

# UC Irvine

## UC Irvine Previously Published Works

### Title

Handling boundary constraints for particle swarm optimization in high-dimensional search space

### Permalink

<https://escholarship.org/uc/item/14r1b39q>

### Journal

Information Sciences, 181(20)

### ISSN

1069-0115

### Authors

Chu, Wei

Gao, Xiaogang

Sorooshian, Soroosh

### Publication Date

2011-10-01

### DOI

10.1016/j.ins.2010.11.030

### Copyright Information

This work is made available under the terms of a Creative Commons Attribution License, available at <https://creativecommons.org/licenses/by/4.0/>

Peer reviewed



ELSEVIER

Contents lists available at ScienceDirect

Information Sciences

journal homepage: [www.elsevier.com/locate/ins](http://www.elsevier.com/locate/ins)

# Handling boundary constraints for particle swarm optimization in high-dimensional search space

Wei Chu<sup>\*</sup>, Xiaogang Gao, Soroosh Sorooshian

Department of Civil and Environmental Engineering, University of California, Irvine, CA 92697, USA

## ARTICLE INFO

### Article history:

Available online 3 December 2010

### Keywords:

Particle swarm optimization  
Bound-handling strategy  
High-dimensional optimization  
Complex benchmark functions  
Real-world applications

## ABSTRACT

Despite the fact that the popular particle swarm optimizer (PSO) is currently being extensively applied to many real-world problems that often have high-dimensional and complex fitness landscapes, the effects of boundary constraints on PSO have not attracted adequate attention in the literature. However, in accordance with the theoretical analysis in [11], our numerical experiments show that particles tend to fly outside of the boundary in the first few iterations at a very high probability in high-dimensional search spaces. Consequently, the method used to handle boundary violations is critical to the performance of PSO. In this study, we reveal that the widely used random and absorbing bound-handling schemes may paralyze PSO for high-dimensional and complex problems. We also explore in detail the distinct mechanisms responsible for the failures of these two bound-handling schemes. Finally, we suggest that using high-dimensional and complex benchmark functions, such as the composition functions in [19], is a prerequisite to identifying the potential problems in applying PSO to many real-world applications because certain properties of standard benchmark functions make problems inexplicit.

© 2010 Elsevier Inc. All rights reserved.

## 1. Introduction

Simulating the social behavior of individuals in a swarm that searches for better living conditions in a complex natural environment, the particle swarm optimizer (PSO) [16,17] has become one of the major evolutionary global optimization algorithms. Moreover, it has been applied to a wide range of real-world problems in many fields [5,10,14,18,24,25,28,29,33]. The algorithm starts with a randomly selected initial population and successively evolves individuals of the population successively. The position of an individual particle in the search space ( $x_i$ ) is updated by a displacement called velocity ( $v_i$ ) based on three attributes, namely, (1) the particle's velocity in the previous iteration ( $v_{i-1}$ ), (2) the particle's best-ever position ( $\hat{x}_i$ ), and (3) the swarm's best-ever position ( $\hat{g}$ ),

$$v_i = c_0 v_{i-1} + c_1 r_1 \otimes (\hat{x}_i - x_{i-1}) + c_2 r_2 \otimes (\hat{g} - x_{i-1}), \quad (1)$$

$$x_i = x_{i-1} + v_i,$$

where  $c_0$ ,  $c_1$ , and  $c_2$  are the significance coefficients that control the influence of each of the velocity components;  $r_1, r_2 \in R^D$  are random vectors  $\{r_p = U(0, 1)\}_{p=1}^D$  that perturb the components (2) and (3);  $D$  is the dimensionality of the search space; and  $\otimes$  denotes element-by-element vector multiplication.

<sup>\*</sup> Corresponding author. Tel.: +1 949 2325086.

E-mail address: [wchu2@uci.edu](mailto:wchu2@uci.edu) (W. Chu).

