Hindawi
Computational Intelligence and Neuroscience
Volume 2018, Article ID 5025672, 17 pages
https://doi.org/10.1155/2018/5025672

Research Article

Improved Particle Swarm Optimization Algorithm Based on Last-Eliminated Principle and Enhanced Information Sharing

Xueying Lv,[1] Yitian Wang,[1] Junyi Deng,[2] Guanyu Zhang ,[1,3] and Liu Zhang [1,3]

[1] *College of Instrumentation & Electrical Engineering, Jilin University, Changchun 130061, China*
[2] *College of Computer Science and Technology, Jilin University, Changchun 130022, China*
[3] *National Engineering Research Center of Geophysics Exploration Instruments, Jilin University, Changchun 130061, China*

Correspondence should be addressed to Guanyu Zhang; zhangguanyu@jlu.edu.cn and Liu Zhang; zhangliu@jlu.edu.cn

Received 18 May 2018; Revised 20 September 2018; Accepted 2 October 2018; Published 5 December 2018

Academic Editor: Cornelio Yáñez-Márquez

In this study, an improved eliminate particle swarm optimization (IEPSO) is proposed on the basis of the last-eliminated principle to solve optimization problems in engineering design. During optimization, the IEPSO enhances information communication among populations and maintains population diversity to overcome the limitations of classical optimization algorithms in solving multiparameter, strong coupling, and nonlinear engineering optimization problems. These limitations include advanced convergence and the tendency to easily fall into local optimization. The parameters involved in the imported "local-global information sharing" term are analyzed, and the principle of parameter selection for performance is determined. The performances of the IEPSO and classical optimization algorithms are then tested by using multiple sets of classical functions to verify the global search performance of the IEPSO. The simulation test results and those of the improved classical optimization algorithms are compared and analyzed to verify the advanced performance of the IEPSO algorithm.

1. Introduction

The development of industrial society has led to the successful application of the optimal design methods to diverse engineering practices, such as path planning, structural design, control theory, and control engineering [1–10]. In 1995, the foraging behavior of bird swarm inspired Kennedy and Eberhart to propose the particle swarm optimization (PSO) algorithm. PSO requires few parameter adjustments and is easy to implement; hence, it is the most commonly used swarm intelligence algorithm [11–20]. However, in practical applications, most problems are complicated design problems with multiple parameters, strong coupling, and nonlinearity. Therefore, improving the global optimization capability of an optimization algorithm is important in solving complex engineering optimization problems. To improve the capability of traditional PSO, many scholars have proposed improvement strategies, including the adjustment of parameters and combinations of various mechanisms.

Shi and Eberhant [21] proposed an inertial weight improvement strategy (SPSO) with strong global search capability at the beginning of an iteration, strong local search capability in the latter iteration, and fine search near the optimal solution. Although the SPSO improves the convergence speed of the algorithm, the "premature" phenomenon remains. Zhang [22] proposed an improved PSO algorithm with adaptive inertial weight that is based on Bayesian technology to balance the development and exploration capability of populations. Ratnawecra [23] proposed a linear adjustment method for learning factors. In the early stages of the iteration, the particle flight was mainly based on the historical information of the particle itself, and the latter particle flight was mainly based on the social information between the particle and the global optimal particle. However, this method still has defects. The best fit for the initial global search is similar to the local optimum. Moreover, convergence is only limited to some optimal regions rather than globally, thereby causing the PSO algorithm to fall into the local extrema. Chen and Ke [24] proposed a chaotic dynamic weight (CDW) PSO (CDW-PSO) algorithm. Chaotic maps and dynamic weights were used to modify the search process. Although CDW-PSO indicates an improved search performance relative to other natural heuristic optimization algorithms, it also easily falls into the local optimum. Chen [25] proposed a dynamic multiswarm differential learning PSO (DMSDL-PSO) algorithm, in which the differential evolution method is applied to each subgroup combined with a differential mutation method to conduct a global search, and a quasi-Newton method is applied for local search. The DMSDL-PSO algorithm has good exploration and exploitation capabilities. Jiang [26] proposed a new binary hybrid PSO with wavelet mutation (HPSOWM), in which the motion mechanism and mutation process of particles are converted into binary elements and the problem is transformed from a continuous space problem into a discrete domain one. Although the convergence speed of the HPSOWM algorithm is stable and robust, its convergence rate is lower than those of other intelligent optimization algorithms. To solve the dynamic multiobjective optimization problem with rapid environmental change, a study proposed a cooperative multiswarm PSO for dynamic multiobjective optimization (CMPSODMO) [27]. In comparison with other dynamic multiobjective optimization algorithms, CMPSODMO indicates a better effect in addressing uncertain rapid environmental changes. Ye [28] proposed a multiswarm PSO algorithm with dynamic learning strategies, in which the population is divided into ordinary and communication particles. The dynamic communication information of communication particles was applied to the algorithm to maintain particle population diversity. Although this method improves the capability of the algorithm to handle complex multimodal functions, it increases the computational complexity of the algorithm. Cui [29] proposed a globally optimal prediction-based adaptive mutation PSO (GPAM-PSO) to avoid the local optimal problem of traditional PSO algorithms. However, GPAM-PSO is limited to the dimensionality reduction of nonzero mean data. Zhang [30] proposed a vector covariance PSO algorithm that divides all the dimensions of a particle into several parts randomly and optimizes each part to enhance the global and local search capabilities. However, the algorithm continues to fall into the local extrema.

PSO has attracted considerable research attention due to its easy implementation, few parameter adjustments, and adaptability. Scholars use PSO to solve engineering optimization problems and gradually penetrate various fields of application, such as parameter optimization, path planning, predictive control, and global optimization. Zhao [31] used PSO to optimize wavelet neural network parameters, reduce the limitations of the assessment of network security situational awareness, and thereby meet the requirements of network security in a big data environment. The parameter-related coefficients in a nonlinear regression analysis model were optimized by combining particle swarm with a genetic phase [32] to reduce the vibrations caused by mine blasting that damages the structures around the blasting area. The derived diffusion-free PSO algorithm was used to estimate the parameters of an infinite impulse response system and improve the energy utilization of an infinite sensor network [33]. Wang [34] used a multiobjective PSO algorithm to solve a path-planning problem of mobile robots in a static rough terrain environment. Wang [35] combined PSO with chaos optimization theory to establish a mathematical model of a path-planning problem in the radioactive environment of nuclear facilities to ensure personnel safety. Lopes [36] proposed a novel particle swarm-based heuristic technique to allocate electrical loads in an industrial setting throughout the day. Multiobjective PSO was used to solve the problem of service allocation in cloud computing [37]. Petrovic [38] proposed chaos PSO to achieve an optimal process dispatching plan. Zhang [39] proposed an adaptive PSO to solve problems in reservoir operation optimization with complex and dynamic nonlinear constraints.

An improved PSO algorithm was used for a time-series prediction of a grayscale model [40]. The algorithm reduces the average relative error between the recovery and measured values of the model to avoid the problems caused by the optimization of background values. Gulcu [41] used PSO to establish a power demand forecasting model.

In view of the aforementioned methods, an improved PSO (IEPSO) algorithm is proposed in the present work. In IEPSO, the last-eliminated principle is used to update the population and maintain particle population diversity. The global search capability of the IEPSO algorithm is improved by adding local-global information sharing terms. A multigroup test function is used for comparison with the IEPSO. A classical optimization algorithm and its improved versions are used to test and verify the global optimization performance of the IEPSO algorithm.

2. IEPSO

2.1. Standard PSO. The initial population of the PSO algorithm is randomized. The IEPSO updates the position and speed of the particle swarm by adaptive learning, as shown in following formulas:

$$\begin{aligned} v_{id}^{t+1} &= \omega v_{id}^{t} + C_1 R_1 \left(p_{id}^{t} - x_{id}^{t} \right) + C_2 R_2 \left(p_{gd}^{t} - x_{id}^{t} \right), \\ x_{id}^{t} &= x_{id}^{t} + v_{id}^{t+1}, \end{aligned} \tag{1}$$

where ω is the inertial weight, C_1 and C_2 are the acceleration terms, R_1 and R_2 are the random variables uniformly distributed in the range of (0, 1), P_g^t is the global better position, P_i^t is the particle that finds the best position in history, x_i^d is the particle position in the current iteration, and v_{id}^{t+1} is the particle update speed at the next iteration.

2.2. IEPSO. The IEPSO algorithm is mainly based on the last-eliminated principle and enhances the local-global information sharing capability to improve its global optimization performance. The specific implementation of the IEPSO algorithm is shown in Figure 1.

The position and velocity of particles in a population are randomly initialized, and the fitness value of the particles is calculated. Information on the current individual and global optimal particles, including their positions and fitness values, is saved. Then, the particle swarm operation is conducted. In the IEPSO algorithm, Formula (2) is used to update the speed to balance the exploration and exploitation capabilities of the particles in the global optimization process. Formula (3) is the local-global information sharing term:

