

Research Article

Z-Shaped Transfer Functions for Binary Particle Swarm Optimization Algorithm

Sha-sha Guo,¹ Jie-sheng Wang ,^{1,2} and Meng-wei Guo¹

¹School of Electronic and Information Engineering, University of Science & Technology Liaoning, Anshan 114044, China

²National Financial Security and System Equipment Engineering Research Center, University of Science & Technology Liaoning, Anshan 114044, China

Correspondence should be addressed to Jie-sheng Wang; wang_jiesheng@126.com

Received 20 October 2019; Revised 22 April 2020; Accepted 23 May 2020; Published 8 June 2020

Academic Editor: Juan Carlos Fernández

Copyright © 2020 Sha-sha Guo et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Particle swarm optimization (PSO) algorithm is a swarm intelligent searching algorithm based on population that simulates the social behavior of birds, bees, or fish groups. The discrete binary particle swarm optimization (BPSO) algorithm maps the continuous search space to a binary space through a new transfer function, and the update process is designed to switch the position of the particles between 0 and 1 in the binary search space. Aiming at the existed BPSO algorithms which are easy to fall into the local optimum, a new Z-shaped probability transfer function is proposed to map the continuous search space to a binary space. By adopting nine typical benchmark functions, the proposed Z-probability transfer function and the V-shaped and S-shaped transfer functions are used to carry out the performance simulation experiments. The results show that the proposed Z-shaped probability transfer function improves the convergence speed and optimization accuracy of the BPSO algorithm.

1. Introduction

The particle swarm optimization (PSO) algorithm is one of the widely used evolutionary algorithms inspired by animal social behaviors [1, 2]. It has the search speed, high efficiency, simple algorithm, and so on and has been widely used in crystal structure prediction [3], medical detection [4], grid scheduling [5], robot path planning [6], clustering problem [7], neural network, and many other areas [8–10]. However, many optimization problems have binary searching space, so it is necessary to develop binary optimization algorithm to solve them.

The binary particle swarm optimization (BPSO) algorithm was proposed by Kennedy and Eberhart in 1997 [11]. Typical binary particle swarm optimization algorithm has two different components: a new transfer function and a different location update program. The transfer function is used to map the continuous search space to a binary space, and the update process is designed to switch the position of the particle between 0 and 1 in the binary searching space. The original version of BPSO algorithm is prone to fall into

local extreme, so many scholars have made some improvements to BPSO algorithm. In 2008, Yin proposed a new improved BPSO algorithm, which uses the new updated position formula digital curve with the best polygon approximation [12]. In 2013, Wang et al. [13] used the average information of the individual population and the individual extremum of the particle swarm to determine the current probability value of the particle and removed the influence of the current value of the particle on the next generation. At the same time, the greedy thought is added so that the algorithm not only has the overall optimization characteristics of the particle swarm optimization algorithm but also accelerates the convergence speed of the algorithm. In 2014, Luo [14] revised the sigmoid function, taking the particle velocity as the correction term of the particle position, fully considering the guiding effect between the particles, and ensuring that the particle follows the optimization mode of the algorithm and applies it to solve the vehicle. On the path problem, the effectiveness of the algorithm is improved. Similarly, many other algorithms reference transfer functions, which are converted to binary versions by their

successive versions. In 2013, Sharafi et al. [15] added a transfer function in the tracking mode of cat swarm algorithm, changed the meaning of velocity vector into the probability of mutation in each dimension of cats, and transformed continuous cat swarm algorithm into discrete binary cat swarm algorithm. In 2014, Mirjalili et al. [16] proposed a binary version of the bat algorithm, which is also a probability value that maps velocity values to updated locations using a transfer function. In the same year, Sabba and Chikhi [17] proposed a discrete binary bat algorithm (BINBA) for solving binary space optimization problems. The algorithm is based on the sigmoid function used by Kennedy and Eberhart in the binary particle swarm optimization algorithm proposed in 1997 [11]. BBA was tested on an example of a multidimensional knapsack problem. Compared with other bionic algorithms, the results have a good application prospect. In 2017, Fei [18] proposed a V-shaped transfer function to improve the binary bat algorithm to ensure that the probability of reducing the position of the vector of the position vector of the bat is an element equal to the current best position, and increasing the probability of changing the element of their position vector to the element of the bat is the element of the unequal current optimal position, which helps to enhance the optimization of the binary bat algorithm. In 2015, Emary et al. [19] proposed two new binary wolf optimization algorithms; one of which is to use the sigmoid function to compress the positions of consecutive updates, and then randomly these values are thresholded to obtain an updated double-valued gray wolf position. This method is used for binary gray wolf optimization (BGWO) to find feature subsets to maximize classification accuracy while minimizing the number of affected features. In 2017, Panwar et al. [20] proposed a heuristic binary method for solving the unit commitment problem (UC). This method estimates the continuous and valuable update of the wolves to the global optimal solution, followed by the sigmoid transformation. The simulation results show that BGWO has better performance in solving small- and medium-sized system UC problems compared with other existing heuristic and binary methods. In 2019, Hussien et al. [21] modified the original version of WOA to handle binary optimization problems. To this end, two transfer functions (s-shaped and v-shaped) are proposed to map a continuous search space to a search space. In order to illustrate the function and performance of the proposed binary whale optimization algorithm (BWOA) and apply it to 22 objective functions, 3 engineering optimization problems, and a worldwide sales problem, the results are obtained and the validity is verified. In the same year, Reddy et al. [22] mapped the binary natural PBUC problem to a continuous, real-valued whale position/location mapped to a binary search space through various transformation functions. They introduced three variants of BWOA, using a hyperbolic function, an inverse tangent function, and a sigmoid transfer function, respectively. The convergence characteristics, quality of the solution, and consistency of the results of different BWOA variables are discussed, and the superiority and statistical significance of the proposed method over the existing methods are given.

This shows that the transfer function is the most important part of the binary version of the algorithm. In this paper, a new Z-shaped transfer function is proposed and applied to the particle swarm optimization algorithm. The simulation results show that the new transfer function improves the convergence speed and optimization precision of the algorithm. The structure of this paper is organized as follows. Section 2 introduces the standard particle swarm algorithm. Section 3 introduces the binary particle swarm algorithm. Section 4 proposes a new Z-shaped transfer function. In Section 5, the effectiveness of the improved algorithm is verified by simulation experiments on typical test functions. Section 6 summarizes the full text and proposes future research directions.

2. Particle Swarm Optimization Algorithm

The particle swarm optimization (PSO) algorithm is an intelligent optimization algorithm proposed by Kennedy and Eberhart and Beheshti and Shamsuddin [1, 2] in 1995. It simulates the bird's flight foraging behavior and optimizes the swarm through collective collaboration between birds. In the PSO algorithm, the potential solution to each optimization problem is a bird in the search space, named as a particle. Later, Shi and Eberhart added a new impact factor w , which improved the detection and exploratory, and formed the current standard particle swarm optimization algorithm [23]. All particles have an appropriate value determined by an optimized function, and each particle has a velocity that determines the direction and distance of their flight. Then, the particle follows the current optimal particle to search in the solution space.

The particle's velocity and position are updated by

$$\begin{aligned} v_{id} &= w * v_{id} + c_1 r_1 (p_{id} - x_{id}) + c_2 r_2 (p_{gd} - x_{id}), \\ x_{id} &= x_{id} + v_{id}, \end{aligned} \quad (1)$$

where v_{id} is the velocity of the i th particle in the d th dimension, p_{id} is the optimal position of the i th particle so far, x_{id} is the position of the current i th particle in the d th dimension, p_{gd} is the optimal position that the particle swarm has searched so far, and w is the inertia weight. In this paper, the linear decreasing weight is used, namely, $w_{\max} - ((t * (w_{\max} - w_{\min})) / t_{\max})$, where w_{\max} represents the maximum value of inertia weight, w_{\min} represents the minimum value of inertia weight, t represents the current number of iterations, and t_{\max} represents the maximum number of iterations. This principle is still used in binary PSO algorithm. r_1 and r_2 are two randomly generated acceleration weight coefficients between [0, 1], c_1 and c_2 are acceleration factors, and the value 2 is taken in this paper [24].

There are three parts in the velocity updating formula of particles. The first part is inertia or momentum, which reflects the movement habit of particles, and represents the tendency of particles to maintain their previous velocity. The second part is the cognitive part, which reflects the particle's memory or recall of its own historical experience, and represents that the particle tends to approach the optimal position of its own history. The third part is the social part, which reflects the collective historical experience of

cooperation and knowledge sharing among particles, and represents the tendency of particles to approach the optimal historical position of the community.

3. Binary Particle Swarm Optimization Algorithm

In the binary particle swarm optimization (BPSO) algorithm, the velocity update equation has not changed, but a new velocity transfer function has been introduced to map the original continuous search space to the binary search space. The transfer function concept was originally proposed by Kennedy and Eberhart [11], which allows PSO algorithm to run in the binary searching space. In this version, particles can only be zero or 1 by taking their position vector. The effect of velocity is to indicate the probability that the bits take 0 or 1, so they propose a Sigmoid transfer function, as shown in equation (2), which can convert all real values of velocity into probability values [0, 1]:

$$T(v_i^k(t)) = \frac{1}{1 + e^{-v_i^k(t)}}, \quad (2)$$

where $v_i^k(t)$ represents the velocity of particle i at iteration t in k dimension.

After converting the velocity to a probability value, the position vector can be updated with the probability of its velocity by

$$x_i^k(t+1) = \begin{cases} 0, & \text{If rand} < T(v_i^k(t)), \\ 1, & \text{If rand} \geq T(v_i^k(t)), \end{cases} \quad (3)$$

where $v_i^k(t)$ represents the velocity of particle i at iteration t in k dimension.

The flowchart of BPSO algorithm is shown in Figure 1. According to the experimental analysis, the original BPSO has some shortcomings, such as premature convergence and easy to fall into local optimal, so it has been continuously improved since BPSO was proposed. In 2008, Lee et al. proposed another modification of BPSO algorithm [25], which allows the continuous update velocity and position of PSO algorithm. In this improvement, they replace the velocity with the displacement in the transfer function. The probability formula is described as follows:

$$T(x_i^k(t)) = \frac{1}{1 + e^{-x_i^k(t)}}, \quad (4)$$

where $x_i^k(t)$ represents the position of particle i at iteration t in k dimension.

The corresponding position updating formula is defined as follows:

$$x_i^k(t+1) = \begin{cases} 0, & \text{If rand} < T(x_i^k(t)), \\ 1, & \text{If rand} \geq T(x_i^k(t)), \end{cases} \quad (5)$$

where $x_i^k(t)$ represents the position of particle i at iteration t in k dimension.

This transfer function is named as S-shaped transfer function, and a set of S-shaped transfer functions are formed by changing parameters, whose expressions and graphs are shown in Table 1 and Figure 2, respectively [24]. It can be

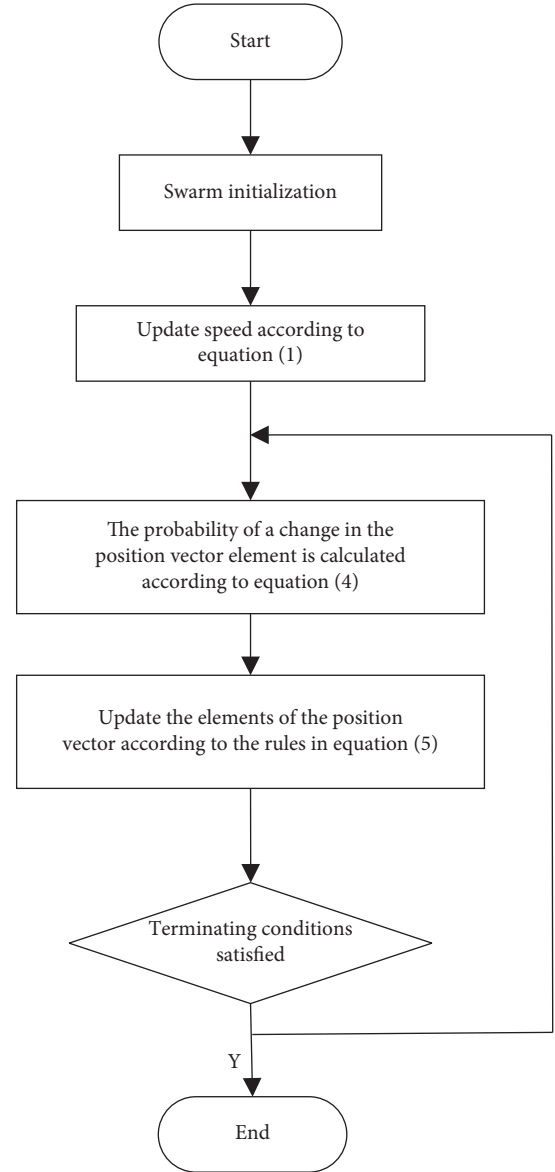


FIGURE 1: Flow chart of BPSO algorithm.

