Hindawi
Discrete Dynamics in Nature and Society
Volume 2021, Article ID 8378579, 13 pages
https://doi.org/10.1155/2021/8378579

Research Article

Particle Swarm Optimization Algorithm with Multiple Phases for Solving Continuous Optimization Problems

Jing Li,[1] Yifei Sun ,[2] and Sicheng Hou[3]

[1]*Department of Basic Course, Shaanxi Railway Institute, Weinan 714000, China*
[2]*School of Physics & Information Technology, Shaanxi Normal University, Xi'an 710119, China*
[3]*Graduate School of Information, Production and Systems, Waseda University, Kitakyushu 808-0135, Japan*

Correspondence should be addressed to Yifei Sun; yifeis@snnu.edu.cn

Received 16 April 2021; Revised 1 June 2021; Accepted 18 June 2021; Published 28 June 2021

Academic Editor: Lianbo Ma

An algorithm with different parameter settings often performs differently on the same problem. The parameter settings are difficult to determine before the optimization process. The variants of particle swarm optimization (PSO) algorithms are studied as exemplars of swarm intelligence algorithms. Based on the concept of building block thesis, a PSO algorithm with multiple phases was proposed to analyze the relation between search strategies and the solved problems. Two variants of the PSO algorithm, which were termed as the PSO with fixed phase (PSOFP) algorithm and PSO with dynamic phase (PSODP) algorithm, were compared with six variants of the standard PSO algorithm in the experimental study. The benchmark functions for single-objective numerical optimization, which includes 12 functions in 50 and 100 dimensions, are used in the experimental study, respectively. The experimental results have verified the generalization ability of the proposed PSO variants.

1. Introduction

Particle swarm optimization (PSO) algorithm is a population-based stochastic algorithm modeled on the social behaviors observed in flocking birds [1, 2]. As a well-known swarm intelligence algorithm, each particle, which represents a solution in the group, flies through the search space with a velocity that is dynamically adjusted according to its own and its companion's historical behaviors. The particles tend to fly toward better search areas throughout the search process [3].

Many swarm intelligence algorithms have been proposed to solve different kinds of problems, which include single- or multiple-objective optimization and real-world applications. These algorithms include particle swarm optimization algorithm [1, 2], brain storm optimization (BSO) algorithm [4–6], and pigeon-inspired optimization (PIO) [7], just to name a few. Usually, a kind of swarm intelligence algorithm has many variants, which have different search strategies or parameters. Take the PSO algorithm as an example; for single-objective optimization, there are adaptive PSO algorithm [8], time-varying attractor in PSO algorithm [9], interswarm interactive learning strategy in PSO algorithm [10], triple archives PSO algorithm [11], social learning PSO algorithm for scalable optimization [12], PSO variant for mixed-variable optimization problems [13], etc. For multiobjective optimization, there are adaptive gradient multiobjective PSO algorithm [14], coevolutionary PSO algorithm with bottleneck objective learning strategy [15], normalized ranking based PSO algorithm for many-OBJECTIVE optimization [16], etc. Besides, the classical PSO algorithm and its variants have been used in various real-world applications [17], such as search-based data analytics problems [18]. Different variants of optimization algorithms have different components and strategies during the search process. The designing of proper components and strategies is vital for a search algorithm before solving the problem.

The building blocking thesis could be embedded into the genetic algorithm [19]. Building blocks indicate the components from all levels to understand and recognize the complex things or structures. An evolutionary computation algorithm could be divided into several components with the building block thesis. In the PSO algorithm, there are several components. The different settings of parameters, structures, and strategies of the PSO algorithm could perform differently on the same problem. Thus, the proper setting of the optimization algorithm is vital for the solved problem. However, there is no favorable method that could find out the best setting before the optimization. Many approaches have been introduced to analyze the components of the PSO algorithm. For example, the setting of parameter size in the PSO algorithm was surveyed and discussed in [20].

Based on the building block thesis, a PSO algorithm with multiple phases was proposed in this paper. This paper could be useful for understanding the search components in the PSO algorithm and designing the PSO algorithm for specific problems. It has two targets in this paper.

(1) For the theoretical analysis of the PSO algorithm, the effectiveness of different components of various PSO algorithms, such as inertia weight, acceleration coefficient, and topology structures is studied
(2) For the application of the PSO algorithm, more effective PSO algorithms could be designed for solving different real-world applications

The remainder of this paper is organized as follows. The basic PSO algorithm and topology structure were briefly introduced in Section 2. Section 3 gives an introduction to the proposed PSO algorithms with multiple phases. In Section 4, comprehensive experimental studies are conducted on 12 benchmark functions with 50 or 100 dimensions to verify the effectiveness of the proposed algorithms, respectively. Finally, Section 5 concludes with some remarks and future research directions.

2. Backgrounds

PSO algorithm is a widely used swarm intelligence algorithm for optimization. It is easy for understanding and implementation. A potential solution, which is termed as a particle in PSO algorithm, is a search point in the solution space with D-dimensions. For each particle, it is connected with two vectors, *i.e.*, the vector of velocity and the vector of position. Normally, the number of particles is from 1 to S and the index of the particles or solutions is represented by notation i. The number of dimensions is from 1 to D and the index of the dimensions is represented by notation d. The S is the total number of particles and D is the total number of dimensions. The position of the ith particle is represented as $\mathbf{x}_i$, $\mathbf{x}_i = [x_{i1}, x_{i2}, \ldots, x_{id}, \ldots, x_{iD}]$. x_{id} is the value of the dth dimension for the ith solution, where $i = 1, 2, \ldots, S$, and $d = 1, 2, \ldots, D$. The velocity of a particle is labeled as $\mathbf{v}_i$, $\mathbf{v}_i = [v_{i1}, v_{i2}, \ldots, v_{id}, \ldots, v_{iD}]$.

2.1. Particle Swarm Optimization Algorithm. The basic process of PSO algorithm is given in Algorithm 1. There are two equations, the update of velocity and position for each particle, in the basic process of PSO algorithm. The framework of canonical PSO algorithm is shown in Figure 1. The fitness value is evaluated based on the calculation on all dimensions, while the update on the velocity or position is based on each dimension. The update equations for the velocity v_{id} and the position x_{id} are as follows [17, 21]:

$$v_{id}^{t+1} = w_i v_{id}^{t} + c_1 \mathrm{rand}\left(p_{id}^{t} - x_{id}^{t}\right) + c_2 \mathrm{rand}\left(p_{nd}^{t} - x_{id}^{t}\right), \tag{1}$$

$$x_{id}^{t+1} = x_{id}^{t} + v_{id}^{t+1}, \tag{2}$$

where w denotes the inertia weight, c_1 and c_2 are two positive acceleration constants, and rand is a random function to generate uniformly distributed random numbers in the range $[0, 1)$. $\mathbf{p}_i^t = [p_{i1}^t, p_{i2}^t, \ldots, p_{id}^t, \ldots, p_{im}^t]$ is termed as personal best at the tth iterations, which refers to the best position found so far by the ith particle, and $\mathbf{p}_n^t = [p_{n1}^t, p_{n2}^t, \ldots, p_{nd}^t, \ldots, p_{nm}^t]$ is termed as local best, which refers to the position found so far by the members in the ith particle's neighborhood that has the best fitness value. The parameter w was introduced to control the global search and local search ability. The PSO algorithm with the inertia weight is termed as the canonical/classical PSO.