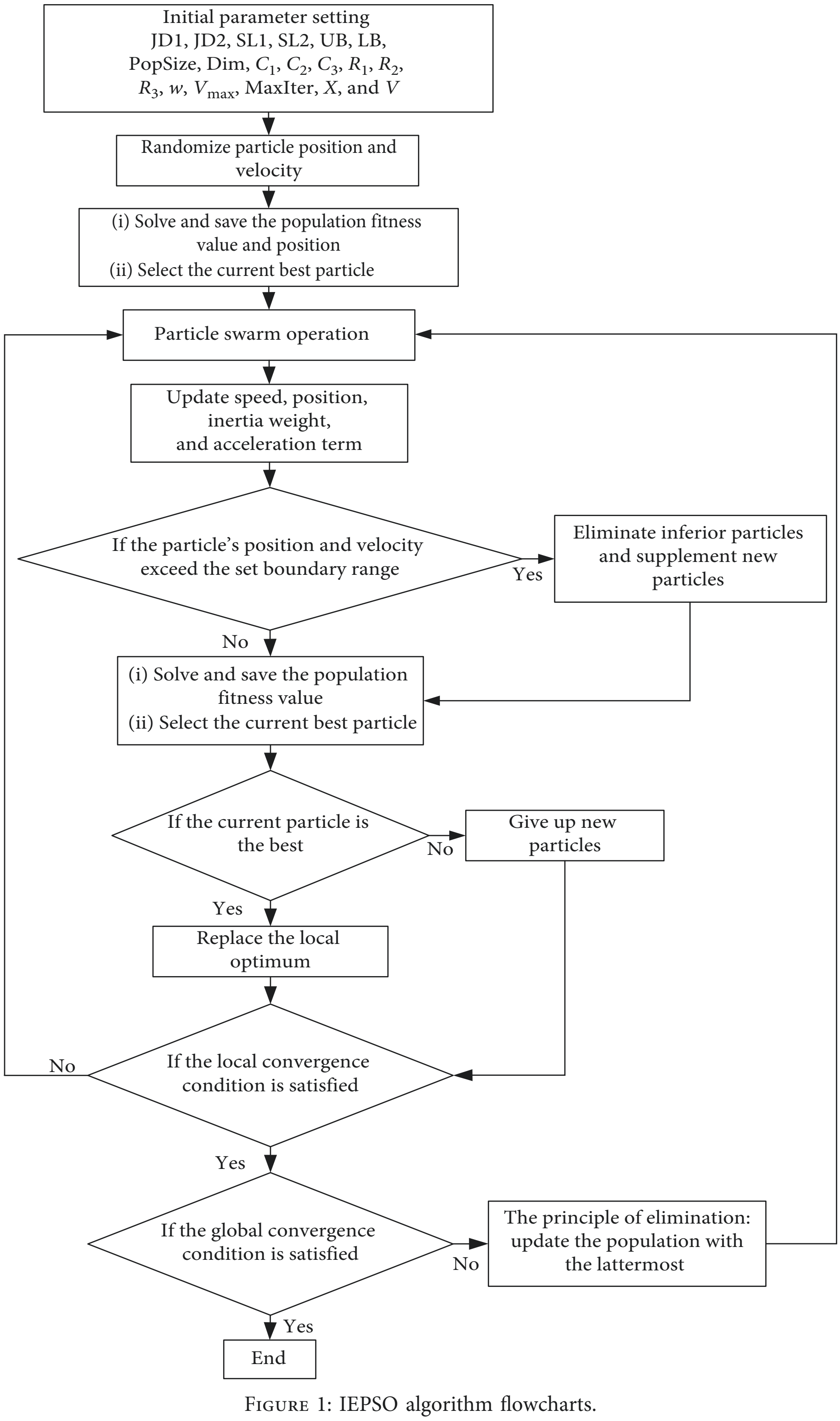

Initial parameter setting
JD1, JD2, SL1, SL2, UB, LB, PopSize, Dim, C_1, C_2, C_3, R_1, R_2, R_3, w, $V_{\max}$, MaxIter, X, and V
Randomize particle position and velocity
(i) Solve and save the population fitness value and position
(ii) Select the current best particle
Particle swarm operation
Update speed, position, inertia weight, and acceleration term
If the particle's position and velocity exceed the set boundary range
Yes
Eliminate inferior particles and supplement new particles
No
(i) Solve and save the population fitness value
(ii) Select the current best particle
If the current particle is the best
No
Give up new particles
Yes
Replace the local optimum
No
If the local convergence condition is satisfied
Yes
If the global convergence condition is satisfied
No
The principle of elimination: update the population with the lattermost
Yes
End

Figure 1: IEPSO algorithm flowcharts.

$$v_{id}^{t+1} = \omega v_{id}^{t} + C_1R_1\left(p_{id}^{t} - x_{id}^{t}\right) + C_2R_2\left(p_{gd}^{t} - x_{id}^{t}\right) + C_3R_3\left|p_{gd}^{t} - p_{id}^{t}\right|, \qquad (2)$$

$$\varphi_3 = C_3R_3\left|p_{gd}^{t} - p_{id}^{t}\right|. \qquad (3)$$

Formula (2) comprises four parts, namely, the inheritance of the previous speed, particle self-cognition, local information sharing, and "local-global information sharing."

The IEPSO algorithm is not limited to one-way communication between global and individual particles. The local-global information sharing term (φ_3) is added to the information exchange between the local optimum and global optimal particles obtained by the current iteration, and the population velocity is updated by Formula (2). In the early stage of the algorithm, the entire search space is searched at a relatively high speed to determine the approximate range of the optimal solution; the result is beneficial for global search. In the latter stage, most of the particle search space is gradually reduced and concentrated in the neighborhood of the optimal value for deep search; the result is beneficial for local search.

The particles that have not exceeded the predetermined range after the speed update continue to retain their original speed. The maximum value of the velocity is assigned to the particle that is beyond the predetermined range after the speed is updated. The particles that have not exceeded the predetermined range after the location update continue to retain their original positions. When the particles are beyond the predetermined range, inferior particles are eliminated by adding new particles to the population within the predetermined range, thereby forming a new population. The fitness value of the new population is recalculated, and the information of the individual particle and its global optimal position and fitness value obtained by the current iteration are preserved. In all the algorithms, particles have good global search capability at the beginning of the iteration, and as individual particles move closer to the local optimal particle, the algorithms gradually lose particle diversity. On the basis of the idea of population variation of the traditional genetic algorithm (GA), the last-eliminated principle is applied in the IEPSO algorithm to maintain particle population diversity. When the PSO satisfies the local convergence condition, the optimal value obtained at this time may be the local optimal value. Particle population diversity is maintained by using the particle fitness function as the evaluation criterion, thereby eliminating particles with poor fitness or high similarity. New particles are added to a new species in a predetermined range, and the particle swarm operations are reexecuted. If the number of the current iteration reaches the required predefined convergence accuracy, the iteration is stopped, and the optimal solution is produced. The complexity and runtime of the algorithm increase due to the increased local-global information sharing and the last-eliminated principle. Nevertheless, experimental results show that the improved method can enhance the accuracy of the algorithm.

3. Experimental Study

Eleven test functions are adopted in this study to test the performance of the proposed IEPSO. In this test, f_1–f_5 are unimodal functions, whereas f_6–f_{11} are multimodal functions. f_6 (Griewank) is a multimodal function with multiple local extrema, in which achieving the theoretically global optimum is difficult. f_7 (Rastrigin) possesses several local minima, in which finding the global optimal value is difficult. f_{10} (Ackley) is an almost flat area modulated by a cosine wave to form a hole or a peak; the surface is uneven, and entry to a local optimum during optimization is easy. f_{11} (Cmfun) possesses multiple local extrema around the global extremum point, and falling into the local optimum is easy. Table 1 presents the 11 test functions, where D is the space dimension, S is the search range, and CF is the theoretically optimal value.

3.1. Parameter Influence Analysis of Local-Global Information Sharing Term.

This study proposes the addition of a local-global information sharing term, which involves the parameter C_3. Therefore, the following exploration is conducted in a manner in which C_3 is selected by using the 11 test functions.

(1) When C_3 takes a constant value, constant 2 is selected.

(2) The linear variation formula of C_3 is as follows:

$$C_3 = k\left(C_3\_start - (C_3\_\text{start} - C_3\_\text{end}) \times \frac{t}{t_{\max}}\right), \quad (4)$$

where k is the control factor. When $k = 1$, C_3 is a linearly decreasing function; when $k = -1$, C_3 is a linearly increasing function. C_3_start and C_3_end are the initial and termination values of C_3, respectively. T is the iteration number, and $t_{\max}$ is the maximum number of iterations.

Tables 2 and 3 and Figure 2 show that C_3 is a constant that linearly declines and linearly increases in three cases. When the parameter C_3 of the local-global information sharing term is a linearly decreasing function, the average fitness value of the testing function is optimal, and the convergence speed and capability to jump out of the local extrema are higher than those in the other two cases. When C_3 takes a constant, the algorithm cannot balance the global and local search, resulting in a "premature" phenomenon. When C_3 adopts the linearly decreasing form, the entire area can be quickly searched at an early stage, and close attention is paid to local search in the latter part of the iteration to enhance the deep search ability of the algorithm. While C_3 adopts a linearly increasing form, it focuses on the global-local information exchange in the latter stage of the iteration. Although this condition can increase the deep search ability of the algorithm, it will cause the convergence speed to stagnate. Therefore, compared with the linearly increasing form, the linearly decreasing form shows a simulation curve that converges faster and with higher precision.

Therefore, the selection rules of the parameter C_3 of local-global information sharing in a decreasing function are investigated in this study. The nonlinear variation formula of C_3 is as follows:

$$C_3 = (C_3\_\text{start} - C_3\_\text{end}) \times \tan\left(0.875 \times \left(1 - \frac{t}{t_{\max}}\right)^k\right) + C_3\_\text{end}, \quad (5)$$

where C_3_start and C_3_end are the initial and termination values of the acceleration term C_3, respectively, and k is the control factor. When $k = 0.2$, C_3 is a convex decreasing function; when $k = 2$, C_3 is a concave decreasing function. t is the iteration number, and $t_{\max}$ is the maximum number of iterations.

Table 1: 11 test functions.

No.	Test function	S	CF
f_1	Sphere: $f_1(x) = \sum_{i=1}^{D} x_i^2$ [10]	$[-100,100]^D$	0
f_2	Schaffer: $f(x, y) = 0.5 + (\sin^2\sqrt{x^2 + y^2} - 0.5)/(1 + 0.001(x^2 + y^2))^2$ [33]	$[-100,100]^D$	0
f_3	Step: $f_3(x) = \sum_{i=1}^{D} [x_i + 0.5]^2$ [10]	$[-100,100]^D$	0
f_4	SumSquares: $f_4(x) = \sum_{i=1}^{D} i x_i^2$ [10]	$[-10,10]^D$	0
f_5	Zakharov: $f_5(x) = \sum_{i=1}^{D} x_i^2 + (\sum_{i=1}^{D} 0.5 \times i x_i^2)^2 + (\sum_{i=1}^{D} 0.5 \times i x_i)^4$ [10]	$[-100,100]^D$	0
f_6	Griewank: $f_6(x) = 1/4000 \sum_{i=1}^{D} x_i^2 - \prod_{i=1}^{D} \cos(x_i/\sqrt{i} + 1)$ [10]	$[-600,600]^D$	0
f_7	Rastrigin: $f_7(x) = \sum_{i=1}^{D} [x_i^2 - 10\cos(2\pi x_i + 10)]$ [10]	$[-5.12,5.12]^D$	0
f_8	Alpine: $f_8(x) = \sum_{i=1}^{D} (\lvert x_i \sin x_i \rvert + 0.1 x_i)$ [6]	$[-10,10]^D$	0
f_9	Shubert: $\min f_9(x, y) = \left\{\sum_i^5 i\cos[(i + 1)x + i]\right\} \times \left\{\sum_i^5 i\cos[(i + 1)y + i]\right\}$	$[-10,10]^D$	−186.731
f_{10}	Ackley: $f_{10}(x) = -20\exp(-0.2\sqrt{1/D\sum_{i=1}^{D} x_i^2}) - \exp(1/D\sum_{i=1}^{D}\cos 2\pi x_i) + 20 + e$ [10]	$[-32,32]^D$	0
f_{11}	Cmfun: $f_{11}(x, y) = x\sin(\sqrt{\lvert x \rvert}) + y\sin(\sqrt{\lvert y \rvert})$	[−500,500]	−837.966