TABLE 1: S-shaped transfer functions.

Experiment name	Name	Expression
BPSO1	s_1	$T_1(x) = (1/(1 + e^{-2x}))$
BPSO2	s_2	$T_2(x) = (1/(1 + e^{-x}))$
BPSO3	s_3	$T_3(x) = (1/(1 + e^{(-x/2)}))$
BPSO4	s_4	$T_4(x) = (1/(1 + e^{(-x/3)}))$

seen from Figure 3 that the velocity value of s_1 is larger than that of s_2 , and the saturation speed is also accelerated. Similarly, when the speed becomes smaller, as shown in s_3 and s_4 , the saturation speed will decrease. To sum up, the change probability of position vector increases with the increase of slope of these functions. Therefore, when the velocity is the same, the probability value returned by s_1 is the largest. In the following Section 5, the influence of different slopes on solving the function optimization problem will be studied.

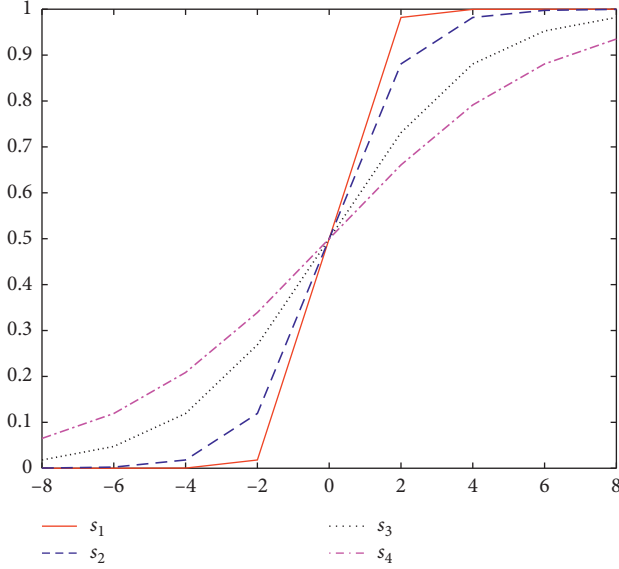


FIGURE 2: S-shaped transfer functions.

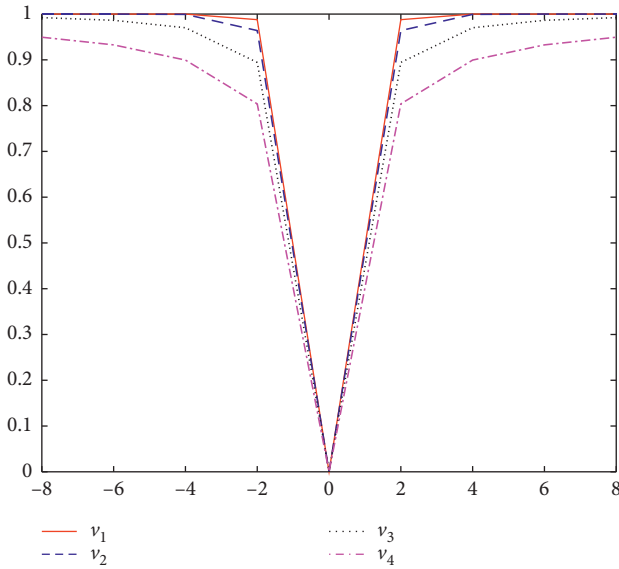


FIGURE 3: V-shaped transfer functions.

In 2009, Rashedi et al. proposed a new transfer function named as V-shaped transfer function and a new position update strategy [26]. The formula is described as follows:

$$x_i^k(t+1) = \begin{cases} (x_i^k(t+1))^{-1}, & \text{If } \text{rand} < T(x_i^k(t)), \\ x_i^k(t), & \text{If } \text{rand} \geq T(x_i^k(t)), \end{cases} \quad (6)$$

where $x_i^k(t)$ represents the velocity of particle i at iteration t in k dimension and $(x_i^k(t+1))^{-1}$ is the complement of $x_i^k(t)$.

According to the characteristics of V-shaped transfer function, a series of V-shaped transfer functions are proposed by using different functional equations, whose expressions and graphs are shown in Table 2 and Figure 3, respectively.

TABLE 2: V-shaped transfer functions [24].

Experiment name	Name	Expression
BPSO5	V_1	$T_5(x) = \text{erf}((\sqrt{\pi}/2)x) $
BPSO6	V_2	$T_6(x) = \tanh(x) $
BPSO7	V_3	$T_7(x) = (x)/\sqrt{1+x^2} $
BPSO8	V_4	$T_8(x) = (2/\pi)\arctan((\pi/2)x) $

According to Figure 3, this function is a symmetric function. When the absolute value of velocity is larger, the probability of particle position change is higher. For different V-shaped functions, it is easy to find that when the slope of the function is higher, the probability of the particle position change is higher. In other words, when the velocity is constant, the function with higher slope has a higher probability of the returned particle change. Similarly, the influence of different slopes on the processing of function optimization problems will be analyzed in Section 5.

4. Improved Binary Particle Swarm Optimization Algorithm

Based on the characteristics of binary particle swarm optimization algorithm, the continuous search space is mapped to the discrete binary space. The purpose of the transfer function is to represent the probability that the element of the position vector goes from 0 to 1, so the transfer function must be a bounded function of $[0, 1]$. In addition, when the velocity value is 0, the probability of change should be relatively small because when the particle finds the optimal value, the velocity should be reduced to 0, and the probability of the position change of the particle should be 0. According to the characteristics of transfer function, a new Z-shaped transfer function is proposed, which is defined as follows:

$$T(x_i^k(t)) = \sqrt{1 - a^{x_i^k(t)}}, \quad (7)$$

where $x_i^k(t)$ represents the velocity of particle i at iteration t in k dimension and a is a positive integer. By changing the value of a , a set of Z-shaped function families is obtained, whose expressions and figures are shown in Table 3 and Figure 4, respectively.

As shown in Figure 4, the mapping function is an asymmetric mapping function. The asymmetric mapping function basically satisfies the absolute value of the velocity to determine the mapping probability of the particle position vector variation, so the convergence speed is fast. When the parameter $Dim_{particle} = Dim_{Function} \times 15$ changes, the slope of the function also changes. The larger $Dim_{particle} = Dim_{Function} \times 15$, the smaller the slope of the function. That is to say, when the speed is the same, the probability that the returned particle position of the parameter $Dim_{particle} = Dim_{Function} \times 15$ is small changes is larger.

5. Simulation Experiments and Result Analysis

5.1. Selection of Test Functions. The process of mapping continuous search space to discrete search space using the

TABLE 3: Z-shaped transfer functions.

Experiment name	Name	Expression
BPSO9	Z_1	$T_9(x) = \sqrt{1 - 2^x}$
BPSO10	Z_2	$T_{10}(x) = \sqrt{1 - 5^x}$
BPSO11	Z_3	$T_{11}(x) = \sqrt{1 - 8^x}$
BPSO12	Z_4	$T_{12}(x) = \sqrt{1 - 20^x}$

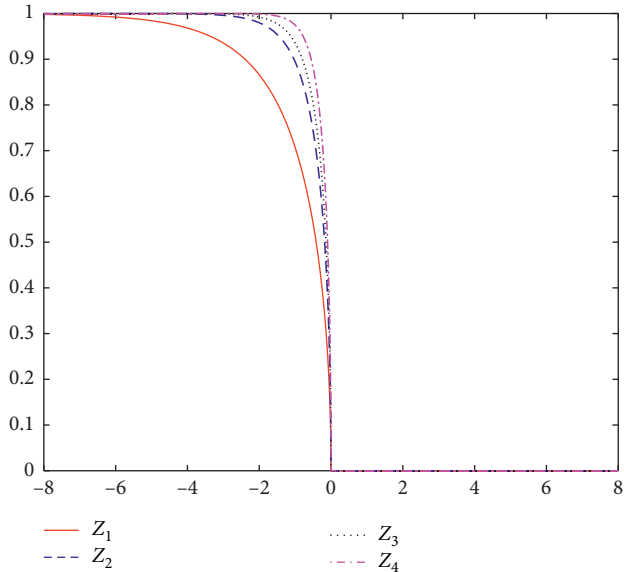


FIGURE 4: Z-shaped transfer functions.

proposed transfer function is shown in Figure 5. To represent each continuous variable in binary, 15 bits are adopted. It should be noted that each function variable retains a bit of its sign. Therefore, the particle size is calculated by

$$\text{Dim}_{\text{particle}} = \text{Dim}_{\text{Function}} \times 15. \quad (8)$$

According to equation (8), when the test function has 5, 10, and 30 dimensions, the corresponding particle sizes are 75, 150, and 450, respectively. In order to evaluate the improved BPSO algorithm based on the Z-shaped transfer functions, the simulation experiments were carried out by using the nine benchmark functions proposed in the CEC special conference. The expressions of the benchmark functions and the three-dimensional graphics are shown in Table 4 and Figure 6, respectively.

5.2. Performance Comparison of Z-Shaped Transfer Functions.

In order to test the performance of the Z-shaped transfer function, the parameters of the Z-shaped transfer function are set as 2, 5, 8, and 20, respectively, and the influence of different parameter settings on the simulation experiments is analyzed. In order to further verify the optimization accuracy of Z-shaped transfer function, each test function was run 10 times, and the best, worst, average, and STD values of 9 functions in 5 and 30 dimensions were recorded,

respectively. The maximum number of iterations is $\text{Max_iter} = 500$. The simulation results of $D = 5$ and $D = 30$ are shown in Figures 7 and 8, respectively, and the statistical results are shown in Tables 5 and 6, respectively.

It can be seen from Figure 7 and Table 5 that when the test functions have 5 dimensions, changing the parameters has no effect on the final convergence results, and the final convergence results are the same. However, the rate of convergence will be different. In the function convergence curves, it can be seen that increasing the value of parameters can improve the convergence speed of the algorithm, but it is not the case that the larger the parameters are, the faster the convergence speed will be. In the convergence curves of $F_1, F_2, F_3, F_7,$ and F_9 , the transfer function whose parameter is set to 20 has the fastest convergence speed. In the convergence curves of $F_4, F_5,$ and F_8 , the transfer function with parameter set to 8 has the fastest convergence speed.

It can be seen from Figure 8 and Table 6 that when the test functions have 30 dimensions, changing the size of parameters has no obvious influence on the convergence speed of the algorithm, but has an influence on the final convergence result. In the functions $F_1, F_2, F_3, F_4, F_5, F_7, F_8,$ and F_9 , the parameters are set to 5, 8, and 20, and the effect is better when the parameter is 2, that is to say that increasing the parameter can improve the accuracy of the algorithm. However, when the parameters are set to 5, 8, and 20, the results are the same. That is to say, it is not the case that the larger the parameters setting, the better the performance. When the parameters are set to a certain size, increasing the parameters will not improve the accuracy of the algorithm.