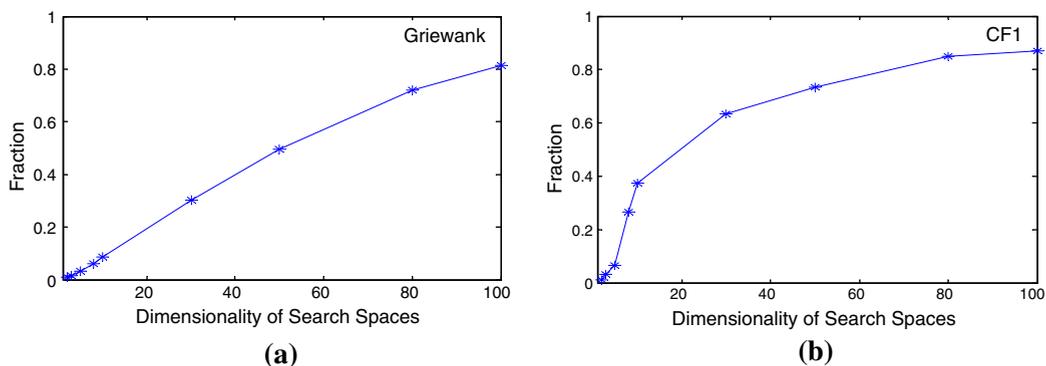
In order to improve the performance of PSO, researchers have recently developed many variants of PSO.

- (1) The inertia weight,  $c_0$ , and acceleration coefficients,  $c_1$  and  $c_2$ , play critical roles in balancing exploration and exploitation capabilities. Therefore, adaptively and dynamically adjusting these coefficients may increase the robustness and efficiency of the search. Shi and Eberhart [30] developed fuzzy methods to nonlinearly update the inertia weight; Chatterjee and Siarry [4] also proposed a scheme for nonlinear inertia weight variation. The concept of time-varying acceleration coefficients was also investigated in [26,32].
- (2) Improved swarm structures and learning schemes are also among the latest advances in PSO. In [12,31], dynamic neighborhoods were used; in [23], Mendes designed a fully informed particle swarm to model the influence on each individual of the best performers among its neighbors. In addition, Liang et al. [20] presented a clever learning strategy through which the historical best information of all other particles is used to update a particular particle's velocity.
- (3) More and more researchers are hybridizing PSO with other search strategies. Van den Bergh and Engelbrecht [2] implemented a cooperative approach to optimize the different components of the solution vector; Liu et al. [21] combined PSO with a chaotic local search. In [9], Gaussian local search and differential mutation were integrated into PSO. Preference order scheme was used to adapt PSO for multi-objective optimization in [34].
- (4) Recently, the study of swarm dynamics has received increasing attention [3,8]. Lozano et al. [22] proposed a new replacement strategy to increase the diversity of the population. In addition, in [1,15,37], swarm dynamics were also used to design and evaluate new PSO variants.

In contrast with these intensively studied aspects of PSO, PSO bound-handling strategies have not drawn adequate attention from researchers. However, in real-world applications, the search spaces are often high-dimensional and bounded in order to preserve the physical meaning of the parameters. Therefore, addressing boundary violations during a search becomes an essential issue due to the causes listed below.

- (1) Boundary violation increases enormously as dimensionality increases. Helwig and Wanka [11] mathematically proved that in a high-dimensional search using PSO, “all particles are initialized very close to the boundary with overwhelming probability, and the global guide is expected to leave the search space in every fourth dimension”. This is further illustrated by Fig. 1, in which the fraction of particles in the swarm flying outside of the boundary in the first iteration is shown based on results from two benchmark functions, namely, Griewank's function and a more complex composition function (CF1). The latter function will be introduced in Section 3.1. In each run, the swarm has a size of  $10 \times$  dimensionality; the number of particles hitting the boundary in the first iteration was counted. For both functions, there exists a clear trend that this fraction increases significantly with the increase of search space dimensionality; moreover, this fraction tends to approach one (Fig. 1).
- (2) Boundary-handling is crucial to the proper functioning of PSO in high-dimensional applications. Our results reveal that improper bound-handling schemes can paralyze PSO when optimizing complex problems. Previous studies on bound-handling mechanisms in PSO [13,35,36] are mostly limited to the use of relatively low-dimensional ( $\leq 50$ ), simple, and standard benchmark functions in which the potential influences of boundary-handling on the performance of PSO are not fully explored. In this study, we utilize 100-D composition benchmark functions in order to better simulate real-world problems and thus investigate the effects of boundary-handling in real-world applications.