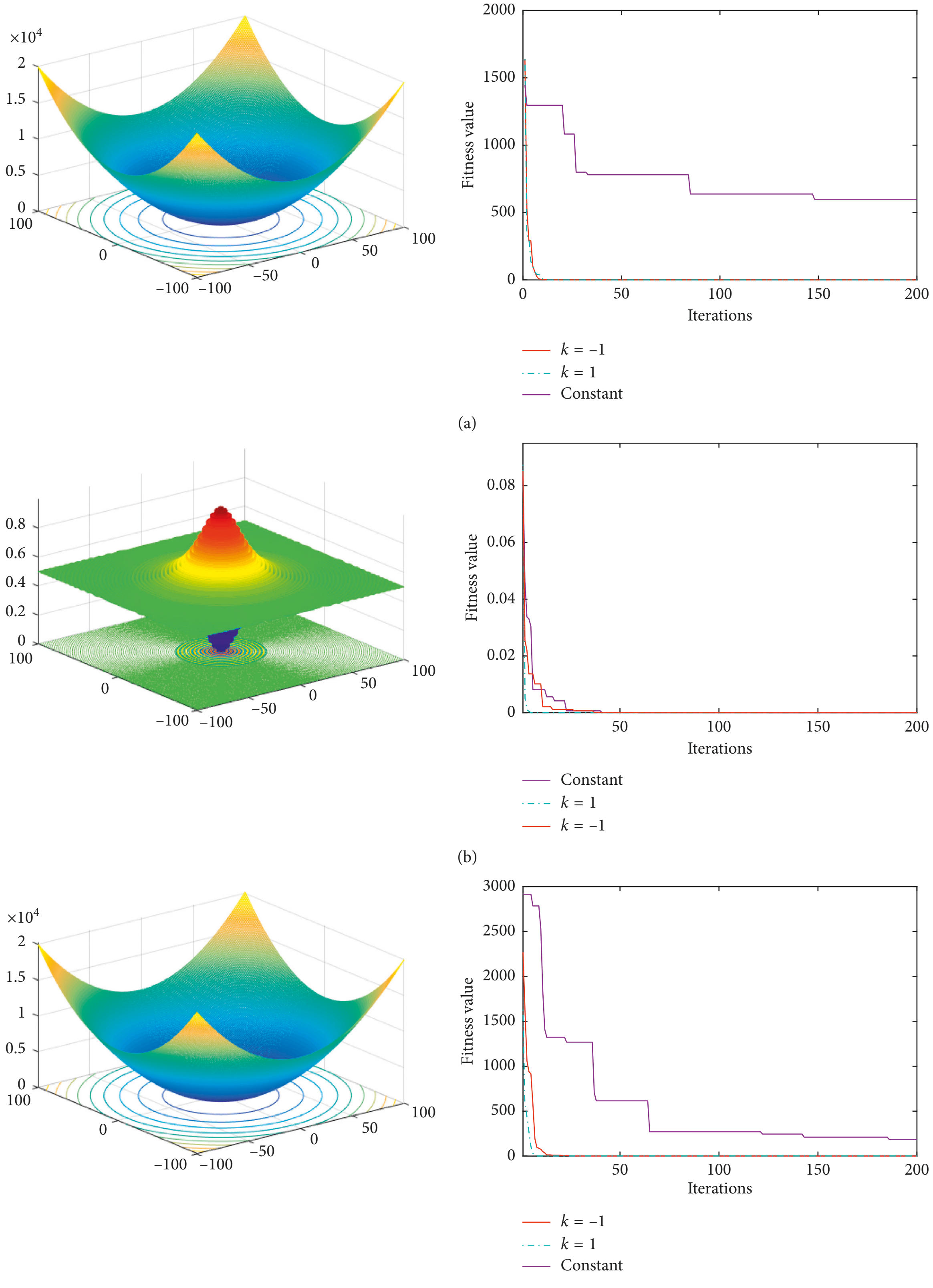
Table 2: Unimodal test functions.

Functions	Criteria	$C_3 = 2$	$C_3 = 2{\sim}0\ k = -1$	$C_3 = 2{\sim}0\ k = 1$
f_1	Mean	7.22E + 02	1.07E − 06	**4.50E − 20**
	SD	3.97E + 04	1.11E − 12	**3.75E − 16**
	Best	4.05E + 02	2.41E − 08	**1.55E − 25**
f_2	Mean	2.50E − 06	2.22E − 17	**0**
	SD	2.32E − 12	2.59E − 33	**0**
	Best	2.85E − 07	0	**0**
f_3	Mean	1.99E + 02	8.03E − 07	**1.82E − 20**
	SD	2.15E + 04	1.04E − 12	**1.05E − 39**
	Best	35.81	3.95E − 08	**3.22E − 24**
f_4	Mean	7.110	2.45E − 08	**8.20E − 20**
	SD	9.57	7.95E − 16	**5.11E − 38**
	Best	1.47	4.08E − 09	**8.43E − 26**
f_5	Mean	1.74E + 03	3.86E − 04	**5.56E − 11**
	SD	2.44E + 05	1.49E − 07	**4.88E − 14**
	Best	8.29E + 02	9.78E − 06	**3.54E − 11**

Table 3: Multimodal test functions.

Functions	Criteria	$C_3 = 2$	$C_3 = 2{\sim}0\ k = -1$	$C_3 = 2{\sim}0\ k = 1$
f_6	Mean	1.10	8.18E − 02	**4.92E − 02**
	SD	4.6E − 03	8.37E − 04	**5.96E − 04**
	Best	0.96	4.33E − 02	**1.23E − 02**
f_7	Mean	35.03	4.10	**1.9Ee − 04**
	SD	8.44	2.5461	**5.649**
	Best	29.67	2.057	**2.25E − 05**
f_8	Mean	2.93	1.33E − 03	**5.28E − 10**
	SD	0.30	3.10E − 08	**2.23E − 12**
	Best	2.02	1.34E − 05	**5.83E − 13**
f_9	Mean	−186.7295	**−186.7309**	**−186.7309**
	SD	1.20E − 06	**0**	**0**
	Best	−186.7307	**−186.7309**	**−186.7309**
f_{10}	Mean	7.649	2.35E − 04	**1.84E − 11**
	SD	0.415	4.39E − 09	**2.27E − 22**
	Best	6.513	5.73E − 05	**2.50E − 12**
f_{11}	Mean	**−837.9658**	**−837.9658**	**−837.9658**
	SD	**4.50E − 09**	**0**	**4.40E − 09**
	Best	**−837.9658**	**−837.9658**	**−837.9658**

Table 4 shows that when C_3 is a convex function, the precision and robustness of the algorithm can obtain satisfactory results on f_1–f_5. Table 5 shows that when C_3 is a convex function, the algorithm obtains a satisfactory solution and shows a fast convergence rate on f_6, f_8, f_9, f_{10}, and f_{11}. In the unimodal test function, the IEPSO algorithm does

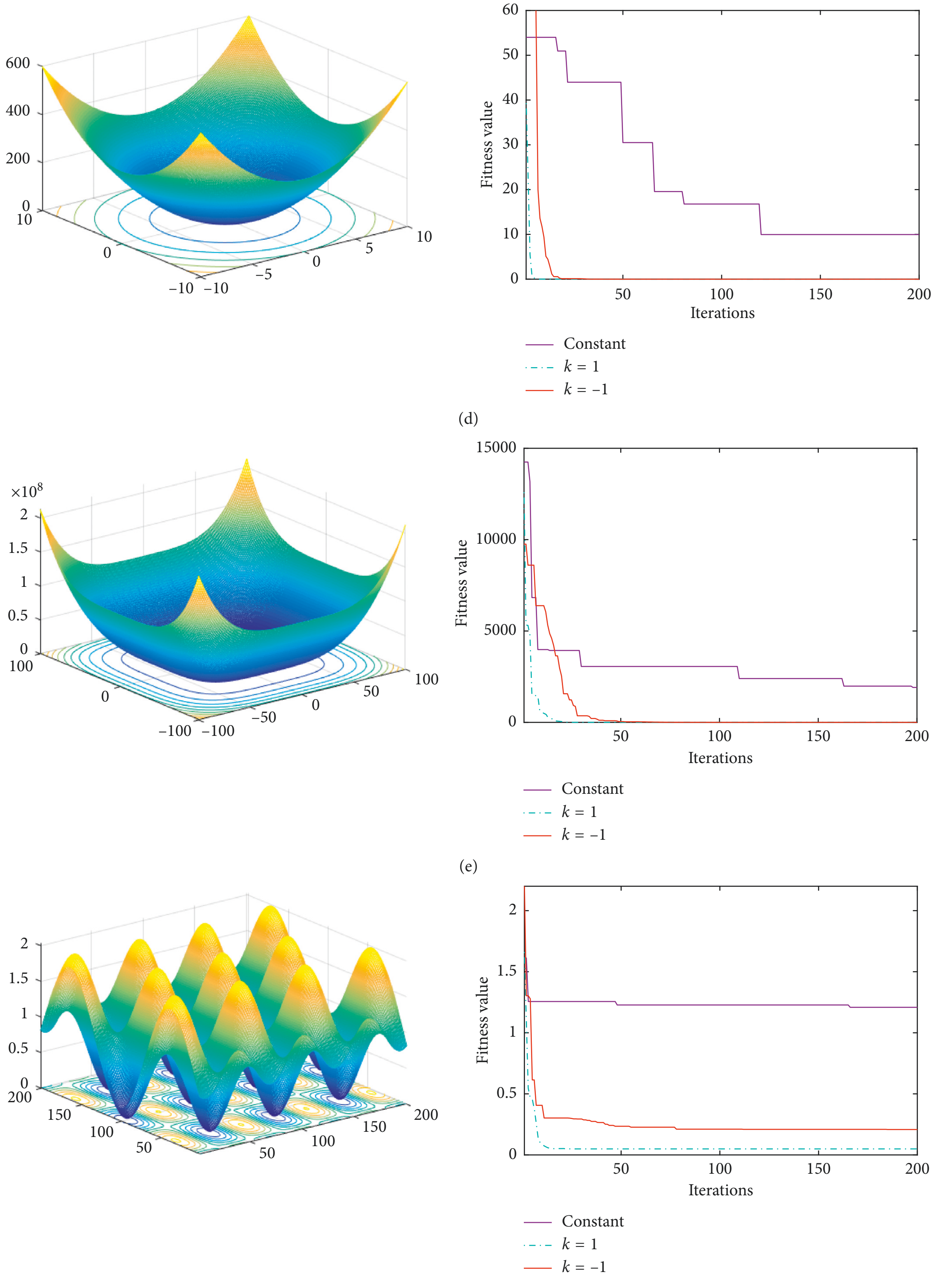


(a)

(b)

(c)

Figure 2: Continued.