5.3. Performance Comparison of Z-Shaped, S-Shaped, and V-Shaped Transfer Functions

5.3.1. Test Function Results and Analysis. To further investigate the validity of the Z-shaped transfer function, the Z-shaped transfer function is compared to the S-shaped and the V-shaped transfer function. Through the above analysis, it can be concluded that when the parameter of the Z-shaped transfer function is set to 5, the convergence precision and convergence speed are optimal. Therefore, in this set of simulation experiments, the parameter of the new Z-shaped transfer function is set as 5, the selected 12 test functions have 10 dimensions, and the maximum number of iterations is 500. The simulation results and numerical analysis are shown in Figure 9 and Table 7, respectively.

It can be seen from the convergence curves that the proposed new transfer function improves the convergence speed of the algorithm. According to the data shown in Table 7, the new transfer function improves the accuracy of the BPSO algorithm. In particular, for functions $F_1, F_2, F_3,$ and F_7 , the optimal value of the function can be accurately derived. Similarly, for functions F_8 , the accuracy is 15 orders of magnitude higher than other transfer functions. For the composite functions $F_{10} \sim F_{12}$, BPSO10 improves the convergence accuracy and convergence speed of the algorithm, but the optimal value of the function

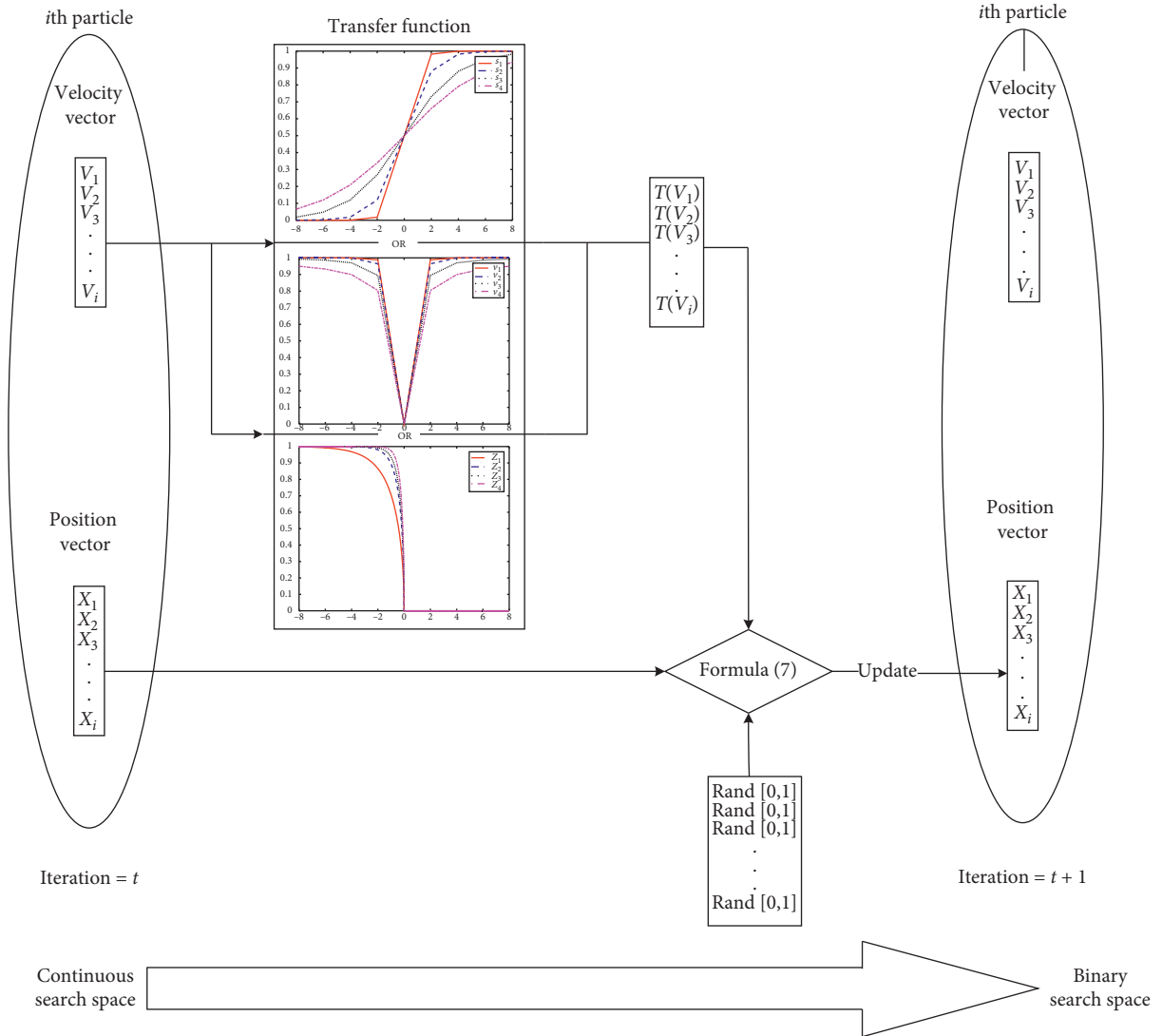


FIGURE 5: Procedure of mapping continuous search space to discrete search space.

cannot be found, which will be the focus of future research and learning. In other functions, it also improves the convergence speed and convergence precision of the algorithm compared to other transfer functions.

5.3.2. Nonparametric Test Analysis. In order to verify the stability of the algorithm, we perform a nonparametric test on the optimal value of the test function. In this work, Wilcoxon rank sum test [27] is used as a nonparametric statistical test to determine the importance of the results. Table 8 depicts the 5% p value obtained from this test. It can be seen from Table 8 that the p values for all test functions are much less than 0.05, highlighting the significant advantages of BPSO10 over other versions of the binary particle swarm algorithm based on the p value (less than 0.05) algorithm.

5.4. Comparison with Other Versions of Binary Particle Swarm Algorithms. In order to further investigate the effectiveness of the transfer function, this paper compares the test results of BPSO10 with other recent binary particle swarm versions. Firstly, it was compared with the binary algorithm (BPSOGSA) proposed by Mirjalili et al. [28] based on particle swarm optimization algorithm and gravity search algorithm. The dimension of the test function was 5, and the comparison result was shown in Table 9. At the same time, it also proposed binary hybrid topology particle swarm optimization (BHTPSO) and binary hybrid topology particle swarm optimization quadratic interpolation (BHTPSO-QI), proposed by Beheshti et al. [29], and swarm optimization algorithm based on genetic algorithm is put forward by the quantum-behaved particle swarm optimization (SOGA) by Sun et al. [30] are compared. The dimension of the test

TABLE 4: Test functions.

Function	Dim	Rang	F _{min}
$F_1(x) = \sum_{i=1}^n x_i^2$	5, 8, 10, 30	[-100, 100]	0
$F_2(x) = \sum_{i=1}^n x_i + \prod_{i=1}^n x_i $	5, 8, 10, 30	[-10, 10]	0
$F_3(x) = \sum_{i=1}^n (\sum_{j=1}^n x_j)^2$	5, 8, 10, 30	[100, 100]	0
$F_4(x) = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	5, 8, 10, 30	[-30, 30]	0
$F_5(x) = \sum_{i=1}^n [(x_i + 5)^2]$	5, 8, 10, 30	[-100, 100]	0
$F_6(x) = \sum_{i=1}^n (ix_i)^4 + \text{random}[0, 1]$	5, 8, 10, 30	[-1.28, 1.28]	0
$F_7(x) = \sum_{i=1}^n [x_i^2 - 10 \cos(2\pi x_i) + 10]$	5, 8, 10, 30	[-5.12, -5.12]	0
$F_8(x) = -20 \exp \times (-0.2\sqrt{1/n} \sum_{i=1}^n x_i) - \exp(1/n \sum_{i=1}^n \cos(2\pi x_i)) + 20 + e$	5, 8, 10, 30	[-32, 32]	0
$F_9 = (\pi/n) \{10 \sin(\pi y_1) + \sum_{i=1}^{n-1} (y_i - 1)^2 [1 + 10 \sin^2(\pi y_i + 1)] + (y_n - 1)^2\} + \sum_{i=1}^n u(x_i, 10, 100, 4)$ $y_i = 1 + \frac{x_i + 1}{4}$	5, 8, 10, 30	[-50, 50]	0
$u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m & x_i > a \\ 0 & -a < x_i < a \\ k(-x_i - a)^m & x_i < -a \end{cases}$			
F_{10} [24]: $f_1 \sim f_2$ = Rotated Expanded Scaffer's FF Function Where FF is Shifted Rosenbrock. $f_3 \sim f_4$ = Rastrigin's Function $f_5 \sim f_6$ = FF4 Function $f_7 \sim f_8$ = Weierstrass Function $f_9 \sim f_{10}$ = Griewank's Function $\sigma = [1, 1, 1, 1, 1, 2, 2, 2, 2, 2]$ $\lambda = [5 * 5/100; 5/100; 5 * 1; 1; 5 * 1; 1; 5 * 10; 5 * 5/200; 5/200]$	10	[-5, 5]	360
F_{11} [24] All settings are the same as F_{10} $x_j = \begin{cases} x_j & x_j - o_{1j} < 1/2 \\ \text{round}(2x_j)/2 & x_j - o_{1j} \geq 1/2 \end{cases}$ for $j = 1, 2, \dots, n$	10	[-5, 5]	360
Except $\text{round}(x) = \begin{cases} a - 1 & \text{if } x \leq 0 \ \& \ b \geq 0.5 \\ 1 & \text{if } b < 0.5 \\ a1 & x > 0 \ \& \ b \geq 0.5 \end{cases}$ Where a is x 's integral part and b is x 's decimal part.			
F_{12} [24] f_1 = Weierstrass Function f_2 = Rotated Expanded Scaffer's FF Function Where FF is Shifted Rosenbrock. f_3 = FF4 Function Where FF is Rastrigin's Function f_4 = Ackley's Function f_5 = Rastrigin's Function f_6 = Griewank's Function f_7 = Noncontinuous Expanded Scaffer's FF Function Where FF is Shifted Rosenbrock. f_8 = Noncontinuous Rastrigin's Function f_9 = High Conditioned Elliptic Function f_{10} = Sphere Function with Noise in Fitness $\sigma_i = 2$, for $i = 1, 2, \dots, n$ $\lambda = [10; 5/20; 1; 5/32; 1; 5/100; 5/50; 1; 5/100; 5/100]$	10	[-5, 5]	260

function was 8 dimensions, and the comparison result is shown in Table 10.

It can be seen from the data in Tables 9 and 10 that when the test function is 5 or 8 dimensions, the test results of the newly proposed z-type transfer function are significantly higher than those of BPSOGSA, BHTPSO, BHTPSO-QI, and SOGA proposed in recent studies. In particular, in the test

functions F_1, F_2 , and F_3 , BPSO10 can still accurately find the optimal value, and the STD data shows that the value is 0, indicating that the algorithm is very stable and has strong robustness. Among other test functions, BPSO10 also improves the optimization accuracy of the algorithm to varying degrees. Therefore, the newly proposed z-type transfer function has good optimization ability.