The position value of a particle was updated in the search space at each iteration. The velocity update equation is shown in equation (1). This equation has three components: the previous velocity, cognitive part, and social part. The cognitive part indicates that a particle learns from its own searching experience, and correspondingly, the social part indicates that a particle can learn from other particles or learn from the good solutions in the neighborhood.

2.2. Topology Structure. Different kinds of topology structures, such as a global star, local ring, four clusters, or Von Neumann structure, could be utilized in PSO variants to solve various problems. A particle in a PSO with a different structure has a different number of particles in its neighborhood with different scope. Learning from a different neighbor means that a particle follows a different neighborhood (or local) best; in other words, the topology structure determines the connections among particles and the strategy used in the propagation process of searching information over iteration.

A PSO algorithm's ability of exploration and exploitation could be affected by its topology structure; i.e., with a different structure, the algorithm's convergence speed and the ability to avoid premature convergence will be various on the same optimization problem, because a topology structure determines the search information sharing speed or direction for each particle. The global star and local ring are the two typical topology structures. A PSO with a global star structure, where all particles are connected, has the smallest average distance in the swarm, and on the contrary, a PSO with a local ring structure, where every particle is connected to two near particles, has the largest average distance in swarm [22].

The global star structure and the local ring structure, which are two commonly used structures, are analyzed in the experimental study. These two structures are shown in Figure 2. Each group of particles has 16 individuals. It should

Initialize each particle's velocity and position with random numbers
while not reaches the maximum iteration or not found the satisfied solution **do**
  Calculate each solution's function value;
  Compare function value between the current position and the best position in history. For each particle, if current position has a better function value than pbest, then update pbest as current position;
  Select a particle that has the best fitness value from the current particle's neighborhood, this particle is termed as the neighborhood best (nbest);
  for each particle **do**
    Update particle's velocity according equation (1);
    Update particle's position according equation (2);

ALGORITHM 1: Basic process of PSO algorithm.

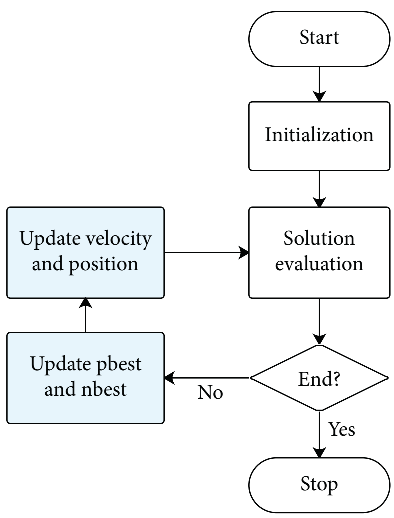


FIGURE 1: The framework of canonical PSO algorithm.

be noted that the nearby particle in the local structure is mostly based on the index of particles.

3. Particle Swarm Optimization with Multiple Phases

The search process could be divided into different phases, such as the exploration phase and the exploitation phase. A discussion has been given on the genetic diversity of an evolutionary algorithm in the exploration phase [23]. To enhance the generalization ability of the PSO algorithm, multiple phases could be combined into one algorithm. The PSO algorithm is a combination of several components. Each component could be seen as a building block of a PSO variant. Some changeable components in PSO algorithms are shown in Figure 3. The setting of a component is independent of other components. Normally, the parameters' settings, topology structure, and other strategies should be determined before the search process. There are three well-known standard PSO algorithms, namely, the standard PSO algorithm by Bratton and Kennedy (SPSO-BK) [24], the standard PSO algorithm by Clerc (SPSO-C) [25], and the canonical PSO (CPSO) algorithm By Shi and Eberhart [26].

Different search phases are emphasized for PSO variants because it performs differently when solving different kinds of problems. Thus, to utilize the strengths of different variants, a PSO algorithm could be a combination of several PSO variants. For example, the global star structure could be used at the beginning of the search to obtain a good exploration ability, while the local ring structure could be used at the ending of the search to promote the exploitation ability. The basic procedure of the PSO algorithm with multiple phase strategy (PSOMP) is given in Algorithm 2. The setting of phases could be fixed or dynamically changed during the search. To validate the performance of different settings, two kinds of PSO algorithms with multiple phases are proposed.

3.1. Particle Swarm Optimization with Fixed Phases. One PSOMP variant is the PSO with fixed phases (PSOFP) algorithm. The process of the PSO algorithm with multiple fixed phases is shown in Figure 4. Two standard PSO variants, the CPSO with star structure and the SPSO-BK with ring structure, are combined for the PSOFP algorithm. In the experimental study, both variants perform the same iterations. The PSOFP algorithm started with the CPSO with the star structure and ended with the SPSO-BK with the ring structure. All parameters are fixed for each phase during the search. The PSOFP algorithm switches from one variant to another after the fixed number of iterations.

3.2. Particle Swarm Optimization with Dynamic Phases. The other PSO variant is PSO with dynamic phases (PSODP) algorithm. As the name indicates, the phase could be dynamically changed for this algorithm during the search process. As the same with the PSOFP algorithm, the PSODP algorithm started with the CPSO with the star structure and ended with the SPSO-BK with the ring structure. However, the number of iterations is not fixed during the search. The process of the PSO algorithm with multiple dynamic phases is shown in Figure 5. The global best (gbest) position is the best solution found so far. The gbest not changing for k iterations indicates that the algorithm may be stuck into local optima. The PSO variant will be changed after this phenomenon occurs several times. For the setting of the PSODP algorithm, the search phase will change from CPSO with star structure to SPSO-BK with a ring structure. The condition of phase change is the gbest stuck into a local optimum; i.e., the gbest is not changed for 100 iterations.

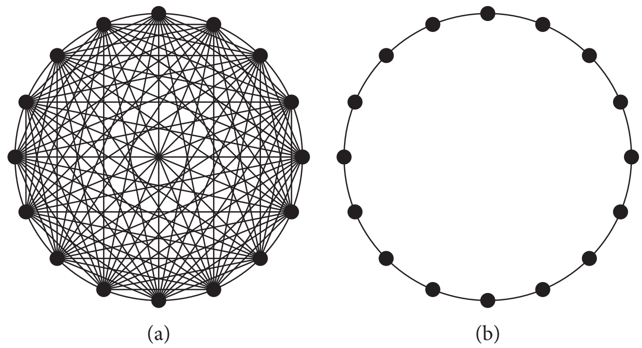


FIGURE 2: Two structures have been used in this paper: (a) global star structure, where all particles are connected; (b) local ring structure, where each particle is connected to two nearby neighbors.

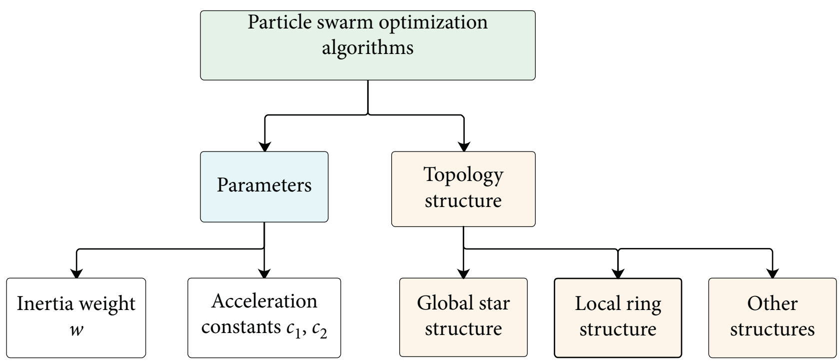


FIGURE 3: Some changeable components in PSO algorithms.