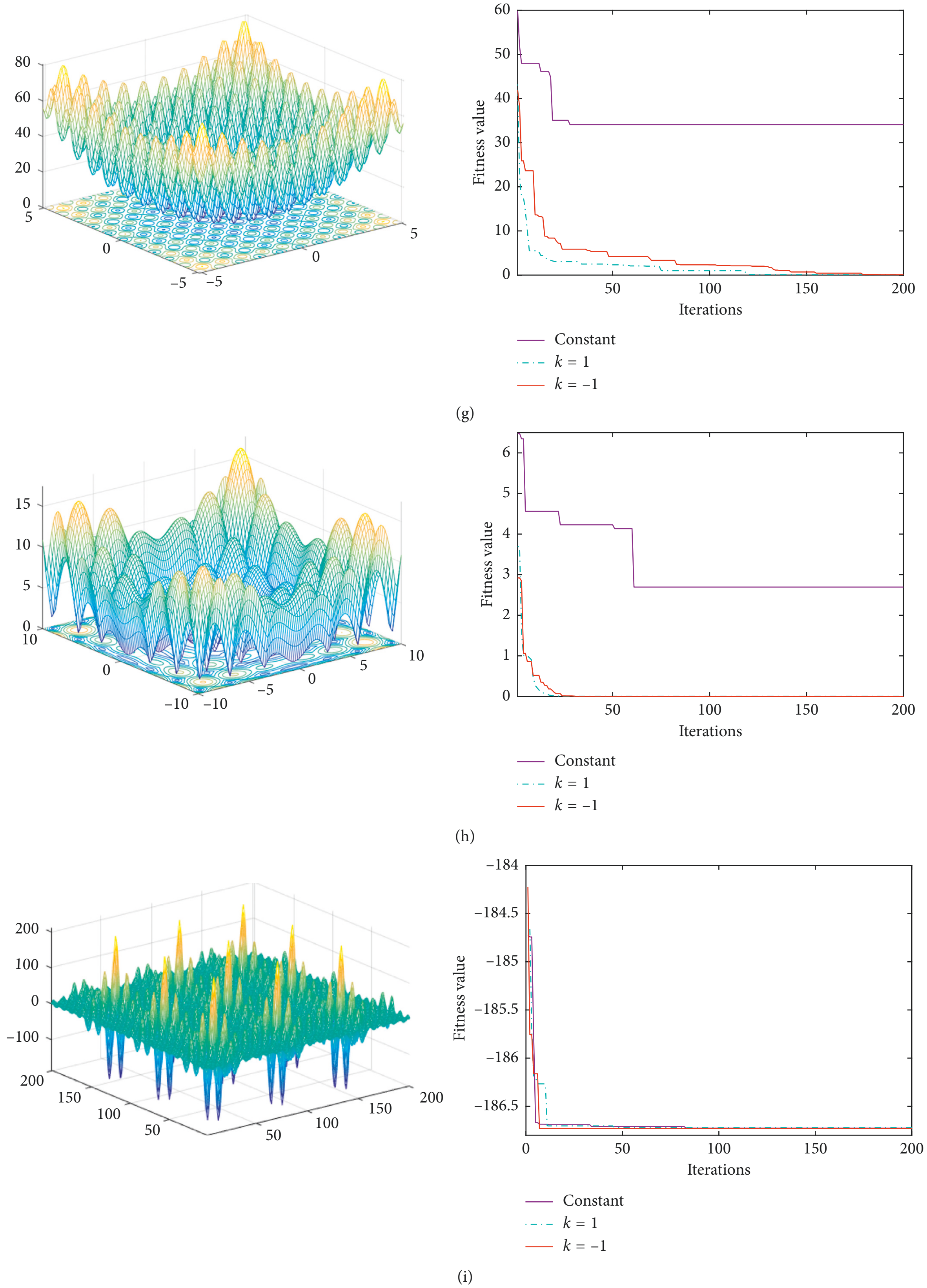


(d)

(e)

(f)

Figure 2: Continued.

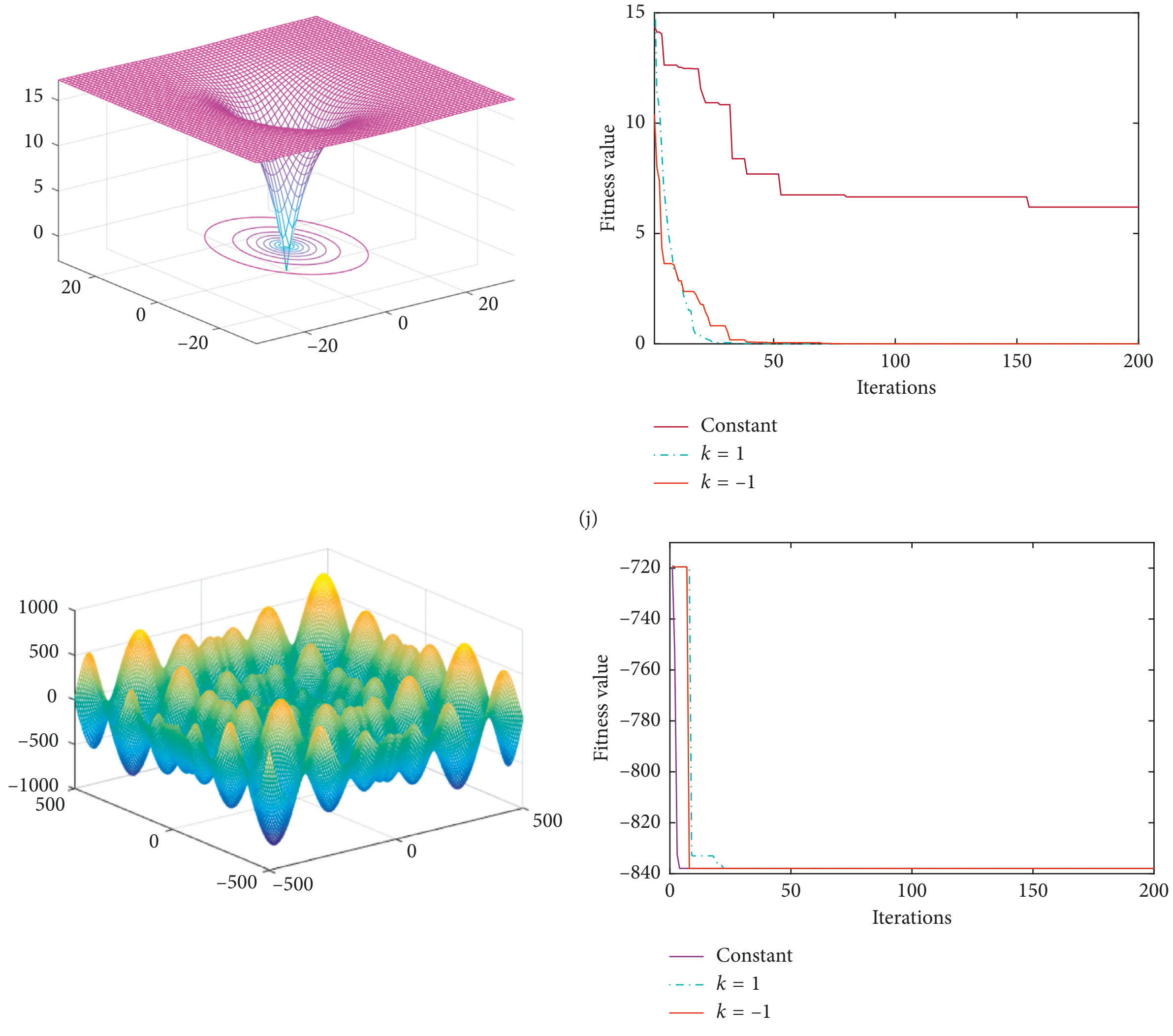


(g)

(h)

(i)

Figure 2: Continued.



(j)

(k)

Figure 2: 11 test functions: (a) f_1 sphere function; (b) f_2 Schaffer function; (c) f_3 step function; (d) f_4 SumSquares function; (e) f_5 Zakharov function; (f) f_6 Griewank function; (g) f_7 Rastrigin function; (h) f_8 alpine function; (i) f_9 Shubert function; (j) f_{10} Ackley function; (k) f_{11} Cmfun function.

Table 4: Unimodal test functions.

Functions	Criteria	C_3 = 2~0 k = 0.2	C_3 = 2~0 k = 2	C_3 = 2~0 k = 1
f_1	Mean	**2.66*E* − 20**	5.51*E* − 10	4.50*E* − 20
	SD	**2.65*E* − 39**	2.87*E* − 19	3.75*E* − 16
	Best	**9.12*E* − 24**	1.38*E* − 11	1.55*E* − 25
f_2	Mean	**0**	**0**	**0**
	SD	**0**	**0**	**0**
	Best	**0**	**0**	**0**
f_3	Mean	6.21*E* − 19	6.04*E* − 10	**1.82*E* − 20**
	SD	2.63*E* − 36	7.79*E* − 19	**1.05*E* − 39**
	Best	1.81*E* − 27	3.08*E* − 11	**3.22*E* − 24**
f_4	Mean	**1.70*E* − 21**	2.42*E* − 11	8.20*E* − 20
	SD	**1.31*E* − 41**	4.40*E* − 22	5.11*E* − 38
	Best	**2.82*E* − 29**	4.36*E* − 12	8.43*E* − 26
f_5	Mean	1.65*E* − 10	2.83*E* − 11	**5.56*E* − 11**
	SD	3.30*E* − 20	3.59*E* − 11	**4.88*E* − 14**
	Best	2.17*E* − 11	1.00*E* − 11	**3.54*E* − 11**

Table 5: Multimodal test functions.