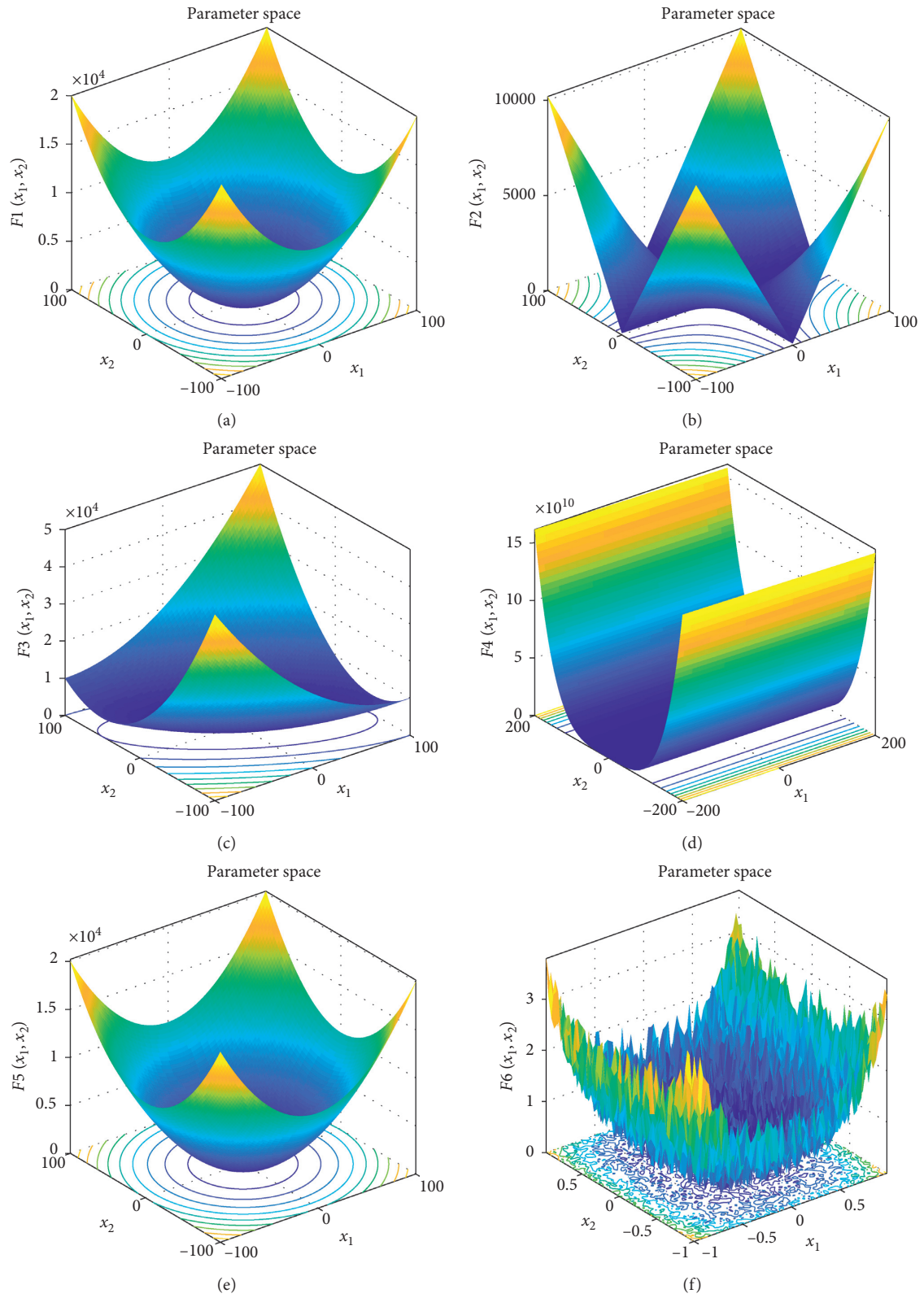


FIGURE 6: Continued.

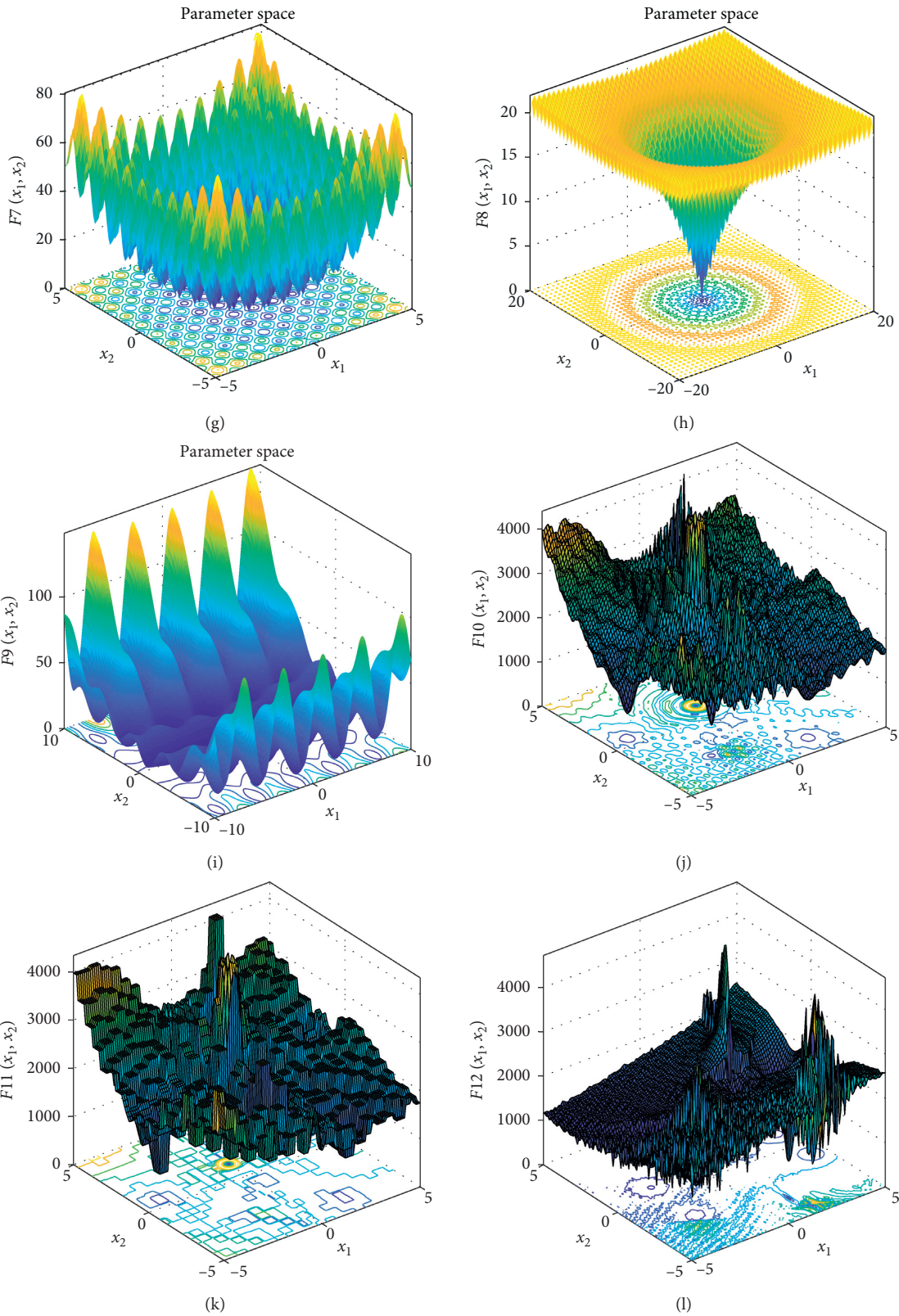


FIGURE 6: 3D graphs of benchmark functions. (a) F_1 . (b) F_2 . (c) F_3 . (d) F_4 . (e) F_5 . (f) F_6 . (g) F_7 . (h) F_8 . (i) F_9 . (j) F_{10} . (k) F_{11} . (l) F_{12} .

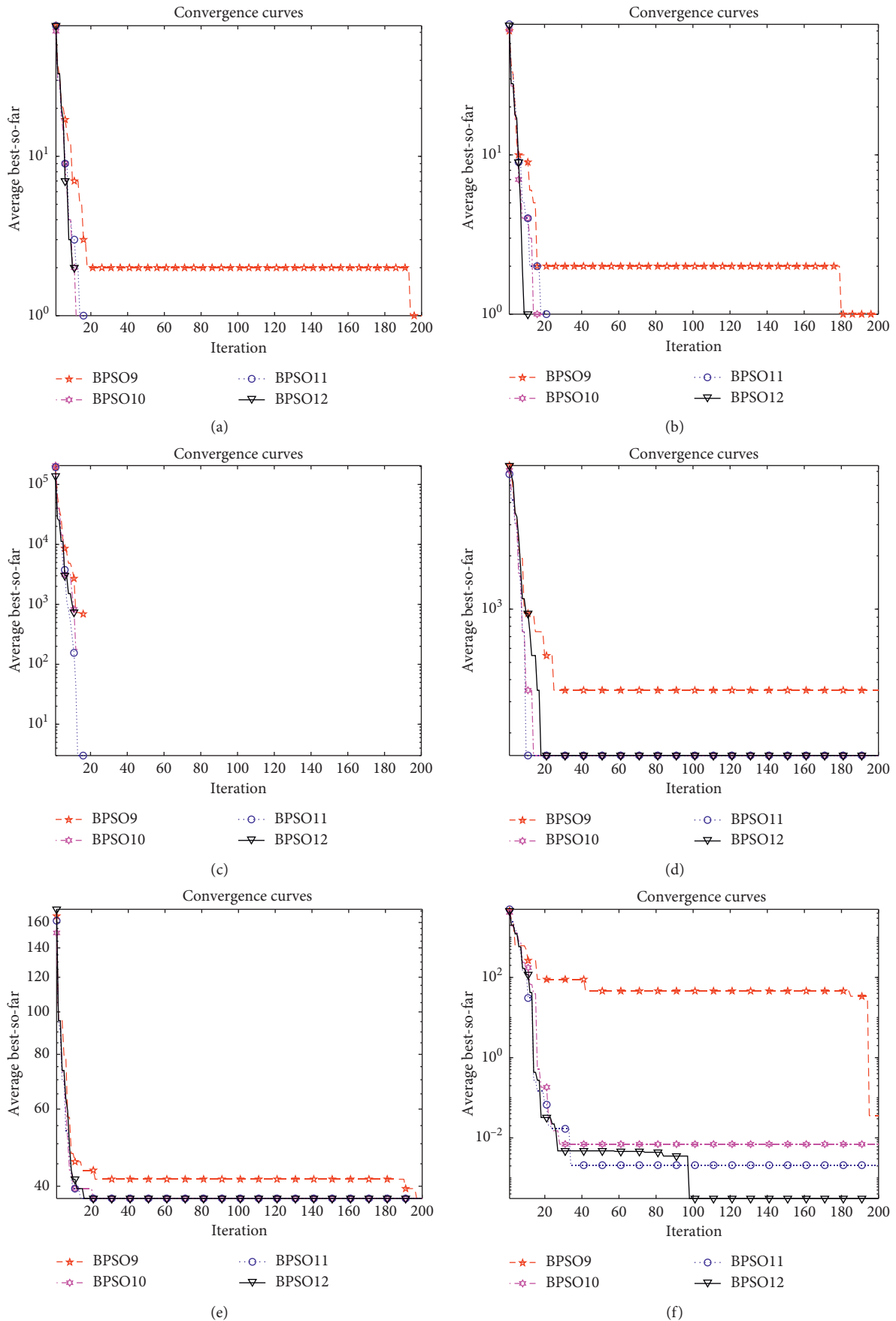


FIGURE 7: Continued.