(1) Initialize each particle's velocity and position with random numbers;
(2) **while** not reaches the maximum iteration or not found the satisfied solution **do**
(3) Calculate each solution's function value;
(4) Compare function value between the current position and the best position in history (personal best, termed as pbest). For each particle, if current position has a better function value than pbest, then update pbest as current position;
(5) Selection a particle which has the best fitness value among current particle's neighborhood, this particle is termed as the neighborhood best;
(6) **for** each particle **do**
(7) Update the particle's velocity according equation (1);
(8) Update the particle's position according equation (2);
(9) **Change the search phase**: update the structure and/or parameter settings;

ALGORITHM 2: Basic procedure of PSO algorithm with multiple phase strategy.

4. Experimental Study

4.1. Benchmark Test Functions and Parameter Setting. Table 1 shows the benchmark functions which have been used in the experimental study. To give an illustration of the search ability of the proposed algorithm, a diverse set of 12 benchmark functions with different types are used to conduct the experiments. These benchmark functions can be classified into two groups. The first five functions $f_0 - f_4$ are unimodal, while the other seven functions are multimodal. All functions are run 50 times to ensure a reasonable statistical result necessary to compare the different approaches, and the value of global optimum is shifted to different $f_{\min}$ for different functions. The function value of the best solution found by an algorithm in a run is denoted by $f(\vec{x}_{\text{best}})$. The error of each run is denoted as $\text{error} = f(\vec{x}_{\text{best}}) - f_{\min}$.

Three kinds of standard or canonical PSO algorithms are tested in the experimental study. Each algorithm has two kinds of structure, the global star structure and the local ring structure. There are 50 particles in each group of PSO algorithms. Each algorithm runs 50 times, 10000 iterations in every run. The other settings for the three standard algorithms are as follows.

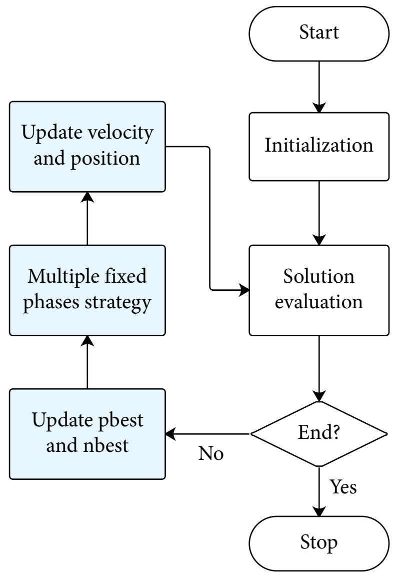


Figure 4: The process of PSO algorithm with multiple fixed phases.

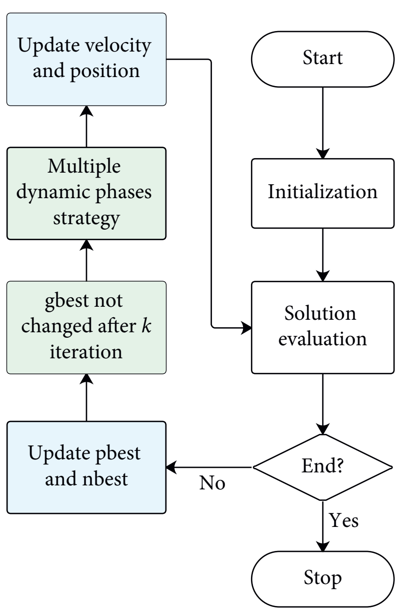


Figure 5: The process of PSO algorithm with multiple dynamic phases.

(i) SPSO-BK settings: $w = 0.72984$; $\varphi_1 = \varphi_2 = 2.05$, *i.e.*, $c_1 = c_2 = 1.496172$ [24]

(ii) SPSO-C settings: $w = (1/2\ \ln(2)) \simeq 0.721$; $c_1 = c_2 = 0.5 + \ln(2) \simeq 1.193$ [25]

(iii) Canonical PSO (CPSO) settings: $w = w_{\min} + (w_{\max} - w_{\min}) \times$ (Iteration− iter/Iteration); *i.e.*, w is linearly decreased during the search; $w_{\max} = 0.9$, $w_{\min} = 0.4$; $c_1 = c_2 = 2$ [26]

4.2. Experimental Results. The experimental results for functions with 50 dimensions are given in Tables 2 and 3. The results for unimodal functions $f_0 - f_4$ are given in Table 2, while results for multimodal functions $f_5 - f_{11}$ are given in Tables 2 and 3. The aim of the experimental study is not to validate the search performance of the proposed algorithm but to attempt to discuss the combination of different search phases. This is a primary study on the learning of the characteristics of benchmark functions.

To validate the generalization ability of PSO variants, the same experiments are conducted on the functions with 100 dimensions. All parameters' settings are the same for functions with 50 dimensions. The results for unimodal functions $f_0 - f_4$ are given in Table 4, while the results for multimodal functions $f_5 - f_{11}$ are given in Tables 4 and 5. From the experiment results, it could be seen that the proposed algorithm could obtain a good solution for most benchmark functions. Based on the combination of different phases, the proposed algorithm could be performed more robustly than the algorithm with one phase.

The convergence graphs of error values for each algorithm on all functions with 50 and 100 dimensions are shown in Figures 6 and 7, respectively. In these figures, each curve represents the variation of the mean value of error over the iteration for each algorithm. The six PSO variants are tested with the proposed two algorithms. The notation of "*S*" and "*R*" indicates the star and ring structure in the PSO variants, respectively. For example, "SPSOBKS" and "SPSOBKR" indicate the SPSO-BK algorithm with the star and ring structure, respectively. The robustness of the proposed algorithm also could be validated from the convergence graph.

4.3. Computational Time. Tables 6 and 7 give the total computing time (seconds) of 50 runs in the experiments for functions with 50 and 100 dimensions, respectively. Normally, the evaluation speed of the PSO algorithm with ring structure is significantly slower than the same PSO algorithm with a star structure.

Table 1: The benchmark functions used in our experimental study, where n is the dimension of each problem, global optimum is $\mathbf{x}^*$, $f_{\min}$ is the minimum value of the function, and $S \subseteq \mathbb{R}^n$.

