, May 1998.

[29] M. Schwaab, E. C. Biscaia, Jr., J. L. Monteiro, and J. C. Pinto, "Nonlinear parameter estimation through particle swarm optimization," *Chemical Engineering Science*, vol. 63, no. 6, pp. 1542–1552, 2008.

[30] Y. Shi and R. C. Eberhart, "Parameter selection in particle swarm optimization," in *Evolutionary Programming VII*, pp. 591–600, Springer, Berlin, Germany, 1998.

[31] A. Chander, A. Chatterjee, and P. Siarry, "A new social and momentum component adaptive PSO algorithm for image segmentation," *Expert Systems with Applications*, vol. 38, no. 5, pp. 4998–5004, 2011.

[32] M. Morgado, P. Batista, P. Oliveira, and C. Silvestre, "Position USBL/DVL sensor-based navigation filter in the presence of unknown ocean currents," *Automatica*, vol. 47, no. 12, pp. 2604–2614, 2011.

[33] B. Jalving, K. Gade, O. K. Hagen, and K. Vestgård, "A toolbox of aiding techniques for the HUGIN AUV integrated inertial navigation system," in *Proceedings of the OCEANS 2003*, vol. 2, pp. 1146–1153, IEEE, September 2003.