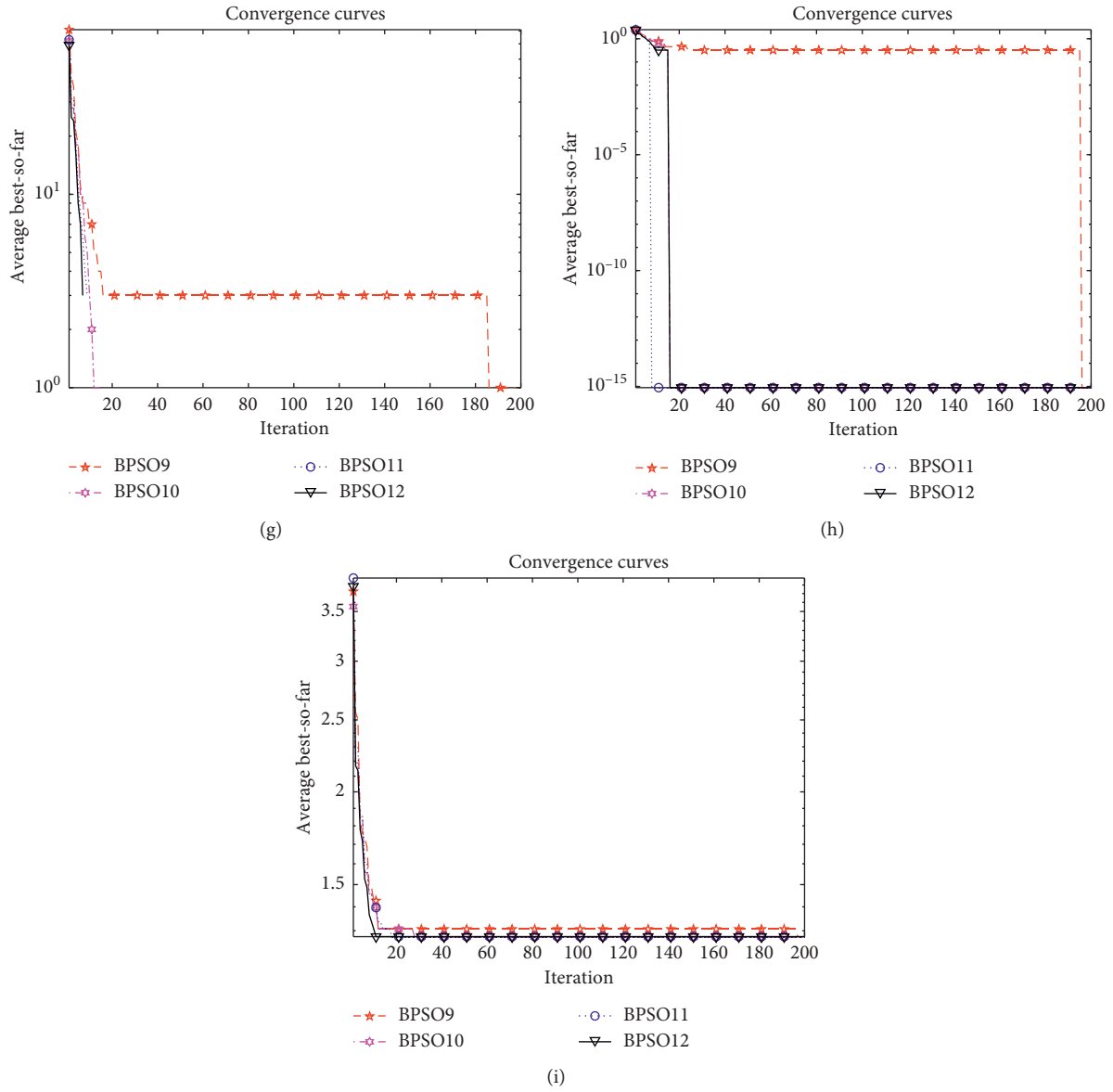
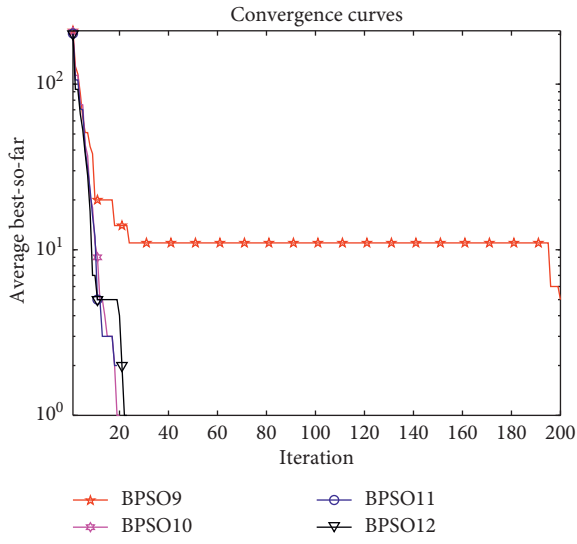
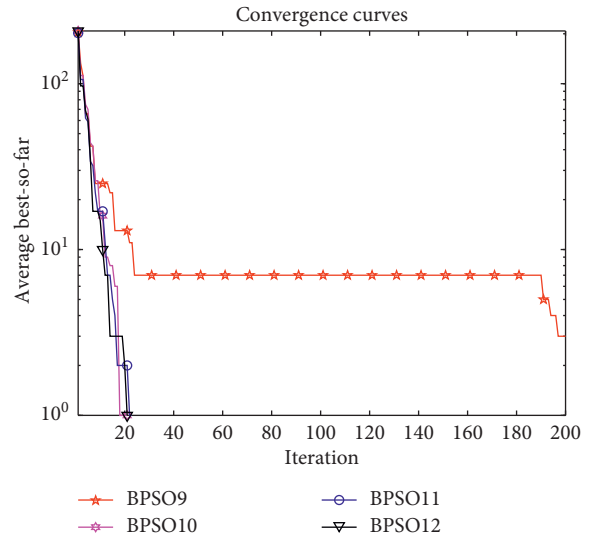


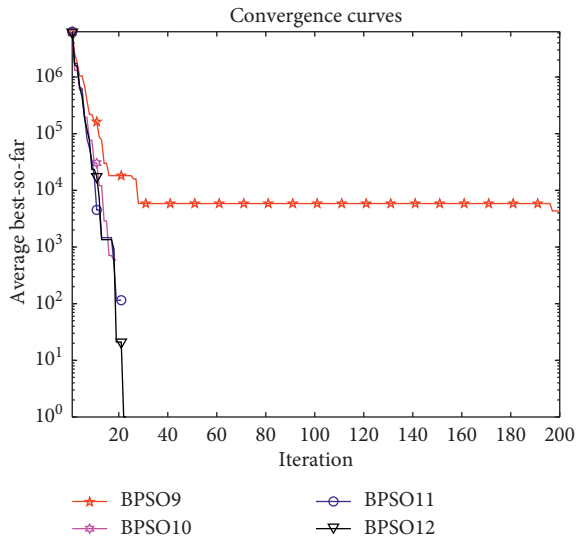
FIGURE 7: Convergence curves of the benchmark functions under $D = 5$. (a) F_1 . (b) F_2 . (c) F_3 . (d) F_4 . (e) F_5 . (f) F_6 . (g) F_7 . (h) F_8 . (i) F_9 .



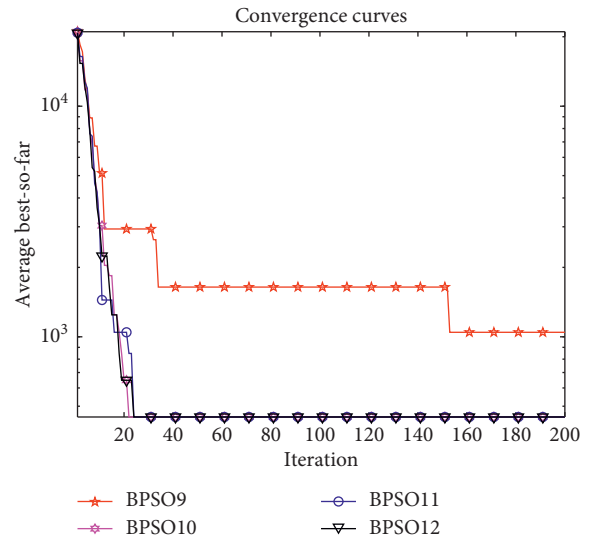
(a)



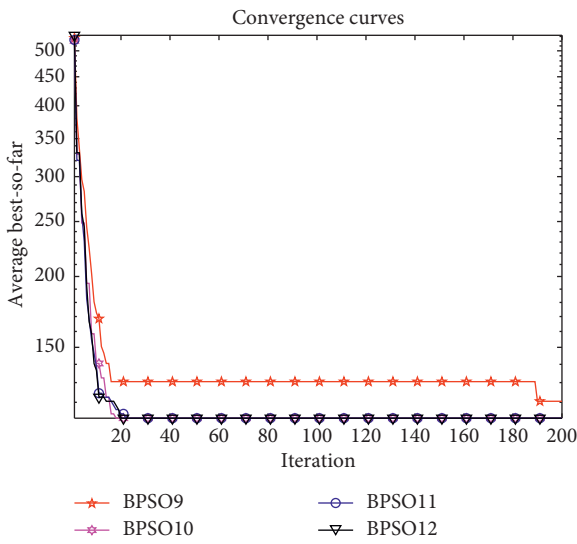
(b)



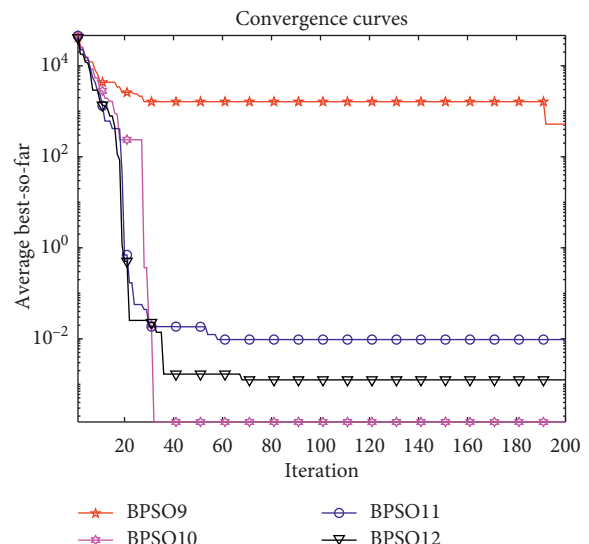
(c)



(d)



(e)



(f)

FIGURE 8: Continued.