Function	Test function	n	S	$f_{\min}$
Parabolic	$f_0(\mathbf{x}) = \sum_{i=1}^{n} x_i^2$	100	$[-100, 100]^n$	−450.0
Schwefel's P2.22	$f_1(\mathbf{x}) = \sum_{i=1}^{n} \lvert x_i \rvert + \prod_{i=1}^{n} \lvert x_i \rvert$	100	$[-10, 10]^n$	−330.0
Schwefel's P1.2	$f_2(\mathbf{x}) = \sum_{i=1}^{n} (\sum_{k=1}^{i} x_k)^2$	100	$[-100, 100]^n$	450.0
Step quartic noise	$f_3(\mathbf{x}) = \sum_{i=1}^{n} (\lfloor x_i + 0.5 \rfloor)^2$	100	$[-100, 100]^n$	180.0
	$f_4(\mathbf{x}) = \sum_{i=1}^{n} i x_i^4 + \text{random}[0, 1)$	100	$[-1.28, 1.28]^n$	120.0
Rosenbrock	$f_5(\mathbf{x}) = \sum_{i=1}^{n-1} [100 (x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	100	$[-10, 10]^n$	−450.0
Schwefel	$f_6(\mathbf{x}) = \sum_{i=1}^{n} -x_i \sin(\sqrt{\lvert x_i \rvert}) + 418.9829n$	100	$[-500, 500]^n$	−330.0
Rastrigin	$f_7(\mathbf{x}) = \sum_{i=1}^{n} [x_i^2 - 10 \cos(2\pi x_i) + 10]$	100	$[-5.12, 5.12]^n$	450.0
Noncontinuous Rastrigin	$f_8(\mathbf{x}) = \sum_{i=1}^{n} [y_i^2 - 10 \cos(2\pi y_i) + 10]$, $y_i = \begin{cases} x_i & \lvert x_i \rvert < (1/2) \\ (\text{round}(2x_i)/2) & \lvert x_i \rvert \geq (1/2) \end{cases}$	100	$[-5.12, 5.12]^n$	180.0
Ackley	$f_9(\mathbf{x}) = -20 \exp(-0.2\sqrt{(1/n) \sum_{i=1}^{n} x_i^2}) - \exp((1/n) \sum_{i=1}^{n} \cos(2\pi x_i)) + 20 + e$	100	$[-32, 32]^n$	120.0
Griewank	$f_{10}(\mathbf{x}) = (1/4000) \sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n} \cos(x_i/\sqrt{i}) + 1$	100	$[-600, 600]^n$	−450.0
Generalized penalized	$f_{11}(\mathbf{x}) = (\pi/n)\left\{10 \sin^2(\pi y_1) + \sum_{i=1}^{n-1} (y_i - 1)^2 \times [1 + 10 \sin^2(\pi y_{i+1})] + (y_n - 1)^2\right\} + \sum_{i=1}^{n} u(x_i, 10, 100, 4)$, $y_i = 1 + (1/4)(x_i + 1)$, $u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m & x_i > a \\ 0 & -a < x_i < a \\ k(-x_i - a)^m & x_i < -a \end{cases}$	100	$[-50, 50]^n$	−330.0

Table 2: Experimental results for functions $f_0 \sim f_5$ with 50 dimensions.

Algorithm		Best	Mean	St.D.	Best	Mean	St.D.
			f_0 parabolic			f_1 Schwefel's P2.22	
SPSO-BK	Star	**−450**	**−450**	4.52E-13	**−330**	**−329.9999**	1.35E-11
	Ring	−449.9999	−449.9999	2.11E-12	−329.9999	**−329.9999**	4.37E-12
SPSO-C	Star	−449.9999	−449.9999	4.12E-09	−329.9999	**−329.9999**	3.08E-05
	Ring	−449.9999	−449.9999	3.13E-12	−329.9999	**−329.9999**	6.01E-12
CPSO	Star	**−450**	**−450**	5.90E-14	**−330**	−329.8	1.4
	Ring	−373.3774	−191.0975	98.981	−329.8322	−325.9999	2.151
PSOFP		**−450**	**−450**	1.47E-12	−329.9999	**−329.9999**	6.25E-09
PSODP		**−450**	**−450**	1.91E-12	−329.9999	**−329.9999**	1.45E-06
			f_2 Schwefel's P1.2			f_3 step	
SPSO-BK	Star	**450.0000**	550.0000	699.9999	**180**	188.36	13.936
	Ring	455.3668	495.6626	38.833	266	360.46	50.708
SPSO-C	Star	**450.0000**	**450.3961**	2.748	207	343.66	106.891
	Ring	450.0958	451.3167	2.5006	373	714.3	167.178
CPSO	Star	476.3968	602.0931	697.433	**180**	**180**	0
	Ring	3121.6019	4966.1359	769.950	395	534.64	70.468
PSOFP		450.0476	454.3896	29.0223	**180**	180.4	2.107
PSODP		450.0013	460.0571	34.158	**180**	187.54	26.872
			f_4 quartic noise			f_5 Rosenbrock	
SPSO-BK	Star	**120.0022**	**120.0044**	0.001	−449.9976	−418.4438	24.245
	Ring	120.0281	120.0630	0.023	−444.3137	−415.0037	9.573
SPSO-C	Star	120.0408	120.1508	0.070	**−449.9999**	**−448.3351**	2.378
	Ring	120.0467	120.2094	0.085	−443.4564	−411.7220	17.957
CPSO	Star	120.0037	120.0090	0.002	−421.5111	−389.7360	27.273
	Ring	120.0194	120.0321	0.006	−218.8776	193.7953	156.862
PSOFP		120.0026	120.0061	0.0024	−439.0752	−395.1315	31.0671
PSODP		120.0029	120.0086	0.0076	−444.2963	−405.6357	25.7581

Table 3: Experimental results for functions $f_6 \sim f_{11}$ with 50 dimensions.

Algorithm		Best	Mean	St.D.	Best	Mean	St.D.
			f_6 Schwefel			f_7 Rastrigin	
SPSO-BK	Star	6603.8424	9786.6971	1359.137	497.7579	523.1492	13.619
	Ring	16100.4628	17437.3017	673.960	497.7580	530.6957	14.832
SPSO-C	Star	7684.6670	9898.3600	1218.2453	**486.8134**	**520.5225**	15.146
	Ring	16299.0367	17523.4040	548.640	499.7950	530.1292	14.646
CPSO	Star	6934.3602	8761.8199	854.741	495.7680	529.6818	22.117
	Ring	15875.0546	17592.1651	661.051	578.8806	631.1157	16.836
PSOFP		6144.7441	8472.2288	1171.500	504.7226	528.2633	13.507
PSODP		**5950.1707**	**8369.7773**	1038.811	493.7781	525.2104	20.126
			f_8 noncontinuous Rastrigin			f_9 Ackley	
SPSO-BK	Star	**186**	**224.58**	22.039	**120**	121.773	0.649
	Ring	232.0000	262.7717	14.463	120.0000	120.0213	0.149
SPSO-C	Star	227	269.985	21.175	121.8030	124.3168	1.090
	Ring	239.0000	277.1382	18.542	120.0000	120.3807	0.558
CPSO	Star	211	240.6001	14.448	**120**	**120**	7.15E-14
	Ring	279.2935	321.5721	17.248	122.7702	124.2613	0.529
PSOFP		210	237.3073	19.334	**120**	120.0499	0.248
PSODP		200	241.0076	20.067	**120**	120.0484	0.237
			f_{10} Griewank			f_{11} generalized penalized	
SPSO-BK	Star	**−450**	−449.9769	0.033	**−330**	−329.7654	0.330
	Ring	−449.9999	**−449.9997**	0.002	−329.9999	−329.9987	0.008
SPSO-C	Star	−449.9999	−449.8900	0.228	−329.9999	−329.0989	1.123
	Ring	−449.9999	−449.9989	0.004	−329.9999	−329.9912	0.039
CPSO	Star	**−450**	−449.9902	0.012	**−330**	**−329.9975**	0.012
	Ring	−448.2587	−445.2495	1.087	−329.2803	−327.6535	0.783
PSOFP		**−450**	−449.9874	0.019	**−330**	−329.9763	0.051
PSODP		**−450**	−449.9902	0.017	**−330**	−329.9763	0.055

Table 4: Experimental results for functions $f_0 \sim f_5$ with 100 dimensions.