Functions	Criteria	C_3 = 2~0 k = 0.2	C_3 = 2~0 k = 2	C_3 = 2~0 k = 1
f_6	Mean	**4.19*E* − 02**	4.79*E* − 02	4.92*E* − 02
	SD	**3.43*E* − 04**	7.07*E* − 04	5.96*E* − 04
	Best	**1.25*E* − 02**	5.7*E* − 03	1.23*E* − 02
f_7	Mean	4.46*E* − 03	**5.00*E* − 05**	1.9*Ee* − 04
	SD	1.73*E* − 04	**3.03*E* − 06**	5.649
	Best	2.31*E* − 12	**3.89*E* − 11**	2.25*E* − 05
f_8	Mean	**2.42*E* − 10**	3.74*E* − 10	5.28*E* − 10
	SD	**6.74*E* − 20**	2.47*E* − 12	2.23*E* − 12
	Best	**3.71*E* − 16**	4.36*E* − 11	5.83*E* − 13
f_9	Mean	**−186.7309**	**−186.7309**	**−186.7309**
	SD	**0**	**0**	**0**
	Best	**−186.7309**	**−186.7309**	**−186.7309**
f_{10}	Mean	**1.13*E* − 11**	2.05*E* − 10	1.84*E* − 11
	SD	**2.21*E* − 22**	4.37*E* − 12	2.27*E* − 22
	Best	**5.06*E* − 14**	1.75*E* − 10	2.50*E* − 12
f_{11}	Mean	**−837.9658**	**−837.9658**	**−837.9658**
	SD	**0**	**0**	**4.40*E* − 09**
	Best	**−837.9658**	**−837.9658**	**−837.9658**

not show its advantages because of its strong deep search capability. In the complex multimodal test function, when the convex function is used in C_3, the downward trend is slow in the early stage, thus benefiting the global search, and the downward speed increases in the later stage, thus benefiting the local search. When the concave function is used for C_3, the descent speed is fast in the early stage. Although the search speed is improved, the coverage area of the search is reduced, thereby leading to the convergence of the algorithm to the nonoptimal value. From the simulation diagrams (f)–(k), the convergence speed is observed to be slightly slow when C_3 is a convex function, but its ability to jump out of the local extremum and the accuracy of the global search are higher than those in the other two cases. When C_3 is a concave function, the convergence speed is faster than those in the other two cases, and the search accuracy is lower than that when C_3 is a convex function.

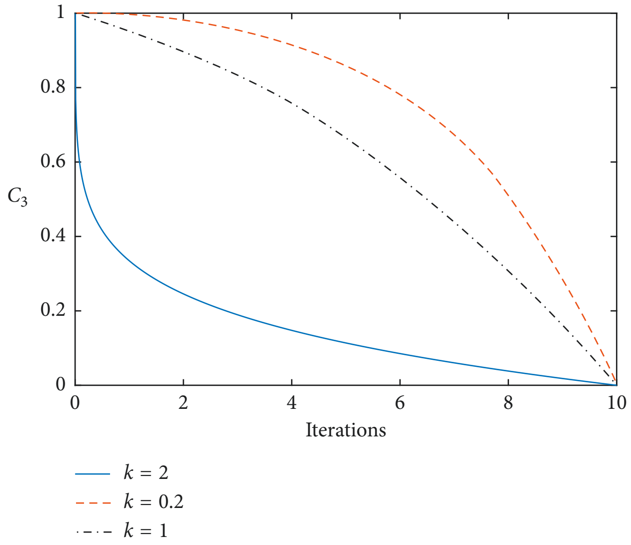
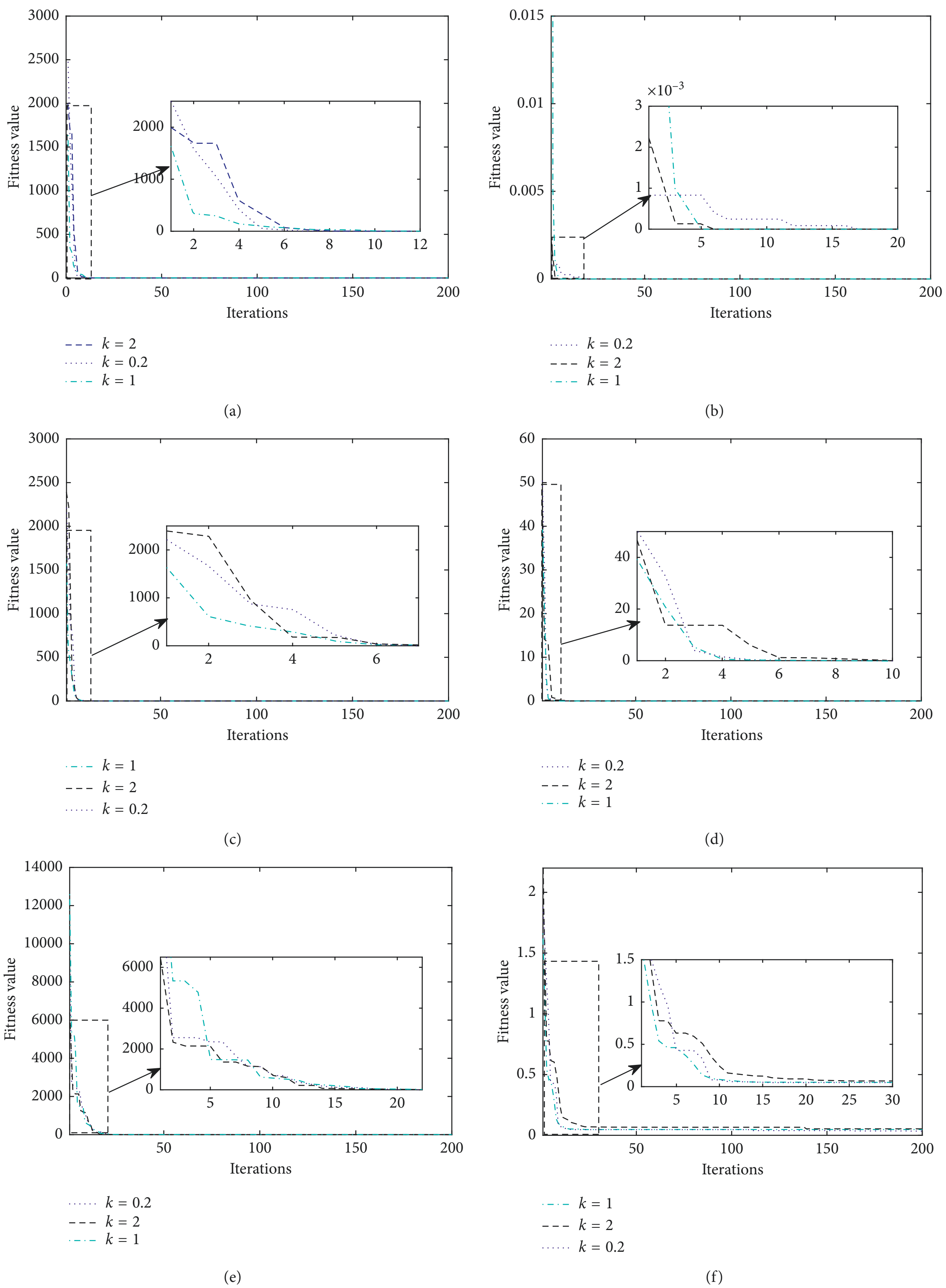


Figure 3: The change curve of C_3 with the number of iterations.

3.2. Comparison of Test Results. The 11 test functions in Figure 1 are used to compare the IEPSO algorithm with classical PSO, SPSO, differential algorithm (DE), and GA. The DE, GA, and PSO algorithms are all stochastic intelligent optimization algorithms with population iterations. The evaluation criteria of algorithm performance include speed of convergence and size of individual population search coverage. The differential optimization algorithm has a low space complexity and obvious advantages in dealing with large-scale and complex optimization problems. The GA has good convergence when solving discrete, multipeak, and noise-containing optimization problems. Based on the traditional PSO algorithm, the SPSO algorithm achieves the balance between global search and local search by adjusting the inertial weight (Figures 3 and 4).

The experimental parameters of the five algorithms are set, as shown in Table 6. Each test function is run independently 10 times, and the average is recorded to reduce the data error. The iteration is stopped when the convergence condition meets the convergence accuracy. The best average fitness value of the five algorithms is blackened. The standard deviation, average fitness, and optimal value of each algorithm are shown in Tables 7 and 8; Figures 5 and 6 plot the convergence curves of the 11 test functions.

Table 7 shows that the IEPSO has the best performance on f_1, f_2, f_3, and f_4. The IEPSO algorithm obtains the theoretical optimal value on f_2. DE can search the global solution on f_5. The deep search capability of the IEPSO algorithm is considerably higher than that of the PSO and SPSO algorithms due to the increased global-local information sharing term and the last-eliminated principle. The crossover, mutation, and selection mechanisms make the DE algorithm perform well in the early stage of the global search. However, the diversity of the population declines in the latter stage because of population differences. The



(a)

(b)