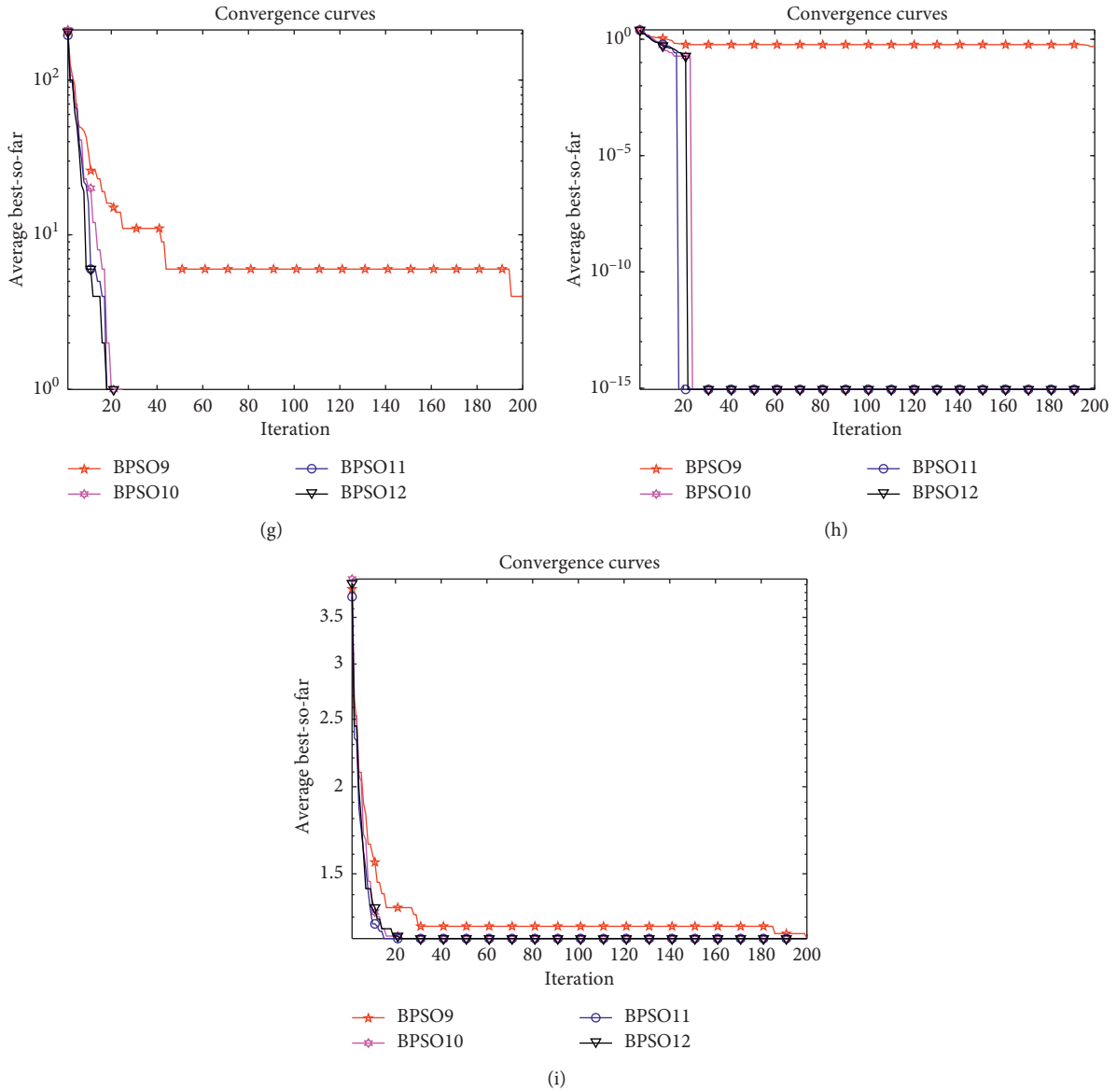


FIGURE 8: Convergence curves of the benchmark functions under $D = 30$. (a) F_1 . (b) F_2 . (c) F_3 . (d) F_4 . (e) F_5 . (f) F_6 . (g) F_7 . (h) F_8 . (i) F_9 .

TABLE 5: Numerical results under $D = 5$.

Function	Arithmetic	Best	Worst	Ave	Std
F_1	BPSO9	0.0000	0.0000	0.0000	0.0000
	BPSO10	0.0000	0.0000	0.0000	0.0000
	BPSO11	0.0000	0.0000	0.0000	0.0000
	BPSO12	0.0000	0.0000	0.0000	0.0000
F_2	BPSO9	0.0000	0.0000	0.0000	0.0000
	BPSO10	0.0000	0.0000	0.0000	0.0000
	BPSO11	0.0000	0.0000	0.0000	0.0000
	BPSO12	0.0000	0.0000	0.0000	0.0000
F_3	BPSO9	0.0000	0.0000	0.0000	0.0000
	BPSO10	0.0000	0.0000	0.0000	0.0000
	BPSO11	0.0000	0.0000	0.0000	0.0000
	BPSO12	0.0000	0.0000	0.0000	0.0000
F_4	BPSO9	74.0000	74.0000	74.0000	0.0000
	BPSO10	74.0000	74.0000	74.0000	0.0000
	BPSO11	74.0000	74.0000	74.0000	0.0000
	BPSO12	74.0000	74.0000	74.0000	0.0000

TABLE 5: Continued.

Function	Arithmetic	Best	Worst	Ave	Std
F_5	BPSO9	18.7500	18.7500	18.7500	0.0000
	BPSO10	18.7500	18.7500	18.7500	0.0000
	BPSO11	18.7500	18.7500	18.7500	0.0000
	BPSO12	18.7500	18.7500	18.7500	0.0000
F_6	BPSO9	0.0039	0.0836	0.0279	0.0219
	BPSO10	0.0002	0.0130	0.0044	0.0043
	BPSO11	0.0001	0.0043	0.0018	0.0013
	BPSO12	0.0001	0.0024	0.0010	0.0008
F_7	BPSO9	0.0000	0.0000	0.0000	0.0000
	BPSO10	0.0000	0.0000	0.0000	0.0000
	BPSO11	0.0000	0.0000	0.0000	0.0000
	BPSO12	0.0000	0.0000	0.0000	0.0000
F_8	BPSO9	$8.8818E-16$	$8.8818E-16$	$8.8818E-16$	0.0000
	BPSO10	$8.8818E-16$	$8.8818E-16$	$8.8818E-16$	0.0000
	BPSO11	$8.8818E-16$	$8.8818E-16$	$8.8818E-16$	0.0000
	BPSO12	$8.8818E-16$	$8.8818E-16$	$8.8818E-16$	0.0000
F_9	BPSO9	1.3744	1.3744	1.3744	0.0000
	BPSO10	1.3744	1.3744	1.3744	0.0000
	BPSO11	1.3744	1.3744	1.3744	0.0000
	BPSO12	1.3744	1.3744	1.3744	0.0000

TABLE 6: Numerical results under $D = 30$.

Function	Arithmetic	Best	Worst	Ave	Std
F_1	BPSO9	2.0000	7.0000	4.2000	1.4697
	BPSO10	0.0000	0.0000	0.0000	0.0000
	BPSO11	0.0000	0.0000	0.0000	0.0000
	BPSO12	0.0000	0.0000	0.0000	0.0000
F_2	BPSO9	2.0000	8.0000	4.5000	1.9621
	BPSO10	0.0000	0.0000	0.0000	0.0000
	BPSO11	0.0000	0.0000	0.0000	0.0000
	BPSO12	0.0000	0.0000	0.0000	0.0000
F_3	BPSO9	9.0000	2683.0000	1517.0000	890.0239
	BPSO10	0.0000	0.0000	0.0000	0.0000
	BPSO11	0.0000	0.0000	0.0000	0.0000
	BPSO12	0.0000	0.0000	0.0000	0.0000
F_4	BPSO9	449.0000	2041.0000	1205.2000	434.1667
	BPSO10	449.0000	449.0000	449.0000	0.0000
	BPSO11	449.0000	449.0000	449.0000	0.0000
	BPSO12	449.0000	449.0000	449.0000	0.0000
F_5	BPSO9	118.5000	124.5000	120.9000	1.9596
	BPSO10	112.5000	112.5000	112.5000	0.0000
	BPSO11	112.5000	112.5000	112.5000	0.0000
	BPSO12	112.5000	112.5000	112.5000	0.0000
F_6	BPSO9	210.8445	1124.7927	618.0885	271.3089
	BPSO10	0.0001	0.0526	0.0216	0.0155
	BPSO11	0.0000	0.0132	0.0042	0.0039
	BPSO12	0.0004	0.0049	0.0018	0.0012
F_7	BPSO9	1.0000	7.0000	4.9000	1.8682
	BPSO10	0.0000	0.0000	0.0000	0.0000
	BPSO11	0.0000	0.0000	0.0000	0.0000
	BPSO12	0.0000	0.0000	0.0000	0.0000
F_8	BPSO9	$8.8818E-16$	0.4927	0.3065	0.1329
	BPSO10	$8.8818E-16$	$8.8818E-16$	$8.8818E-16$	0.0000
	BPSO11	$8.8818E-16$	$8.8818E-16$	$8.8818E-16$	0.0000
	BPSO12	$8.8818E-16$	$8.8818E-16$	$8.8818E-16$	0.0000
F_9	BPSO9	1.2209	1.2811	1.2520	0.0176
	BPSO10	1.2108	1.2108	1.2108	0.0000
	BPSO11	1.2108	1.2108	1.2108	0.0000
	BPSO12	1.2108	1.2108	1.2108	0.0000

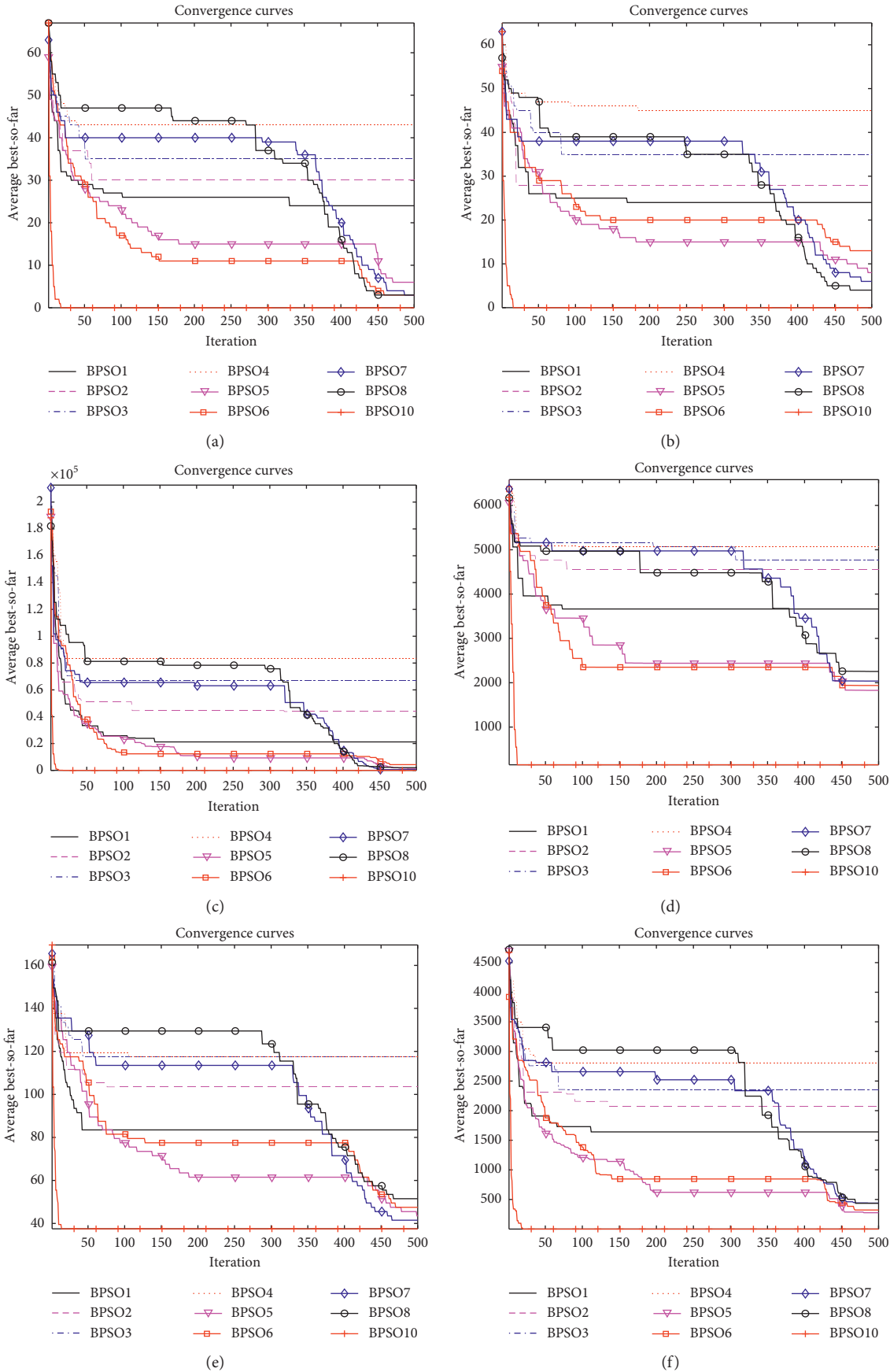


FIGURE 9: Continued.