Algorithm		Best	Mean	St.D.	Best	Mean	St.D.
			f_0 parabolic			f_1 Schwefel's P2.22	
SPSO-BK	Star	**−450**	**−449.9999**	2.97E-10	**−329.9999**	**−329.9999**	1.90E-07
	Ring	−449.9999	**−449.9999**	2.72E-11	**−329.9999**	**−329.9999**	7.34E-11
SPSO-C	Star	−449.9999	−449.9947	0.0369	**−329.9999**	−329.8556	0.4552
	Ring	−449.9999	**−449.9999**	1.40E-11	**−329.9999**	**−329.9999**	3.09E-11
CPSO	Star	−449.9999	**−449.9999**	2.13E-09	**−329.9999**	−329.5999	1.9595
	Ring	−149.6541	1344.4583	532.677	−325.8224	−309.5554	7.5546
PSOFP		−449.9999	**−449.9999**	1.36E-07	**−329.9999**	**−329.9999**	1.36E-09
PSODP		−449.9999	**−449.9999**	3.34E-07	**−329.9999**	**−329.9999**	0.0006
			f_2 Schwefel's P1.2			f_3 step	
SPSO-BK	Star	**513.6355**	**1602.1760**	3006.812	208	377.32	137.165
	Ring	2426.1604	5355.3297	1565.104	671	978.24	174.267
SPSO-C	Star	555.4292	2998.8775	3037.685	1208	2683.48	755.401
	Ring	1188.1057	2075.1955	769.974	1585	2448.46	531.394
CPSO	Star	5145.9671	10822.5242	3423.207	**180**	**181.7**	1.3
	Ring	20782.011	27917.191	3168.315	1791	2506.12	287.989
PSOFP		1399.1752	3987.6253	5750.961	182	199.76	28.642
PSODP		989.7956	2861.3122	5655.407	185	215.36	115.588
			f_4 quartic noise			f_5 Rosenbrock	
SPSO-BK	Star	**120.0138**	**120.0278**	0.0110	−397.0471	−329.9329	41.398
	Ring	120.1851	120.3265	0.1013	−373.0928	−346.4626	25.371
SPSO-C	Star	120.5708	121.3554	0.4501	**−397.2747**	−304.3235	40.5292
	Ring	120.4383	120.9911	0.3064	−371.9901	**−339.0829**	28.481
CPSO	Star	120.0379	120.0658	0.014	−363.4721	−294.8193	50.003
	Ring	120.1338	120.2321	0.043	606.9444	3907.8062	1468.905
PSOFP		120.0224	120.0438	0.014	−380.2421	−290.0060	47.549
PSODP		120.0219	120.0518	0.036	−368.7163	−299.5108	43.518

Table 5: Experimental results for functions $f_6 \sim f_{11}$ with 100 dimensions.

Algorithm		Best	Mean	St.D.	Best	Mean	St.D.
			f_6 Schwefel			f_7 Rastrigin	
SPSO-BK	Star	16056.678	20638.485	2182.500	569.3949	611.7217	25.514
	Ring	34073.736	37171.637	944.548	560.5540	627.2546	25.699
SPSO-C	Star	16879.295	20909.081	2451.469	**552.4806**	**598.6068**	23.697
	Ring	34864.588	37164.686	841.829	569.4197	624.0863	22.440
CPSO	Star	15445.328	19374.588	1691.740	585.3142	639.1086	30.012
	Ring	34235.521	37141.853	835.129	943.5140	1021.6837	37.831
PSOFP		14734.074	**18723.674**	1738.027	583.3243	648.9117	26.8680
PSODP		**14094.634**	18899.269	1994.712	574.8307	639.9075	28.179
			f_8 noncontinuous Rastrigin			f_9 Ackley	
SPSO-BK	Star	**212.0000**	**275.1500**	30.6297	122.5793	123.5817	0.7107
	Ring	307.2505	358.3455	27.633	**120.0000**	121.8010	0.4814
SPSO-C	Star	288.0000	374.4500	41.1362	126.3445	128.3848	1.1134
	Ring	332.2500	395.7141	34.1028	**120.0000**	121.8958	0.5702
CPSO	Star	322.0000	375.1452	31.873	**120.0000**	**120.0000**	0.0001
	Ring	580.5314	660.6651	34.490	123.0676	126.4343	0.7101
PSOFP		357	360.7510	38.267	120.0000	121.6460	0.485
PSODP		280	355.1481	29.234	120.0000	121.6061	0.562
			f_{10} Griewank			f_{11} generalized penalized	
SPSO-BK	Star	**−450**	−449.8481	0.2569	**−330**	−329.6599	0.3758
	Ring	−449.9999	−449.9994	0.0022	−329.9999	**−329.9880**	0.0334
SPSO-C	Star	−449.9999	−449.4670	0.6696	−329.9999	−329.2211	0.8798
	Ring	−449.9999	**−449.9995**	0.0024	−329.9999	−329.9526	0.0640
CPSO	Star	−449.9999	−449.9933	0.0092	−329.9999	−329.9707	0.0432
	Ring	−437.0433	−428.5081	2.0322	−320.4856	−312.0457	4.2756
PSOFP		−449.9999	−449.9817	0.032	−329.9999	−329.9069	0.151
PSODP		−449.9999	−449.9828	0.031	−329.9999	−329.9224	0.108

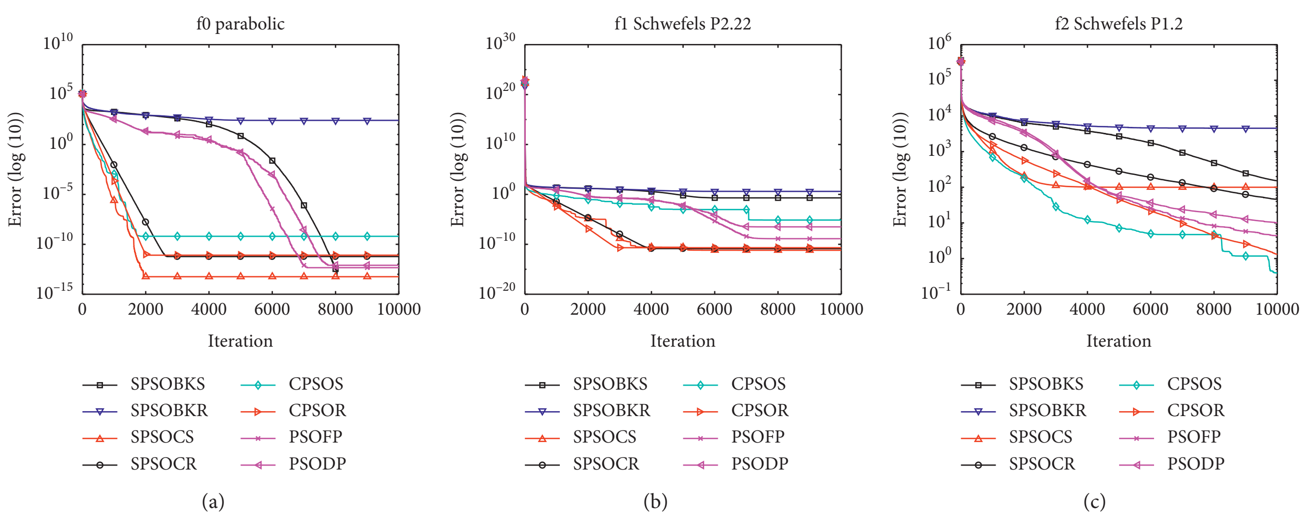


Figure 6: Continued.