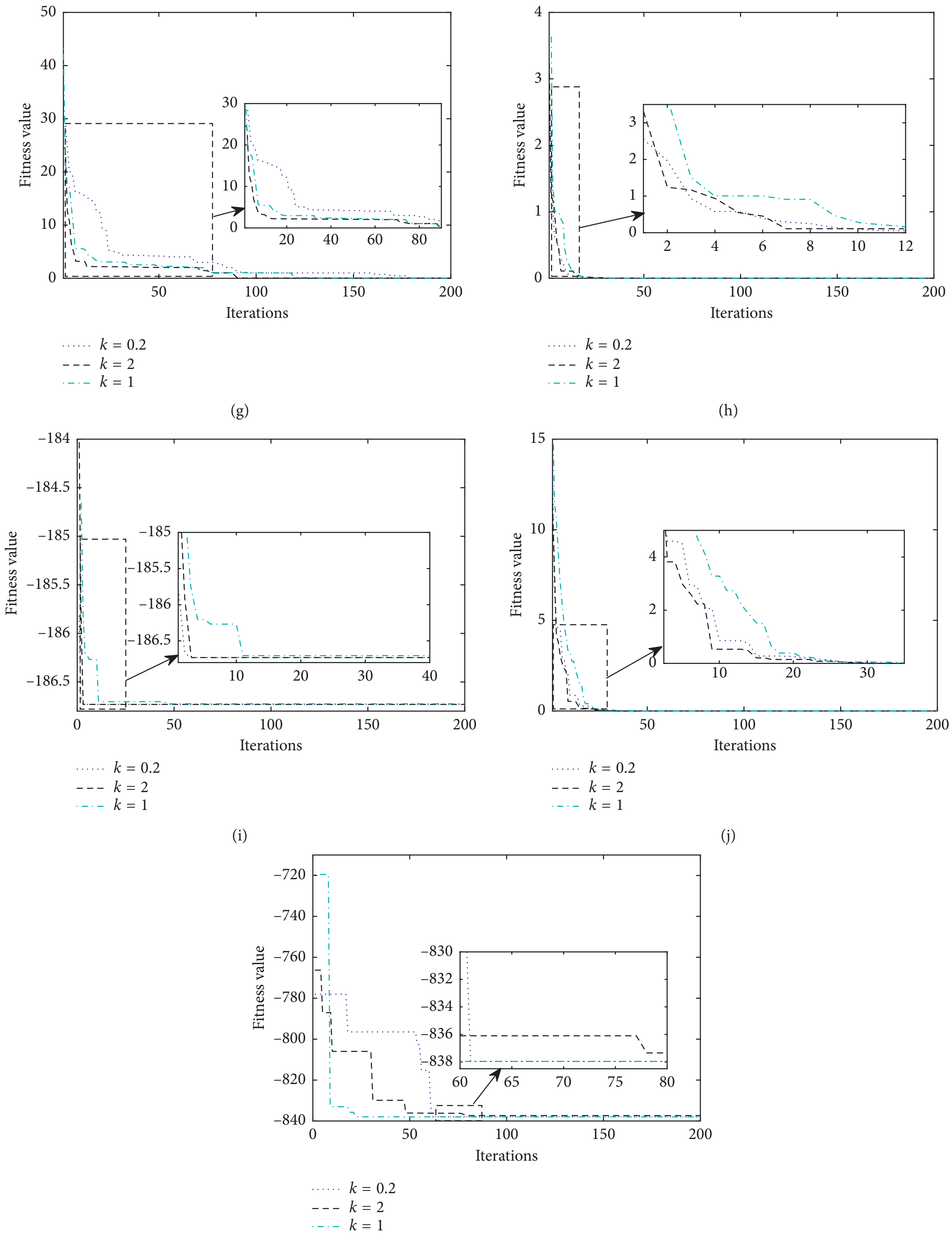
(c)

(d)

(e)

(f)

Figure 4: Continued.



(g)

(h)

(i)

(j)

(k)

Figure 4: 11 test functions: (a) f_1 sphere function; (b) f_2 Schaffer function; (c) f_3 step function; (d) f_4 SumSquare function; (e) f_5 Zakharov function; (f) f_6 Griewank function; (g) f_7 Rastrigin function; (h) f_8 alpine function; (i) f_9 Shubert function; (j) f_{10} Ackley function; (k) f_{11} Cmfun function.

Table 6: Parameter settings.

Algorithm	Population	Maximum iteration	Dim of each object	Others
PSO	40	1000	10	$C_1 = C_2 = 2$; $R_1 = R_2 = 0.5$
SPSO	40	1000	10	$\omega = 0.9$–0.4; $C_1 = C_2 = 2$; $R_1 = R_2 = 0.5$
DE	40	1000	10	—
GA	40	1000	10	$GGAP = 0.5$; $PRECI = 25$
IEPSO	40	1000	10	$\omega = 0.9$–0.4; $C_1 = C_2 = 2$; $C_3 = 2$–0; $R_1 = R_2 = R_3 = 0.5$

Table 7: Unimodal test functions.

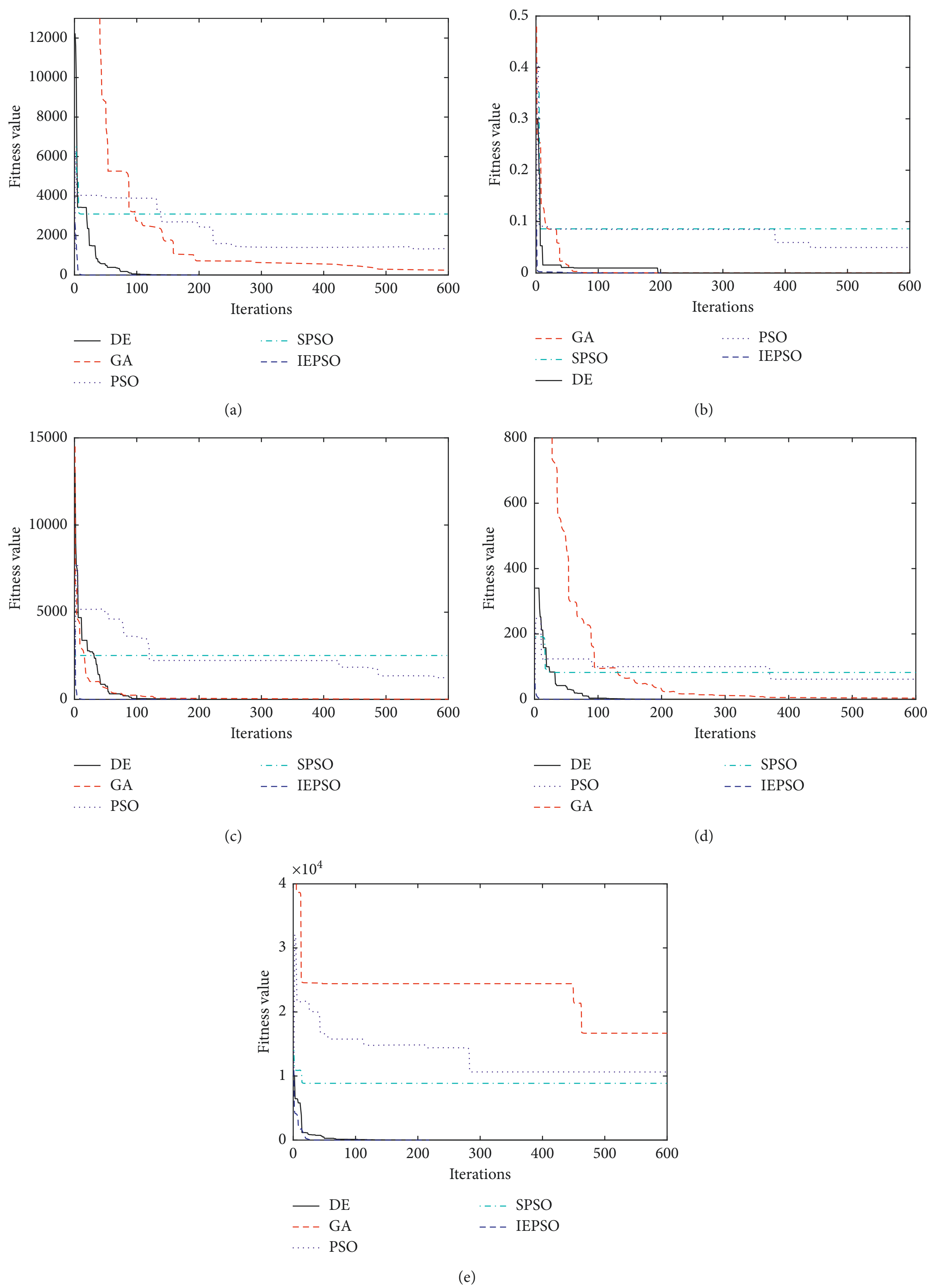
Functions	Criteria	PSO	SPSO	DE	IEPSO	GA
f_1	Mean	1.33E + 03	3.08E + 03	7.31E − 12	**8.92E − 22**	11.696
	SD	2.53E + 05	1.21E + 06	2.25E − 23	**2.65E − 39**	44.192
	Best	1.14E + 03	1.20E + 03	2.42E − 12	**7.72E − 27**	4.660
f_2	Mean	2.96E − 02	8.80E − 02	8.37E − 06	**0**	1.79E − 11
	SD	8.36E − 04	8.96E − 04	1.58E − 10	**0**	0
	Best	4.55E − 03	8.428734	7.55E − 10	**0**	1.79E − 11
f_3	Mean	1.19E + 03	2.51E + 03	1.14E − 11	**6.21E − 19**	7.430
	SD	2.93E + 05	1.82E + 06	9.95E − 23	**2.63E − 36**	5.833
	Best	1.06E + 03	2.82E − 02	2.10E − 12	**1.81E − 27**	4.542
f_4	Mean	82.38	82.10	3.36E − 13	**1.70E − 21**	3.031
	SD	6.86E + 02	1.40E + 03	9.95E − 26	**1.31E − 41**	0.835
	Best	1.15E + 02	37.39	1.15E − 13	**2.82E − 29**	1.968
f_5	Mean	1.26E + 04	8.60E + 03	**7.02E − 12**	1.65E − 10	3.62E + 03
	SD	2.06E + 07	2.15E + 07	**1.81E − 23**	3.30E − 20	3.44E + 05
	Best	1.04E + 04	1.30E + 02	**2.67E − 12**	2.17E − 11	2.53E + 03

Table 8: Multimodal test functions.

Functions	Criteria	PSO	SPSO	DE	IEPSO	GA
f_6	Mean	1.548	1.752	9.44E − 02	**4.19E − 02**	1.006
	SD	0.026	0.093	4.87E − 04	**3.43E − 04**	0.018
	Best	1.236	1.417	0.06	**0.013**	0.794
f_7	Mean	57.737	43.405	11.945	**4.46E − 03**	8.939
	SD	117.768	65.178	16.502	**1.73E − 04**	3.608
	Best	35.981	3.17E + 01	6.398	**2.31E − 12**	5.040
f_8	Mean	4.996	4.665	3.79E − 02	**2.42E − 10**	0.423
	SD	1.91E + 00	1.056	5.4E − 03	**6.74E − 20**	0.051
	Best	2.933	3.151	4.6E − 03	**3.71E − 16**	0.086
f_9	Mean	−186.448	−186.048	−186.728	**−186.731**	**−186.731**
	SD	1.19E − 01	9.83E − 01	2.29E − 08	**0**	**9.99E − 12**
	Best	−1.87E + 02	−186.731	−186.7309	**−186.7309**	**−186.731**
f_{10}	Mean	13.134	15.560	1.613	**1.13E − 11**	2.515
	SD	14.260	2.163	0	**2.21E − 22**	0.166
	Best	2.861	12.719	1.613	**5.06E − 14**	1.796
f_{11}	Mean	−740.326	−715.438	**−837.966**	**−837.966**	**−837.966**
	SD	8.74E + 03	7.23E + 03	**0**	**0**	**0**
	Best	−837.966	−837.697	**−837.966**	**−837.966**	**−837.966**

simulation diagrams (a)–(e) show that although the DE algorithm converges rapidly in the early stage, its global search performance in the later stage becomes lower than that of the IEPSO algorithm. When the GA is used to solve optimization problems, the individuals in the population fall into the local optimum and do not continue searching for the optimum solution. Therefore, in Figure 5, the simulation curve of the GA converges to the local optimum.