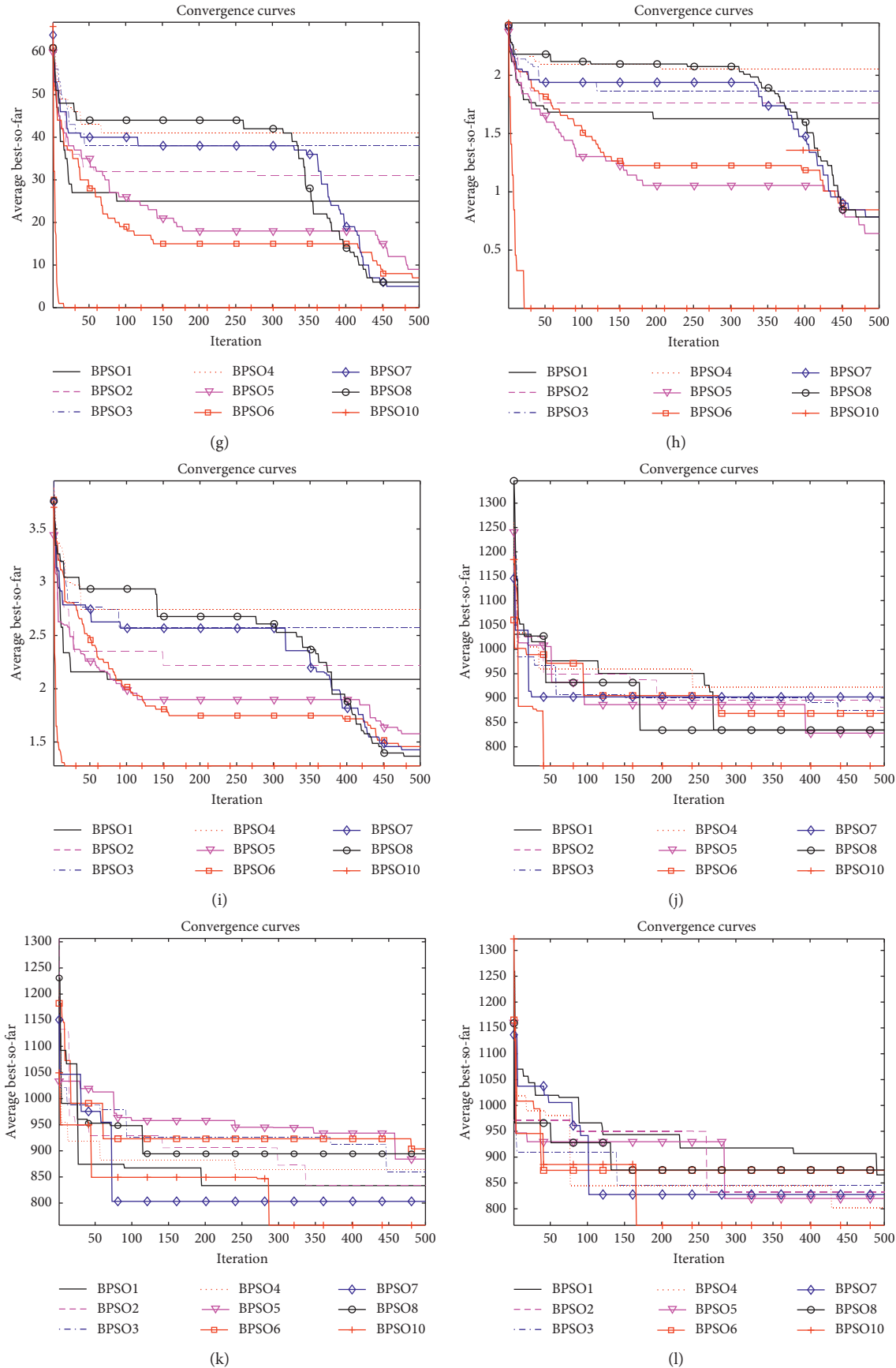


FIGURE 9: Convergence curves of the benchmark functions. (a) F_1 . (b) F_2 . (c) F_3 . (d) F_4 . (e) F_5 . (f) F_6 . (g) F_7 . (h) F_8 . (i) F_9 . (j) F_{10} . (k) F_{11} . (l) F_{12} .

TABLE 7: Numerical statistics.

Function	Arithmetic	Best	Worst	Ave	Std
F_1	BPSO1	20.0000	25.0000	23.7000	1.4177
	BPSO2	28.0000	34.0000	30.8000	1.8330
	BPSO3	36.0000	41.0000	38.4000	1.6852
	BPSO4	38.0000	44.0000	42.1000	1.5780
	BPSO5	2.0000	14.0000	7.7000	2.9343
	BPSO6	4.0000	9.0000	5.7000	1.5524
	BPSO7	3.0000	7.0000	5.0000	1.3416
	BPSO8	2.0000	7.0000	4.0000	1.4142
	BPSO10	0.0000	0.0000	0.0000	0.0000
	F_2	BPSO1	21.0000	28.0000	24.9000
BPSO2		28.0000	35.0000	32.5000	2.1095
BPSO3		38.0000	42.0000	39.2000	1.1662
BPSO4		38.0000	43.0000	41.4000	1.7436
BPSO5		4.0000	12.0000	7.6000	2.3324
BPSO6		4.0000	10.0000	6.8000	1.8868
BPSO7		2.0000	7.0000	4.7000	1.6155
BPSO8		1.0000	6.0000	3.8000	1.6000
BPSO10		0.0000	0.0000	0.0000	0.0000
F_3		BPSO1	2.0936E + 04	2.9895E + 04	2.4866E + 04
	BPSO2	3.5023E + 04	4.7871E + 04	4.0582E + 04	4.5452E + 03
	BPSO3	4.0798E + 04	7.7036E + 04	6.3317E + 04	9.7782E + 03
	BPSO4	6.1630E + 04	8.7746E + 04	7.4415E + 04	7.9859E + 03
	BPSO5	9.2100E + 02	7.8520E + 03	3.2089E + 03	1.9484E + 03
	BPSO6	2.6700E + 02	4.6020E + 03	2.1907E + 03	1.3673E + 03
	BPSO7	9.2000E + 01	3.1230E + 03	1.0619E + 03	1.0267E + 03
	BPSO8	1.5600E + 02	2.5080E + 03	1.0006E + 03	8.4595E + 02
	BPSO10	0.0000	0.0000	0.0000	0.0000
	F_4	BPSO1	3651.0000	4256.0000	3899.0000
BPSO2		4072.0000	4657.0000	4406.2000	159.0540
BPSO3		4163.0000	4971.0000	4750.7000	219.8095
BPSO4		4771.0000	5162.0000	4963.3000	103.9173
BPSO5		1828.0000	3047.0000	2207.5000	324.0522
BPSO6		1937.0000	2454.0000	2135.4000	210.8327
BPSO7		1643.0000	2650.0000	2179.6000	355.8585
BPSO8		1640.0000	2848.0000	2175.5000	386.0337
BPSO10		149.0000	149.0000	149.0000	0.0000
F_5		BPSO1	85.5000	89.5000	87.5000
	BPSO2	97.5000	107.5000	102.5000	3.6056
	BPSO3	105.5000	117.5000	112.3000	3.8158
	BPSO4	115.5000	125.5000	119.9000	3.2000
	BPSO5	37.5000	61.5000	52.1000	6.3906
	BPSO6	45.5000	53.5000	50.3000	2.4000
	BPSO7	41.5000	53.5000	47.7000	4.1425
	BPSO8	43.5000	47.5000	45.5000	1.7889
	BPSO10	37.5000	37.5000	37.5000	0.0000
	F_6	BPSO1	1174.2146	1680.9851	1489.1710
BPSO2		1760.8619	2317.8085	1949.8876	155.5235
BPSO3		2496.6527	2707.5556	2584.3121	63.7418
BPSO4		2691.9604	2984.0308	2850.7730	103.5472
BPSO5		225.0052	567.0026	401.3030	109.7147
BPSO6		162.0050	665.0021	346.3023	133.7193
BPSO7		145.0009	504.0035	245.7022	113.2767
BPSO8		144.0011	482.0005	297.3014	114.2555
BPSO10		0.0001	0.0101	0.0035	0.0028

TABLE 7: Continued.

Function	Arithmetic	Best	Worst	Ave	Std
F_7	BPSO1	22.0000	28.0000	25.3000	2.2383
	BPSO2	29.0000	34.0000	32.0000	1.4142
	BPSO3	36.0000	40.0000	38.1000	1.0440
	BPSO4	35.0000	44.0000	40.8000	2.7129
	BPSO5	4.0000	9.0000	6.4000	1.4283
	BPSO6	1.0000	11.0000	6.5000	2.6926
	BPSO7	2.0000	7.0000	4.8000	1.9391
	BPSO8	1.0000	6.0000	3.3000	1.4866
	BPSO10	0.0000	0.0000	0.0000	0.0000
	F_8	BPSO1	1.4420	1.5979	1.5153
BPSO2		1.6837	1.8667	1.7507	0.0513
BPSO3		1.8418	1.9625	1.9171	0.0450
BPSO4		1.9153	2.0532	1.9989	0.0400
BPSO5		0.4566	1.0066	0.7834	0.1726
BPSO6		0.6426	1.0066	0.8087	0.1022
BPSO7		0.3239	0.8457	0.6732	0.1405
BPSO8		0.3239	0.9028	0.6573	0.1738
BPSO10		8.88178E - 16	8.88178E - 16	8.8818E - 16	0.0000
F_9		BPSO1	1.9687	2.2083	2.0753
	BPSO2	2.1886	2.4374	2.3050	0.0850
	BPSO3	2.5578	2.7476	2.6494	0.0629
	BPSO4	2.5683	2.7973	2.7120	0.0743
	BPSO5	1.3666	1.6074	1.4911	0.0771
	BPSO6	1.3666	1.5472	1.4309	0.0552
	BPSO7	1.3666	1.5773	1.4690	0.0634
	BPSO8	1.3365	1.4975	1.4068	0.0486
	BPSO10	1.2108	1.2108	1.2108	0.0000
	F_{10}	BPSO1	833.4458	897.9789	859.9714
BPSO2		797.1630	894.5194	855.8127	29.2840
BPSO3		822.1992	935.9137	871.3066	31.0668
BPSO4		801.6173	922.5607	860.1476	40.3582
BPSO5		819.9967	922.3191	864.5066	32.5671
BPSO6		798.7684	903.6756	848.3581	32.5885
BPSO7		803.2907	904.1978	862.3321	35.3385
BPSO8		833.7166	898.4921	865.7664	24.2585
BPSO10		732.3788	790.8776	764.5206	20.0896
F_{11}		BPSO1	811.5518	896.1991	860.3674
	BPSO2	797.1630	905.6025	854.5382	29.9453
	BPSO3	822.1992	935.9137	862.4550	36.2997
	BPSO4	801.6173	888.0983	849.7746	28.6370
	BPSO5	819.9967	922.3191	868.3163	29.5430
	BPSO6	798.7684	903.6756	844.8258	35.6381
	BPSO7	803.2907	904.1978	865.2761	29.1488
	BPSO8	788.3447	898.6084	866.7004	37.5416
	BPSO10	732.3788	790.5087	758.0837	22.0786
	F_{12}	BPSO1	761.2894	912.1116	863.0355
BPSO2		803.8999	897.6278	850.4606	32.1798
BPSO3		835.9801	912.0372	878.0874	23.8546
BPSO4		772.1731	916.9646	858.7735	44.5248
BPSO5		765.3802	900.9199	859.8082	41.3350
BPSO6		810.4896	900.3629	856.8248	28.2321
BPSO7		820.4377	918.6237	873.2900	30.3881
BPSO8		811.3776	885.9333	852.6042	20.1360
BPSO10		693.7516	811.1792	758.9200	38.4465

TABLE 8: Nonparametric test results.