The experimental results in this study clearly reveal evident differences in PSO behavior associated with different bound-handling schemes when optimizing high-dimensional and complex problems. Only under the reflecting scheme does PSO behave normally for all of the benchmark functions. “Function normally” means that the swarm finally reaches a minimum point (whether global or local). However, under the random and absorbing schemes, there is a substantial risk that PSO does



**Fig. 1.** The fraction of particles in the swarm flying outside of the boundary in the first iteration as a function of the dimensionality of the benchmark function. The plots are based on the average of 100 independent runs on each function of 2, 3, 5, 8, 10, 30, 50, 80, and 100-D, respectively.

not accomplish this objective. The random scheme may result in extremely slow evolution, and the absorbing scheme may lead the swarm to stagnate to nonstationary boundary points. The responsible mechanisms for the failures of these two bound-handling schemes are investigated. The findings in this study provide essential insight for both the applications and design of PSO for optimizing high-dimensional and complex real-world problems.

The remainder of this paper is as follows: Section 2 describes three fundamental bound-handling schemes addressed in this study. Section 3 presents the experimental configuration using both standard and composition benchmark functions. The comparative results are reported in Section 4. The reasoning behind why the random and absorbing bound-handling schemes paralyze PSO is investigated in Section 5, and the results of experiments on additional composition test functions are presented and discussed in Section 6. How the swarm diversity changes in the evolution process under different bound-handling schemes is explored in Section 7. Finally, Section 8 presents the conclusions of this study.

## 2. Basic bound-handling schemes

Among a variety of bound-handling schemes, the random, reflecting, and absorbing schemes are the most popular and fundamental ones. In addition, they are the basic components of many recently proposed elaborate schemes [13,35]. However, these more elaborate schemes are not discussed in this study because “they do not provide further insight into the swarm’s behavior, and they often even do not yield superior results in comparative experiments” [11]. Hence, only three basic approaches are investigated in this study. In addition, to avoid mixing the effects of adjusting velocity and position at the same time, we also exclude the velocity adjustment scheme from the scope of this study. From Eq. (1), we can see that adjusting position affects the velocity and vice versa. As a result, modifying position and modifying velocity are confounding procedures. Therefore, in practice, many PSO programs only adjust positions when boundary violation occurs. A brief description of each of the basic schemes is addressed below, along with illustrations in Fig. 2.

- (1) Random: This bound-handling scheme is adopted as the default setting in many PSO programs. If a particle flies outside of the boundary of a parameter, a random value drawn from a uniform distribution between the lower and upper boundaries of the parameter is assigned as the corresponding component for the offspring particle.

$$x_i = \begin{pmatrix} \tilde{x}_{i,1} \\ \vdots \\ \tilde{x}_{i,p-1} \\ U(b_l; b_u) \\ \tilde{x}_{i,p+1} \\ \vdots \\ \tilde{x}_{i,D} \end{pmatrix}. \tag{2}$$

- (2) Absorbing: Under the absorbing scheme, a particle flying outside of a parameter’s boundary is relocated at the boundary in that dimension such that the particle appears to be absorbed by the boundary.

$$x_i = \begin{pmatrix} 1 \\ \vdots \\ 1 \\ b_l(\text{or } b_u)/\tilde{x}_{i,p} \\ 1 \\ \vdots \\ 1 \end{pmatrix} \otimes \begin{pmatrix} \tilde{x}_{i,1} \\ \vdots \\ \tilde{x}_{i,p-1} \\ \tilde{x}_{i,p} \\ \tilde{x}_{i,p+1} \\ \vdots \\ \tilde{x}_{i,D} \end{pmatrix}. \tag{3}$$