The test results in Table 8 indicate that the IEPSO has the best performance on f_6, f_7, f_8, f_9, f_{10}, and f_{11} and that the DE and GA can obtain the theoretical optimal value on f_9 and f_{11}. Although the GA and IEPSO algorithm can obtain the global optimal value on f_9, the IEPSO algorithm is more robust than the GA is. As shown in the simulation curve of Figure 6, the diversity of the population is maintained because the supplementary particles in the population are stochastic when the local optimal solution converges gradually. The IEPSO algorithm can jump out of the local extrema points in the face of complex multimodal test functions, and the number of iterations required is correspondingly reduced.



(a)

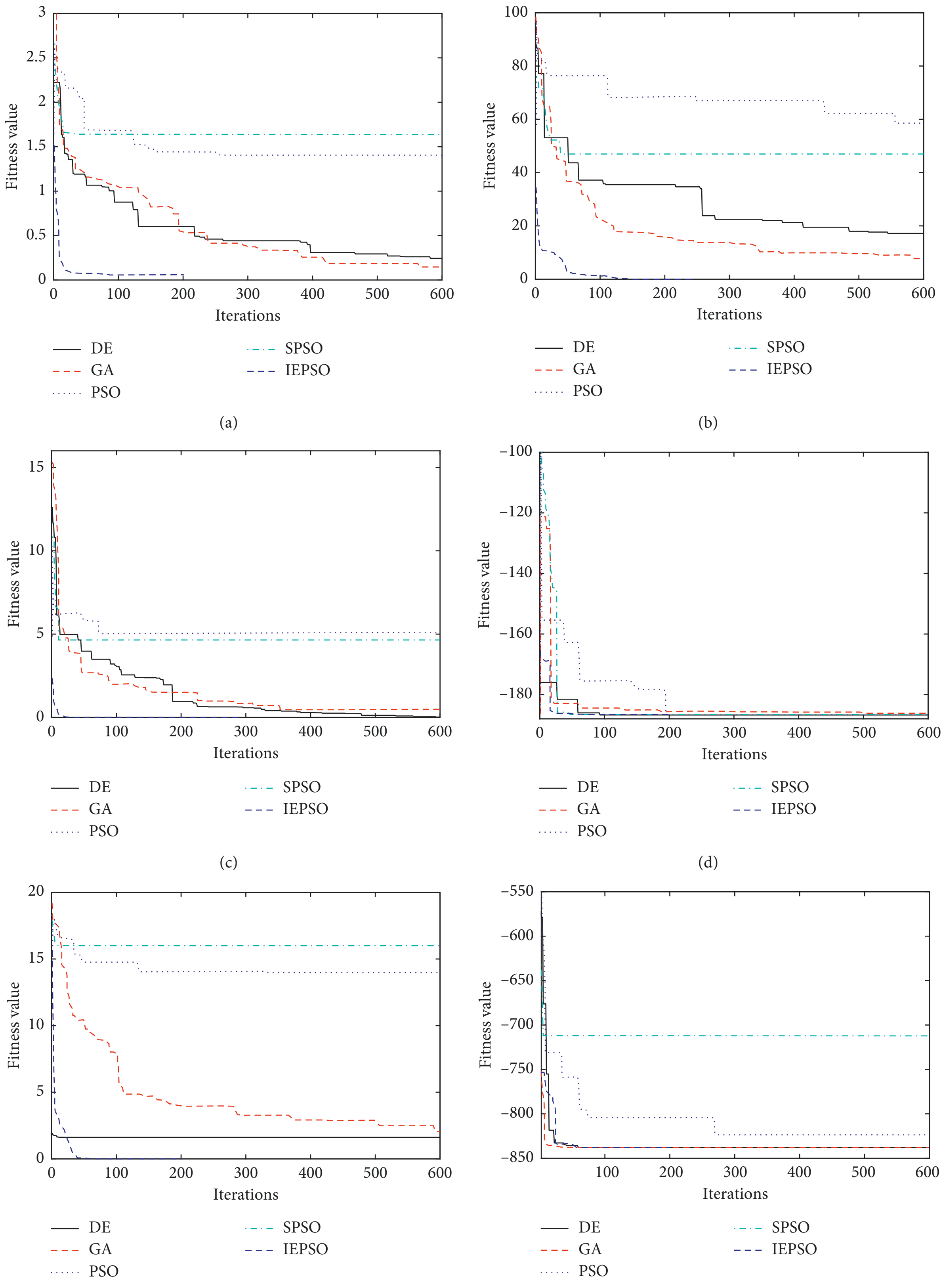
(b)

(c)

(d)

(e)

FIGURE 5: Unimodal functions: (a) f_1 sphere function; (b) f_2 Schaffer function; (c) f_3 step function; (d) f_4 SumSquares function; (e) f_5 Zakharov function.



(a)

(b)

(c)

(d)

(e)

(f)

FIGURE 6: Multimodal functions: (a) f_6 Griewank function; (b) f_7 Rastrigin function; (c) f_8 alpine function; (d) f_9 Shubert function; (e) f_{10} Ackley function; (f) f_{11} Cmfun function.

Table 9: Three improved particle swarm algorithm test results.

Functions	Criteria	IEPSO	DMSDL-PSO [25]	BHPSOWM [26]
f_1	Mean	$\mathbf{8.92E-22}$	$4.73E-10$	42.40
	SD	$\mathbf{2.65E-39}$	$1.81E-09$	52.11
f_3	Mean	$\mathbf{6.21E-19}$	$2.37E+03$	7.61
	SD	$\mathbf{2.63E-36}$	$5.71E+02$	0.07
f_6	Mean	$4.19E-02$	$\mathbf{8.66E-05}$	—
	SD	$3.43E-04$	$\mathbf{2.96E-04}$	—
f_7	Mean	$\mathbf{4.46E-03}$	$9.15E+01$	76.18
	SD	$\mathbf{1.73E-04}$	$1.80E+01$	26.75
f_8	Mean	$\mathbf{2.42E-10}$	$1.31E+02$	—
	SD	$\mathbf{6.74E-20}$	$5.82E+01$	—
f_{10}	Mean	$\mathbf{1.13E-11}$	$1.01E+00$	1.72
	SD	$\mathbf{2.21E-22}$	$2.71E-01$	0

Table 9 shows the test results for the three improved PSO algorithms. The DMSDL-PSO algorithm in [25] is a PSO algorithm combined with differential variation and the quasi-Newton method, whereas the HPSOWM algorithm in [26] is a binary PSO algorithm based on wavelet transform. Table 9 shows that the IEPSO algorithm obtains the best value in 5 out of the 11 test functions, and the above analysis indicates that the IEPSO outperforms the other improved PSO algorithms.

4. Conclusion

In contemporary engineering design, solving the global optimization problems of multiparameter, strongly coupled, and nonlinear systems using conventional optimization algorithms is difficult. In this study, an improved PSO, that is, the IEPSO algorithm, is proposed on the basis of the last-eliminated principle and an enhanced local-global information sharing capability. The comparison and analysis of the simulation results indicate the following conclusions:

(1) The exchange of information between global and local optimal particles enhances the deep search capability of the IEPSO algorithm.

(2) The standard test function is used to simulate the parameter C_3 of the local-global information sharing term. The results show that the global optimization capability of the IEPSO algorithm is strong when C_3 is linearly decreasing. Moreover, the proposed algorithm can show the best search performance when C_3 is a nonlinear convex function.

(3) The last-eliminated principle is used in the IEPSO to maintain particle population diversity. Moreover, PSO is avoided in the local optimal value. A comparison of the IEPSO algorithm with the classical optimization algorithm and its improved versions verifies the global search capability of the IEPSO algorithm.

In summary, the comparative results of the simulation analysis reveal that, with the application of the last-eliminated principle and the local-global information sharing term to the IEPSO, the proposed algorithm effectively overcomes the disadvantages of the classical algorithms, including their precocious convergence and tendency to fall into the local optimum. The IEPSO shows an ideal global optimization performance and indicates a high application value for solving practical engineering optimization problems.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest.

Acknowledgments

This work was supported by Shanghai Rising-Star Program (no. 16QB1401000), Key Project of Shanghai Science and Technology Committee (no. 16DZ1120400), and the National Natural Science Foundation of China (Project no. 51705187), the Postdoctoral Science Foundation of China (Grant no. 2017M621202).

References

[1] 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.

[2] P. Du, R. Barrio, H. Jiang, and L. Cheng, “Accurate Quotient-Difference algorithm: error analysis, improvements and applications,” *Applied Mathematics and Computation*, vol. 309, pp. 245–271, 2017.

[3] L. Jiang, Z. Wang, Y. Ye, and J. Jiang, “Fast circle detection algorithm based on sampling from difference area,” *Optik*, vol. 158, pp. 424–433, 2018.

[4] H. Garg, “A hybrid PSO-GA algorithm for constrained optimization problems,” *Applied Mathematics & Computation*, vol. 274, no. 11, pp. 292–305, 2016.

[5] J. Zhang and P. Xia, “An improved PSO algorithm for parameter identification of nonlinear dynamic hysteretic models,” *Journal of Sound and Vibration*, vol. 389, pp. 153–167, 2017.

[6] R. Saini, P. P. Roy, and D. P. Dogra, “A segmental HMM based trajectory classification using genetic algorithm,” *Expert Systems with Applications*, vol. 93, pp. 169–181, 2018.

[7] P. R. D. O. D. Costa, S. Mauceri, P. Carroll et al., “A genetic algorithm for a vehicle routing problem,” *Electronic Notes in Discrete Mathematics*, vol. 64, pp. 65–74, 2017.

[8] V. Jindal and P. Bedi, “An improved hybrid ant particle optimization (IHAPO) algorithm for reducing travel time in VANETs,” *Applied Soft Computing*, vol. 64, pp. 526–535, 2018.

[9] Z. Peng, H. Manier, and M. A. Manier, “Particle swarm optimization for capacitated location-routing problem,” *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 14668–14673, 2017.

[10] G. Xu and G. Yu, “Reprint of: on convergence analysis of particle swarm optimization algorithm,” *Journal of Shanxi Normal University*, vol. 4, no. 14, pp. 25–32, 2008.

[11] J. Lu, W. Xie, and H. Zhou, “Combined fitness function based particle swarm optimization algorithm for system

identification," *Computers & Industrial Engineering*, vol. 95, pp. 122–134, 2016.