	BPSO10vsBPSO1	BPSO10vsBPSO2	BPSO10vsBPSO3	BPSO10vsBPSO4	BPSO10vsBPSO5	BPSO10vsBPSO6	BPSO10vsBPSO7	BPSO10vsBPSO8
F_1	5.721E-05	6.386E-05	6.069E-05	4.880E-05	5.893E-05	6.069E-05	5.849E-05	5.763E-05
F_2	5.470E-05	6.386E-05	5.721E-05	5.893E-05	6.158E-05	6.113E-05	6.113E-05	6.113E-05
F_3	6.386E-05	6.386E-05	6.386E-05	6.386E-05	6.386E-05	6.386E-05	6.386E-05	6.386E-05
F_4	6.386E-05	6.386E-05	6.294E-05	6.386E-05	6.386E-05	6.340E-05	6.386E-05	6.386E-05
F_5	5.470E-05	5.849E-05	5.721E-05	6.113E-05	2.238E-04	5.980E-05	6.203E-05	5.470E-05
F_6	1.827E-04	1.827E-04	1.827E-04	1.827E-04	1.827E-04	1.827E-04	1.827E-04	1.827E-04
F_7	6.113E-05	5.849E-05	5.470E-05	6.113E-05	6.113E-05	6.249E-05	6.069E-05	5.849E-05
F_8	5.511E-05	6.113E-05	5.721E-05	6.113E-05	6.294E-05	6.113E-05	6.113E-05	6.158E-05
F_9	6.340E-05	6.294E-05	6.340E-05	6.294E-05	6.249E-05	6.158E-05	6.294E-05	5.893E-05
F_{10}	1.806E-04	1.806E-04	1.806E-04	1.806E-04	1.806E-04	1.806E-04	1.806E-04	1.806E-04
F_{11}	1.827E-04	1.827E-04	1.827E-04	1.827E-04	1.827E-04	1.827E-04	1.827E-04	2.461E-04
F_{12}	1.000E-03	3.298E-04	1.827E-04	1.700E-03	1.000E-03	2.461E-04	1.827E-04	1.827E-04

TABLE 9: Comparison results between BPSO10 and BPSOGSA.

Function		BPSO10	BPSOGSA [28]
F_1	Ave	0.0000	0.7539
	Std	0.0000	0.7441
F_2	Ave	0.0000	0.1584
	Std	0.0000	0.1219
F_3	Ave	0.0000	45.2867
	Std	0.0000	94.4522
F_4	Ave	74.0000	281.4150
	Std	0.0000	667.8743
F_5	Ave	18.7500	8.0937
	Std	0.0000	17.6706
F_6	Ave	0.0044	0.0064
	Std	0.0043	0.0089
F_7	Ave	0.0000	1.8752
	Std	0.0000	1.2717
F_8	Ave	8.8818E-16	0.5412
	Std	0.0000	0.8005
F_9	Ave	1.3744	0.3702
	Std	0.0000	0.4851

TABLE 10: Comparison results of BPSO10 with BHTPSO, BHTPSO-QI, and SOGA.

Function		BPSO10	BHTPSO [29]	BHTPSO-QI [29]	SOGA [30]
F_1	Ave	0.0000	0.0010	0.0008	0.0002
	Std	0.0000	0.0029	0.0021	0.0005
F_2	Ave	0.0000	0.0080	0.0082	0.0026
	Std	0.0000	0.0154	0.0153	0.0005
F_3	Ave	0.0000	36.1500	35.0200	1919.6076
	Std	0.0000	66.3300	78.3500	1078.9650
F_5	Ave	3.0000	0.0000	0.0000	0.1000
	Std	0.0000	0.0000	0.0000	0.4026
F_6	Ave	0.0007	0.0028	0.0033	
	Std	0.0005	0.0020	0.0022	
F_8	Ave	8.8818E-16	-136.5000	-137.0000	1.3614
	Std	0.0000	1.7850	1.7240	1.1703
F_8	Ave	2.4053			1.4639
	Std	0.0000			1.4386

6. Conclusions

In this paper, a new Z-shaped transfer function is proposed to improve BPSO algorithm. The simulation results on three sets' data show that the newly proposed Z-shaped transfer function improves the convergence speed and convergence precision of BPSO algorithm. In addition, by changing the slope of the Z-shaped transfer function, it is found that when the test function dimension is low, the slope has no effect on the final convergence result, but it will affect the convergence speed. When the test function dimension is high, the slope has no effect on the convergence speed. However, it will affect the convergence result. Combined with three sets of experiments, it is found that when the parameter is set as 5, the convergence speed is the fastest and the precision is highest. In the future research, BPSO algorithm can be improved by using the new transfer function as a direction of exploration. In addition, certain family functions with different slopes are used for comparative study to find the optimal algorithm.

Data Availability

There are no data available for this paper.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this article.

Authors' Contributions

Sha-Sha Guo participated in the data collection, analysis, algorithm simulation, and draft writing. Jie-Sheng Wang participated in the concept, design, interpretation, and commented on the manuscript. Meng-Wei Guo participated in the critical revision of this paper.

Acknowledgments

This work was supported by the Basic Scientific Research Project of Institution of Higher Learning of Liaoning Province (Grant no. 2017FWDF10) and Project by Liaoning Provincial Natural Science Foundation of China (Grant no. 20180550700).

References

- [1] R. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," in *Proceedings of the Sixth International Symposium on Micro Machine and Human Science*, October 2002.
- [2] Z. Beheshti and S. M. H. Shamsuddin, "CAPSO: centripetal accelerated particle swarm optimization," *Information Sciences*, vol. 258, pp. 54–79, 2014.
- [3] Y. Wang, J. Lv, L. Zhu, and Y. Ma, "Crystal structure prediction via particle swarm optimization," *Physics*, vol. 82, no. 9, pp. 7174–7182, 2010.
- [4] B. Shan, J. Wang, L. Zhang et al., "Fitting of adaptive neuron model to electrophysiological recordings using particle swarm optimization algorithm," *International Journal of Modern Physics B*, vol. 31, no. 5, 2016.
- [5] C. H. Ram Jethmalani, S. P. Simon, K. Sundareswaran et al., "Auxiliary hybrid PSO-BPNN based transmission system loss estimation in generation scheduling," *IEEE Transactions on Industrial Informatics*, vol. 13, no. 4, pp. 1692–1703, 2016.
- [6] Z. Zhou, J. Wang, Z. Zhu, D. Yang, and J. Wu, "Tangent navigated robot path planning strategy using particle swarm optimized artificial potential field," *Optik*, vol. 158, pp. 639–651, 2018.
- [7] R. P. De Gusmao and F. D. A. T. De Carvalho, "Particle swarm optimization applied to relational data clustering," in *Proceedings of the 2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, Budapest, Hungary, October 2016.
- [8] S. Mirjalili and A. S. Sadiq, "Magnetic optimization algorithm for training multi layer perceptron," in *Proceedings of the 2011 IEEE 3rd International Conference on Communication Software and Networks*, May 2011.
- [9] S. Mirjalili, S. Z. Mohd Hashim, and H. Moradian Sardroudi, "Training feedforward neural networks using hybrid particle swarm optimization and gravitational search algorithm," *Applied Mathematics and Computation*, vol. 218, no. 22, pp. 11125–11137, 2012.

- [10] S. Mirjalili, A. Mirjalili, and A. Lewis, "Let a biogeography-based optimizer train your multi-layer perceptron," *Information Sciences*, vol. 269, no. 8, pp. 188–209, 2014.
- [11] J. Kennedy and R. C. Eberhart, "A discrete binary version of the particle swarm algorithm," in *Proceedings of the 1997 IEEE International Conference on Systems, Man, and Cybernetics. Computational Cybernetics and Simulation*, October 1997.
- [12] P.-Y. Yin, "A discrete particle swarm algorithm for optimal polygonal approximation of digital curves," *Journal of Visual Communication and Image Representation*, vol. 15, no. 2, pp. 241–260, 2004.
- [13] Z. G. Wang, H. M. Xia, M. G. Wang et al., "Improved binary particle swarm optimization algorithm for solving multidimensional knapsack problem," *Mathematics in Practice and Theory*, vol. 43, no. 19, pp. 129–137, 2013.
- [14] J. Y. Luo, "Improved binary particle swarm optimization algorithm for solving vehicle routing problem," *Mathematics in Practice and Theory*, vol. 44, no. 23, pp. 205–211, 2014.
- [15] Y. Sharafi, M. A. Khanesar, and M. Teshnehlab, "Discrete binary cat swarm optimization algorithm," in *Proceedings of the 2013 3rd IEEE International Conference on Computer, Control and Communication (IC4)*, pp. 1–6, IEEE, Karachi, Pakistan, September 2013.
- [16] S. Mirjalili, S. M. Mirjalili, and X. S. Yang, "Binary bat algorithm," *Neural Computing and Applications*, vol. 25, no. 3–4, pp. 663–681, 2014.
- [17] S. Sabba and S. Chikhi, "A discrete binary version of bat algorithm for multidimensional knapsack problem," *International Journal of Bio-Inspired Computation*, vol. 6, no. 2, p. 140, 2014.
- [18] S.-w. Fei, "Fault diagnosis of bearing based on relevance vector machine classifier with improved binary bat algorithm for feature selection and parameter optimization," *Advances in Mechanical Engineering*, vol. 9, no. 1, 2017.
- [19] E. Emary, A. E. Zawbaa, and A. E. Hassanien, "Binary grey wolf optimization approaches for feature selection," *Neurocomputing*, vol. 172, pp. 371–381, 2016.
- [20] L. K. Panwar, S. Reddy K, B. K. A. Verma, R. Panigrahi, and R. Kumar, "Binary grey wolf optimizer for large scale unit commitment problem," *Swarm and Evolutionary Computation*, vol. 38, no. 38, pp. 251–266, 2018.
- [21] A. G. Hussien, A. E. Hassanien, E. H. Houssein, M. Amin, and A. T. Azar, "New binary whale optimization algorithm for discrete optimization problems," *Engineering Optimization*, vol. 52, no. 6, pp. 945–959, 2019.
- [22] K. S. Reddy, L. Panwar, and R. Kumar, "Binary whale optimization algorithm: a new metaheuristic approach for profit-based unit commitment problems in competitive electricity markets," *Engineering Optimization*, vol. 51, no. 3, pp. 369–389, 2018.
- [23] Y. Shi and R. Eberhart, "A modified particle swarm optimizer," in *Proceedings of the 1998 IEEE International Conference on Evolutionary Computation Proceedings. IEEE World Congress on Computational Intelligence (Cat. No. 98TH8360)*, May 1998.
- [24] S. Mirjalili and A. Lewis, "S-shaped versus V-shaped transfer functions for binary particle swarm optimization," *Swarm and Evolutionary Computation*, vol. 9, pp. 1–14, 2013.
- [25] S. Lee, S. Soak, S. Oh, W. Pedrycz, and M. Jeon, "Modified binary particle swarm optimization," *Progress in Natural Science*, vol. 18, no. 9, pp. 1161–1166, 2008.
- [26] E. Rashedi, S. H. Nezamabadi-Pour, and S. Saryazdi, "BGSA: binary gravitational search algorithm," *Natural Computing*, vol. 9, no. 3, pp. 727–745, 2010.
- [27] F. A. Hashim, E. H. Houssein, M. S. Mabrouk, W. Al-Atabany, and S. Mirjalili, "Henry gas solubility optimization: a novel physics-based algorithm," *Future Generation Computer Systems*, vol. 101, pp. 646–667, 2019.
- [28] S. Mirjalili, L. d. S. Wang, and L. D. S. Coelho, "Binary optimization using hybrid particle swarm optimization and gravitational search algorithm," *Neural Computing and Applications*, vol. 25, no. 6, pp. 1423–1435, 2014.
- [29] Z. Beheshti, S. Shamsuddin, and S. Hasan, "Memetic binary particle swarm optimization for discrete optimization problems," *Information Sciences*, vol. 299, pp. 58–84, 2015.
- [30] S. Tao and X. Ming-Hai, "A swarm optimization genetic algorithm based on quantum-behaved particle swarm optimization," *Computational Intelligence and Neuroscience*, vol. 2017, Article ID 2782679, 15 pages, 2017.