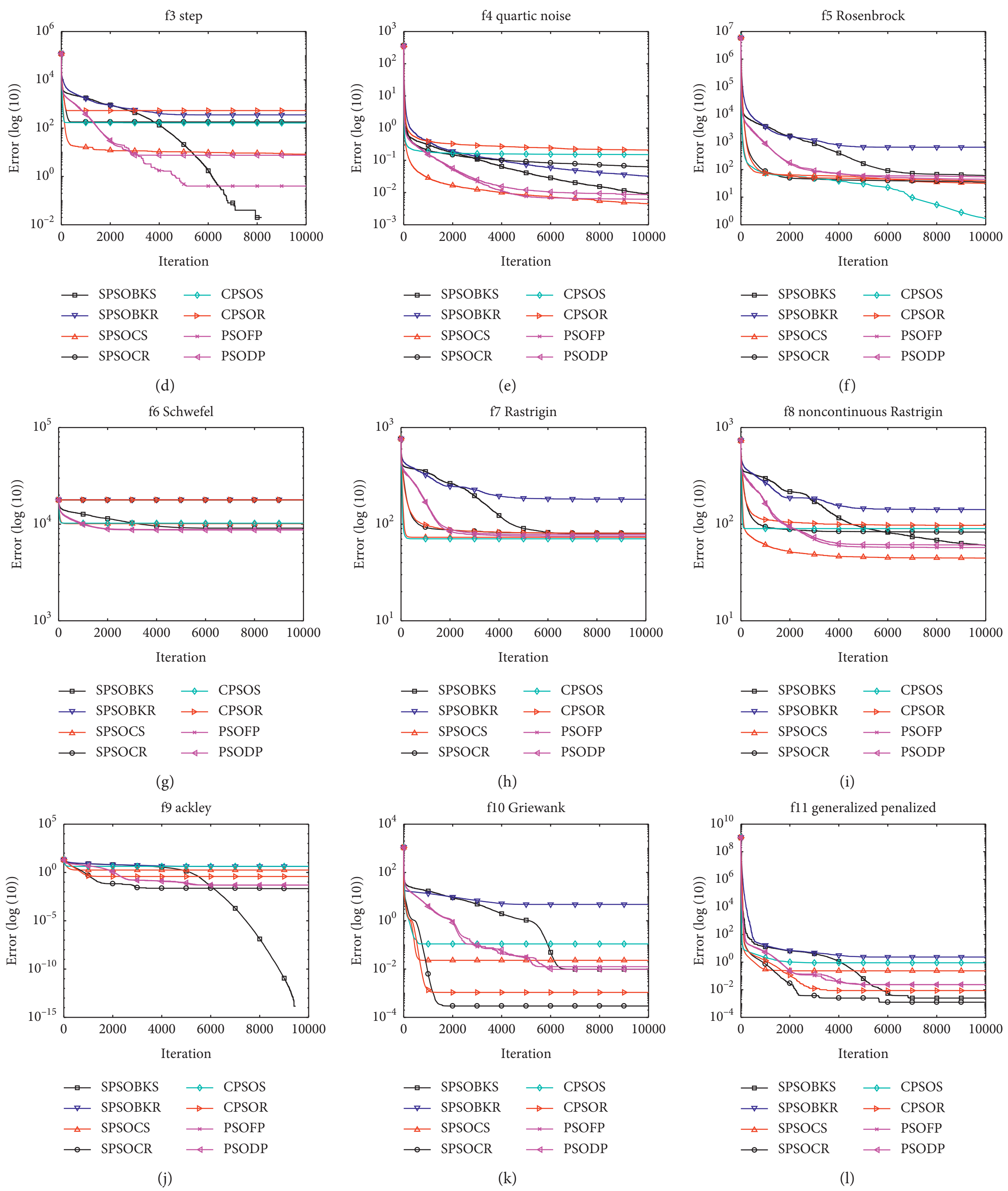


FIGURE 6: Convergence curves of error value for functions with 50 dimensions. (a) f_0. (b) f_1. (c) f_2. (d) f_3. (e) f_4. (f) f_5. (g) f_6. (h) f_7. (i) f_8. (j) f_9. (k) f_{10}. (l) f_{11}.

The PSOFP and PSODP algorithms perform slower than the traditional PSO algorithm on the test benchmark functions. The reason is that some search resources are allocated on the changes of search phases. The variant of PSO with multiple phases needs to determine whether to switch phase and the time of switching phase.