[12] F. Javidrad and M. Nazari, "A new hybrid particle swarm and simulated annealing stochastic optimization method," *Applied Soft Computing*, vol. 60, pp. 634–654, 2017.

[13] J. Jie, J. Zhang, H. Zheng, and B. Hou, "Formalized model and analysis of mixed swarm based cooperative particle swarm optimization," *Neurocomputing*, vol. 174, pp. 542–552, 2016.

[14] A. Meng, Z. Li, H. Yin, S. Chen, and Z. Guo, "Accelerating particle swarm optimization using crisscross search," *Information Sciences*, vol. 329, pp. 52–72, 2016.

[15] L. Wang, B. Yang, and J. Orchard, "Particle swarm optimization using dynamic tournament topology," *Applied Soft Computing*, vol. 48, pp. 584–596, 2016.

[16] M. S. Kiran, "Particle swarm optimization with a new update mechanism," *Applied Soft Computing*, vol. 60, pp. 670–678, 2017.

[17] H. C. Tsai, "Unified particle swarm delivers high efficiency to particle swarm optimization," *Applied Soft Computing*, vol. 55, pp. 371–383, 2017.

[18] S. F. Li and C. Y. Cheng, "Particle swarm optimization with fitness adjustment parameters," *Computers & Industrial Engineering*, vol. 113, pp. 831–841, 2017.

[19] Y. Chen, L. Li, H. Peng, J. Xiao, Y. Yang, and Y. Shi, "Particle swarm optimizer with two differential mutation," *Applied Soft Computing*, vol. 61, pp. 314–330, 2017.

[20] Q. Zhang, W. Liu, X. Meng, B. Yang, and A. V. Vasilakos, "Vector coevolving particle swarm optimization algorithm," *Information Sciences*, vol. 394, pp. 273–298, 2017.

[21] Y. Shi and R. C. Eberhart, "Empirical study of particle swarm optimization[C]//Evolutionary computation," in *Proceedings of the 1999 Congress on Evolutionary Computation-CEC99*, vol. 3, pp. 1945–1950, IEEE, Washington, DC, USA, 1999.

[22] Z. Wang and J. Cai, "The path-planning in radioactive environment of nuclear facilities using an improved particle swarm optimization algorithm," *Nuclear Engineering & Design*, vol. 326, pp. 79–86, 2018.

[23] 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.

[24] K. Chen, F. Zhou, and A. Liu, "Chaotic dynamic weight particle swarm optimization for numerical function optimization," *Knowledge-Based Systems*, vol. 139, pp. 23–40, 2018.

[25] Y. Chen, L. Li, H. Peng, J. Xiao, and Q. Wu, "Dynamic multi-swarm differential learning particle swarm optimizer," *Swarm and Evolutionary Computation*, vol. 39, pp. 209–221, 2018.

[26] F. Jiang, H. Xia, Q. A. Tran, Q. M. Ha, N. Q. Tran, and J. Hu, "A new binary hybrid particle swarm optimization with wavelet mutation," *Knowledge-Based Systems*, vol. 130, pp. 90–101, 2017.

[27] R. Liu, J. Li, C. Mu, J. fan, and L. Jiao, "A coevolutionary technique based on multi-swarm particle swarm optimization for dynamic multi-objective optimization," *European Journal of Operational Research*, vol. 261, no. 3, pp. 1028–1051, 2017.

[28] W. Ye, W. Feng, and S. Fan, "A novel multi-swarm particle swarm optimization with dynamic learning strategy," *Applied Soft Computing*, vol. 61, pp. 832–843, 2017.

[29] L. Zhang, Y. Tang, C. Hua, and X. Guan, "A new particle swarm optimization algorithm with adaptive inertia weight based on Bayesian techniques," *Applied Soft Computing*, vol. 28, pp. 138–149, 2015.

[30] Q. Cui, Q. Li, G. Li et al., "Globally-optimal prediction-based adaptive mutation particle swarm optimization," *Information Sciences*, vol. 418, pp. 186–217, 2017.

[31] D. Zhao and J. Liu, "Study on network security situation awareness based on particle swarm optimization algorithm," *Computers & Industrial Engineering*, vol. 125, pp. 764–775, 2018.

[32] H. Samareh, S. H. Khoshrou, K. Shahriar, M. M. Ebadzadeh, and M. Eslami, "Optimization of a nonlinear model for predicting the ground vibration using the combinational particle swarm optimization-genetic algorithm," *Journal of African Earth Sciences*, vol. 133, pp. 36–45, 2017.

[33] M. Dash, T. Panigrahi, and R. Sharma, "Distributed parameter estimation of IIR system using diffusion particle swarm optimization algorithm," *Journal of King Saud University-Engineering Sciences*, 2017, In press.

[34] B. Wang, S. Li, J. Guo, and Q. Chen, "Car-like mobile robot path planning in rough terrain using multi-objective particle swarm optimization algorithm," *Neurocomputing*, vol. 282, pp. 42–51, 2018.

[35] Z. Wang and J. Cai, "The path-planning in radioactive environment of nuclear facilities using an improved particle swarm optimization algorithm," *Nuclear Engineering & Design*, vol. 326, pp. 79–86, 2018.

[36] R. F. Lopes, F. F. Costa, A. Oliveira et al., "Algorithm based on particle swarm applied to electrical load scheduling in an industrial setting," *Energy*, vol. 147, pp. 1007–1015, 2018.

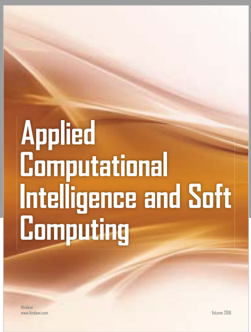
[37] F. Sheikholeslami and N. J. Navimipour, "Service allocation in the cloud environments using multi-objective particle swarm optimization algorithm based on crowding distance," *Swarm & Evolutionary Computation*, vol. 35, pp. 53–64, 2017.

[38] M. Petrović, N. Vuković, M. Mitić et al., "Integration of process planning and scheduling using chaotic particle swarm optimization algorithm," *Expert Systems with Applications*, vol. 64, pp. 569–588, 2016.

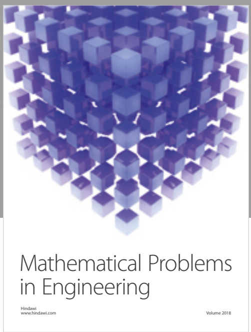
[39] Z. Zhang, Y. Jiang, S. Zhang, S. Geng, H. Wang, and G. Sang, "An adaptive particle swarm optimization algorithm for reservoir operation optimization," *Applied Soft Computing Journal*, vol. 18, no. 4, pp. 167–177, 2014.

[40] K. Li, L. Liu, J. Zhai, T. M. Khoshgoftaar, and T. Li, "The improved grey model based on particle swarm optimization algorithm for time series prediction," *Engineering Applications of Artificial Intelligence*, vol. 55, pp. 285–291, 2016.

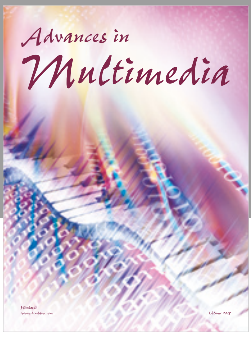
[41] S. Gulcu and H. Kodaz, "The estimation of the electricity energy demand using particle swarm optimization algorithm: a case study of Turkey," *Procedia Computer Science*, vol. 111, pp. 64–70, 2017.

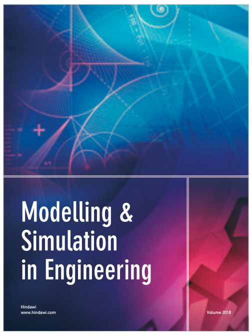

Applied Computational Intelligence and Soft Computing

The Scientific World Journal


Mathematical Problems in Engineering

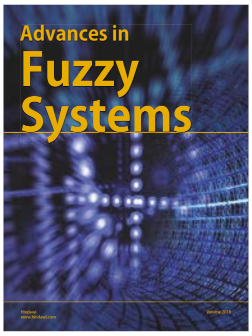
Journal of Engineering

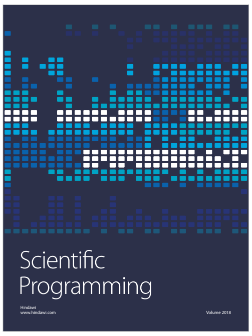

Advances in Multimedia

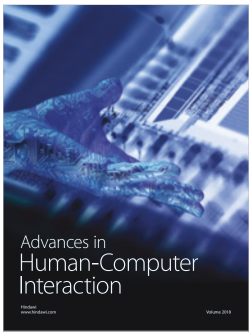

Modelling & Simulation in Engineering

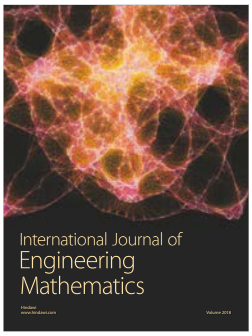
International Journal of Reconfigurable Computing

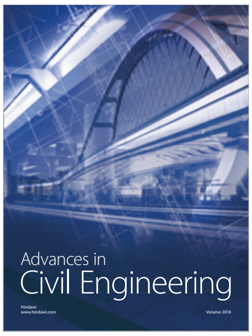
Advances in Artificial Intelligence

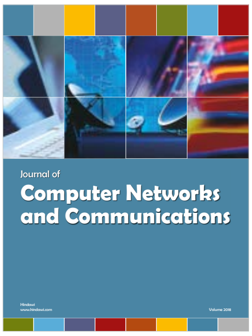

Advances in Fuzzy Systems


Scientific Programming


Advances in Human-Computer Interaction

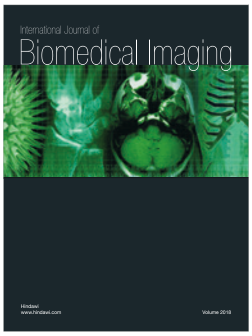

International Journal of Engineering Mathematics


Advances in Civil Engineering

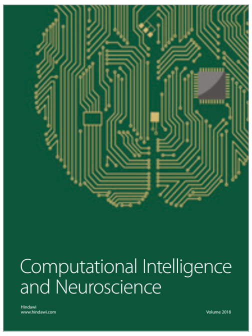

Journal of Computer Networks and Communications

Journal of Robotics

International Journal of Computer Games Technology


International Journal of Biomedical Imaging

Journal of Electrical and Computer Engineering


Computational Intelligence and Neuroscience