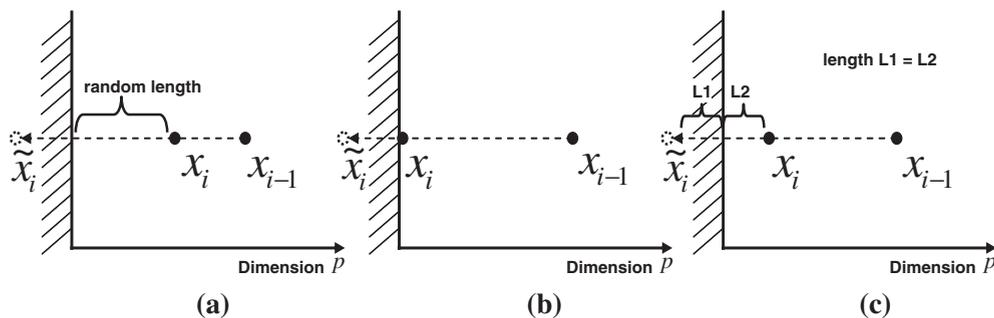


Fig. 2. Schematic explanations for three basic bound-handling schemes: (a) random, (b) absorbing, and (c) reflecting.  $x_{i-1}$  is the particle’s position in last iteration;  $\tilde{x}_i$  is the position that the PSO algorithm projects; and  $x_i$  is the final position.

- (3) Reflecting: In this scheme, when a particle flies outside of a boundary of a parameter, the boundary acts like a mirror and reflects the projection of the particle's displacement.

$$x_i = \begin{pmatrix} 1 \\ \vdots \\ 1 \\ -1 \\ 1 \\ \vdots \\ 1 \end{pmatrix} \otimes \begin{pmatrix} \tilde{x}_{i,1} \\ \vdots \\ \tilde{x}_{i,p-1} \\ \tilde{x}_{i,p} \\ \tilde{x}_{i,p+1} \\ \vdots \\ \tilde{x}_{i,D} \end{pmatrix} + \begin{pmatrix} 0 \\ \vdots \\ 0 \\ 2b_l(\text{or } b_u) \\ 0 \\ \vdots \\ 0 \end{pmatrix}. \tag{4}$$

### 3. Experiments

One of the reasons why bound-handling mechanisms have not attracted adequate attention from researchers is that simple standard benchmark functions are mostly used to examine the performances of PSO algorithms. Certain properties of these standard benchmark functions make the performance of optimization algorithms insensitive to bound-handling. Among these properties are (1) the symmetry of the fitness landscape and (2) the global minimum of the benchmark function located at the origin (or the center of the search domain). In real-world applications, the fitness landscape is irregular and complex, rarely having these ideal properties. Consequently, bound-handling mechanisms play a critical role, as demonstrated in the following experiments.

#### 3.1. Test functions

Two typical benchmark functions are compared. The first is the 100-D Griewank's function, which is a commonly used standard benchmark function with a symmetric fitness landscape and the global optimum locating at the origin. The second function is a 100-D composition function (CF1), which was designed by Liang et al. [19]:

$$f(x) = \sum_{i=1}^{10} w_i(x) [f'_i((x - o_i)/\lambda_i) + bias_i], \quad i = 1, \dots, 10, \quad bias_i = (i - 1) \times 100, \tag{5}$$

where

$f_i(x)$  is the  $i$ th standard benchmark function.

$f'_i(x) = 2000f_i(x)/|f_{\max i}|, f_{\max i} = \max\{f_i(x), \quad i = 1, \dots, 10\}$ .