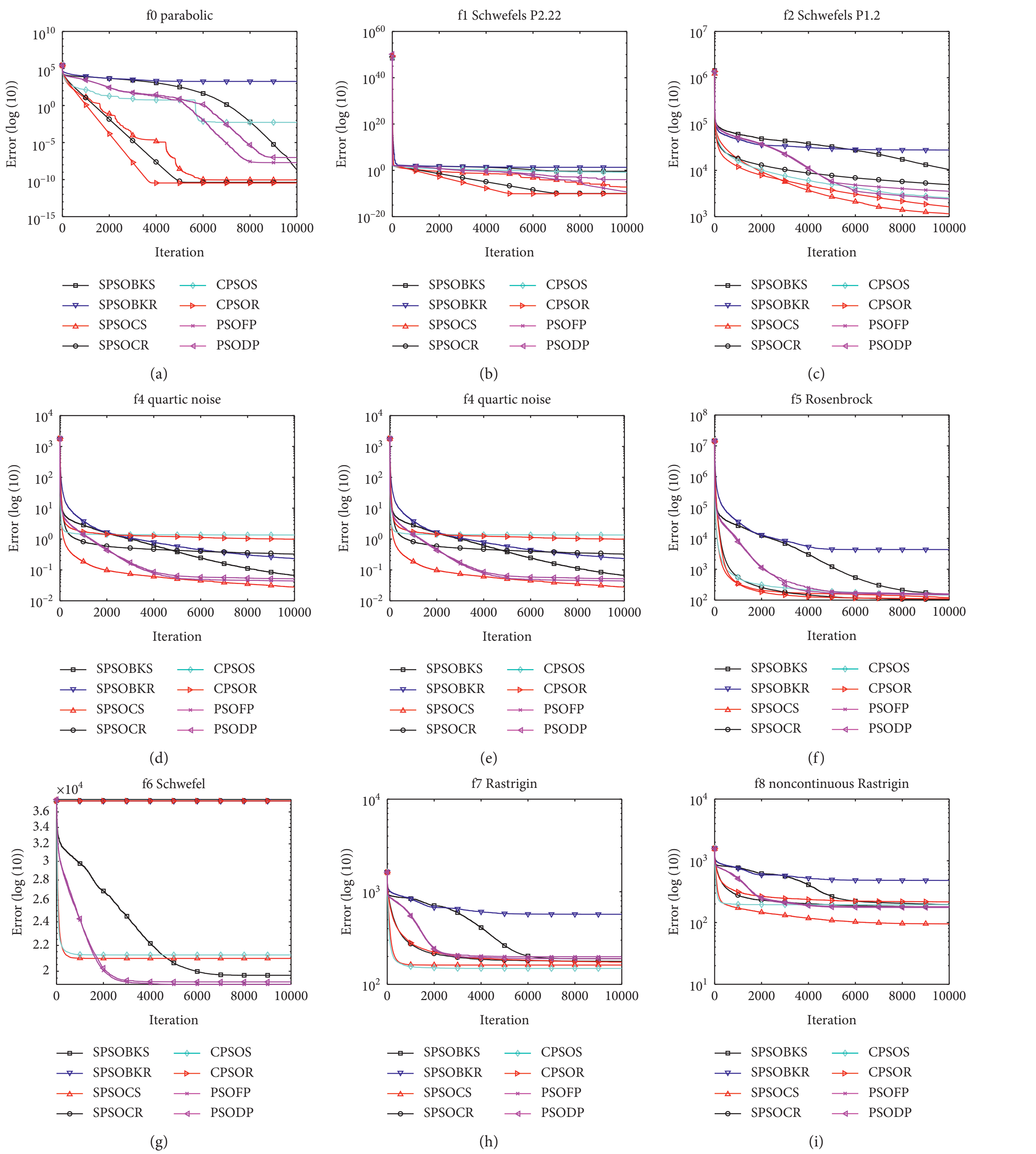
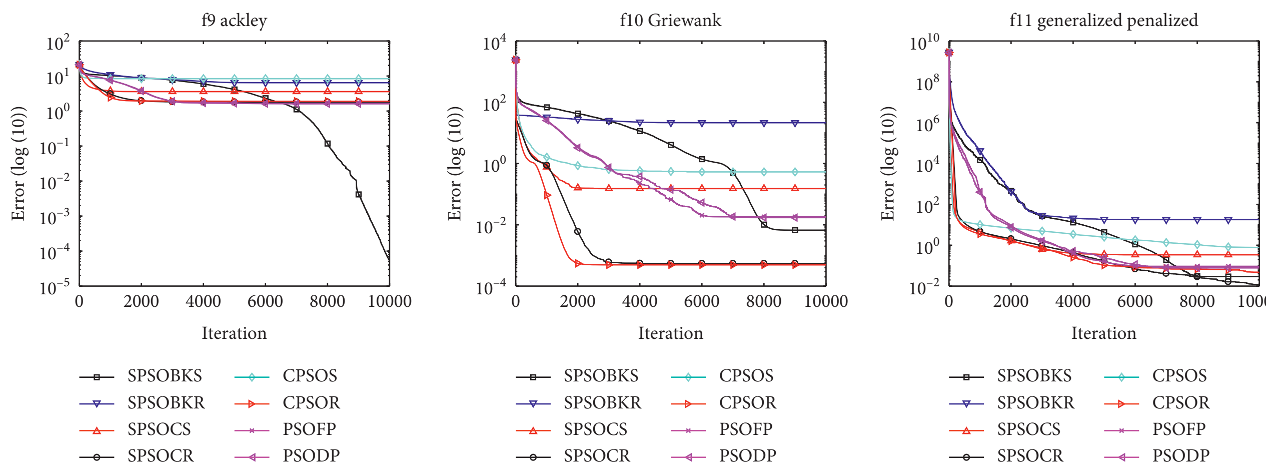


Figure 7: Continued.



(j) (k) (l)

FIGURE 7: Convergence curves of error value for functions with 100 dimensions. (a) f_0. (b) f_1. (c) f_2. (d) f_3. (e) f_4. (f). f_5. (g) f_6. (h) f_7. (i) f_8. (j) f_9. (k) f_{10}. (l) f_{11}.

TABLE 6: Computational time (seconds) spent in 50 runs for the function with 50 dimensions.

Function	SPSO-BK		SPSO-C		CPSO		PSOFP	PSODP
	Star	Ring	Star	Ring	Star	Ring		
f_0	141.4753	1194.9538	154.2324	1393.8880	**127.3165**	779.0309	637.2494	475.0059
f_1	149.6900	977.3025	156.3205	1203.5832	**138.2238**	733.2784	668.1761	496.8441
f_2	248.2647	663.9135	231.0108	326.9424	**226.9095**	772.1789	291.3018	307.8365
f_3	147.8375	1692.1071	152.1913	1737.4540	**138.2383**	716.2581	665.2441	544.4709
f_4	192.5499	777.2614	**191.9068**	852.0861	196.0920	413.9658	432.6319	440.5604
f_5	**131.8303**	203.6976	137.3674	146.3761	137.8663	766.9009	252.0464	197.5675
f_6	247.0545	599.0113	262.3676	603.3503	**229.0871**	487.7298	225.6345	241.2337
f_7	211.3601	568.8381	212.1822	506.8316	**188.6031**	761.6392	746.7872	593.8517
f_8	**210.0288**	1188.4555	223.1706	1180.9223	213.6926	765.8513	608.0168	519.1364
f_9	213.6390	837.5839	203.2908	1179.4232	**182.0255**	755.5640	764.1055	542.1310
f_{10}	206.5978	1244.9973	211.6536	1409.6724	**194.0266**	829.1956	721.1344	550.2234
f_{11}	341.6110	1230.0763	340.1308	1295.6684	**327.5602**	919.8441	864.0743	697.8836

TABLE 7: Computational time (seconds) spent in 50 runs for the function with 100 dimensions.

Function	SPSO-BK		SPSO-C		CPSO		PSOFP	PSODP
	Star	Ring	Star	Ring	Star	Ring		
f_0	243.9461	1378.8133	238.7528	1936.8303	**228.8345**	1461.8672	629.0300	787.2999
f_1	263.1645	594.7633	278.4290	1420.0705	**255.8799**	1371.3888	260.5299	259.7703
f_2	622.4394	1585.8146	628.6117	927.1755	**621.1243**	1624.7289	869.5341	872.5241
f_3	278.0287	3298.0839	292.0504	3303.1495	**261.1934**	1367.3627	1303.8592	979.7108
f_4	358.0799	1460.3739	**356.0119**	1591.3160	360.3514	711.6922	721.3646	646.3844
f_5	**236.5184**	311.3102	248.2506	249.3193	237.4182	1456.7880	251.5607	259.2736
f_6	448.8993	1102.5489	466.0795	1104.2448	**434.8893**	874.0973	440.6255	422.4402
f_7	341.5798	663.7398	344.3478	730.8557	**337.0671**	1381.5354	891.4041	885.4871
f_8	**368.7186**	1308.2763	405.5594	1159.4827	380.6574	1389.6927	1023.6844	886.3957
f_9	345.1321	732.5952	358.2724	1229.2192	**336.9159**	1467.7248	743.4061	814.7588
f_{10}	**377.9289**	1556.1002	392.8346	2195.5736	378.2778	1641.5258	823.5398	880.5851
f_{11}	**619.3468**	950.2162	635.6353	1220.6493	648.2199	1920.6593	862.0097	962.4388

5. Conclusions

The swarm intelligence algorithms perform in varied manner for the same optimization problem. Even for a kind of optimization algorithm, the change of structure or parameters may lead to different results. It is very difficult, if not impossible, to obtain the connection between an algorithm and a problem. Thus, choosing a proper setting for an algorithm before the search is vital for the optimization process. In this paper, we attempt to combine the strengths of different settings for the PSO algorithm. Based on the building blocks thesis, a PSO algorithm with a multiple phase strategy was proposed in this paper to solve single-objective numerical optimization problems. Two variants of the PSO algorithm, which were termed as the PSO with fixed phase (PSOFP) algorithm and PSO with dynamic phase (PSODP) algorithm, were tested on the 12 benchmark functions. The experimental results showed that the combination of different phases could enhance the robustness of the PSO algorithm.

There are two phases in the proposed PSOFP and PSODP methods. However, the number of phases could be increased or the type of phase could be changed for other PSO algorithms with multiple phases. Besides the PSO algorithms, the building block thesis could be utilized in other swarm optimization algorithms. Based on the analysis of different components, the strength and weaknesses of different swarm optimization algorithms could be understood. Utilizing this multiple phase strategy with other swarm algorithms is our future work.

Data Availability

The data and codes used to support the findings of this study have been deposited in the GitHub repository.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was partially supported by the Scientific Research Foundation of Shaanxi Railway Institute (Grant no. KY2016-29) and the Natural Science Foundation of China (Grant no. 61703256).

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*, pp. 39–43, Nagoya, Japan, October 1995.

[2] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of IEEE International Conference on Neural Networks (ICNN)*, pp. 1942–1948, Perth, WA, Australia, November 1995.

[3] R. Eberhart and Y. Shi, "Particle swarm optimization: developments, applications and resources," in *Proceedings of the 2001 Congress on Evolutionary Computation (CEC 2001)*, pp. 81–86, Seoul, Republic of Korea, May 2001.

[4] Y. Shi, "An optimization algorithm based on brainstorming process," *International Journal of Swarm Intelligence Research*, vol. 2, no. 4, pp. 35–62, 2011.

[5] S. Cheng, Q. Qin, J. Chen, and Y. Shi, "Brain storm optimization algorithm: a review," *Artificial Intelligence Review*, vol. 46, no. 4, pp. 445–458, 2016.

[6] L. Ma, S. Cheng, and Y. Shi, "Enhancing learning efficiency of brain storm optimization via orthogonal learning design," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2020.

[7] S. Cheng, X. Lei, H. Lu, Y. Zhang, and Y. Shi, "Generalized pigeon-inspired optimization algorithms," *Science China Information Sciences*, vol. 62, 2019.

[8] Z.-H. Zhi-Hui Zhan, J. Jun Zhang, Y. Yun Li, and H. S.-H. Chung, "Adaptive particle swarm optimization," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 39, no. 6, pp. 1362–1381, 2009.

[9] J. Liu, X. Ma, X. Li, M. Liu, T. Shi, and P. Li, "Random convergence analysis of particle swarm optimization algorithm with time-varying attractor," *Swarm and Evolutionary Computation*, vol. 61, Article ID 100819, 2021.

[10] Q. Qin, S. Cheng, Q. Zhang, L. Li, and Y. Shi, "Particle swarm optimization with interswarm interactive learning strategy," *IEEE Transactions on Cybernetics*, vol. 46, no. 10, pp. 2238–2251, 2016.

[11] X. Xia, L. Gui, F. Yu et al., "Triple archives particle swarm optimization," *IEEE Transactions on Cybernetics*, vol. 50, no. 12, pp. 4862–4875, 2020.

[12] R. Cheng and Y. Jin, "A social learning particle swarm optimization algorithm for scalable optimization," *Information Sciences*, vol. 291, pp. 43–60, 2015.

[13] F. Wang, H. Zhang, and A. Zhou, "A particle swarm optimization algorithm for mixed-variable optimization problems," *Swarm and Evolutionary Computation*, vol. 60, Article ID 100808, 2021.

[14] H. Han, W. Lu, L. Zhang, and J. Qiao, "Adaptive gradient multiobjective particle swarm optimization," *IEEE Transactions on Cybernetics*, vol. 48, no. 11, pp. 3067–3079, 2018.

[15] X.-F. Liu, Z.-H. Zhan, Y. Gao, J. Zhang, S. Kwong, and J. Zhang, "Coevolutionary particle swarm optimization with bottleneck objective learning strategy for many-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 4, pp. 587–602, 2019.

[16] S. Cheng, X. Lei, J. Chen, J. Feng, and Y. Shi, "Normalized ranking based particle swarm optimizer for many objective optimization," in *Simulated Evolution and Learning (SEAL 2017))*, Y. Shi, K. C. Tan, M. Zhang et al., Eds., Springer International Publishing, Newyork, NY, USA, pp. 347–357, 2017.

[17] S. Cheng, H. Lu, X. Lei, and Y. Shi, "A quarter century of particle swarm optimization," *Complex & Intelligent Systems*, vol. 4, no. 3, pp. 227–239, 2018.

[18] S. Cheng, L. Ma, H. Lu, X. Lei, and Y. Shi, "Evolutionary computation for solving search-based data analytics problems," *Artificial Intelligence Review*, vol. 54. , 2021 In press.

[19] J. H. Holland, "Building blocks, cohort genetic algorithms, and hyperplane-defined functions," *Evolutionary Computation*, vol. 8, no. 4, pp. 373–391, 2000.

[20] A. P. Piotrowski, J. J. Napiorkowski, and A. E. Piotrowska, "Population size in particle swarm optimization," *Swarm and Evolutionary Computation*, vol. 58, Article ID 100718, 2020.

[21] J. Kennedy, R. Eberhart, and Y. Shi, *Swarm Intelligence*, Morgan Kaufmann Publisher, Burlington, MA, USA, 2001.

[22] R. Mendes, J. Kennedy, and J. Neves, "The fully informed particle swarm: simpler, maybe better," *IEEE Transactions on Evolutionary Computation*, vol. 8, no. 3, pp. 204–210, 2004.
[23] J. Arabas and K. Opara, "Population diversity of nonelitist evolutionary algorithms in the exploration phase," *IEEE Transactions on Evolutionary Computation*, vol. 24, no. 6, pp. 1050–1062, 2020.
[24] D. Bratton and J. Kennedy, "Defining a standard for particle swarm optimization," in *Proceedings of the 2007 IEEE Swarm Intelligence Symposium (SIS 2007)*, pp. 120–127, Honolulu, HI, USA, April 2007.
[25] M. Clerc, *Standard Particle Swarm Optimisation from 2006 to 2011*, Independent Consultant, Jersey City, NJ, USA, 2012.
[26] Y. Shi and R. Eberhart, "Empirical study of particle swarm optimization," in *Proceedings of the 1999 Congress on Evolutionary Computation (CEC1999)*, pp. 1945–1950, Washington, DC, USA, July 1999.