$w_i$  is the weighting function and is calculated as follows:

$$w_i(x) = \exp\left(-\frac{f_i(x - o_i)}{200\sigma_i^2}\right),$$

$$w_i = \begin{cases} w_i & \text{if } w_i = \max(w_i), \\ w_i(1 - \max(w_i)^{10}) & \text{if } w_i \neq \max(w_i). \end{cases}$$

$$w_i = w_i / \sum_{j=1}^{n=10} w_j.$$

This function is composed of ten standard test functions  $f_i$ , which are ten sphere functions in this case (Table 1). As described by the above equations,  $f'_i$  has its minimum of  $bias_i$  located at  $o_i$ . Therefore,  $f(x)$  has its global minimum of 0 at  $o_1$  and nine major local minima of  $i \times 100$  ( $i = 1, \dots, 9$ ) at  $o_{2-10}$ , respectively.  $o_{1-9}$  are locations randomly generated in the search space, whereas  $o_{10}$  is set at the origin in order to trap algorithms that take advantage of this special location.  $\lambda_i$  is used to stretch ( $\lambda_i > 1$ ) or compress ( $\lambda_i < 1$ ) function  $f'_i(x)$ .  $\sigma_i$  is the variable that controls the coverage range of  $f'_i(x)$ , and a small  $\sigma_i$  value results in a narrow range. For CF1,  $\sigma_{1-10} = 1$ , and  $\lambda_{1-10} = 0.05$  (Table 2).

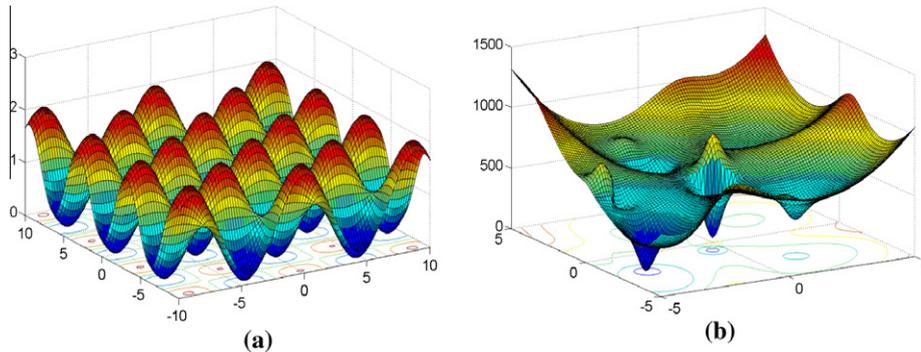
This function's fitness landscape has no symmetry or any other obvious patterns, which makes it a good representative of real-world problems. Fig. 3 shows the fitness landscapes of both test functions in two-dimensional space. More details about the construction of composition functions can be found in [19].

**Table 1**  
Selection of standard functions of composition function CF2–CF6.

	CF2	CF3	CF4	CF5	CF6
$f_{1-2}$	Griewank	Griewank	Ackley	Rastrigin	Rastrigin
$F_{3-4}$	Griewank	Griewank	Rastrigin	Weierstrass	Weierstrass
$F_{5-6}$	Griewank	Griewank	Weierstrass	Griewank	Griewank
$F_{7-8}$	Griewank	Griewank	Griewank	Ackley	Ackley
$F_{9-10}$	Griewank	Griewank	Sphere	Sphere	Sphere

**Table 2**  
Values of  $\lambda_i$  and  $\sigma_i$  of composition function CF2–CF6.

	$\sigma_1, \lambda_1$	$\sigma_2, \lambda_2$	$\sigma_3, \lambda_3$	$\sigma_4, \lambda_4$	$\sigma_5, \lambda_5$	$\sigma_6, \lambda_6$	$\sigma_7, \lambda_7$	$\sigma_8, \lambda_8$	$\sigma_9, \lambda_9$	$\sigma_{10}, \lambda_{10}$
CF2	1, 0.05	1, 0.05	1, 0.051	1, 0.05	1, 0.05	1, 0.05	1, 0.05	1, 0.05	1, 0.05	1, 0.05
CF3	1, 1	1, 1	1, 1	1, 1	1, 1	1, 1	1, 1	1, 1	1, 1	1, 1
CF4	1, 5/32	1, 5/32	1, 1	1, 1	1, 10	1, 10	1, 0.05	1, 0.05	1, 0.05	1, 0.05
CF5	1, 0.2	1, 0.2	1, 10	1, 10	1, 0.05	1, 0.05	1, 5/32	1, 5/32	1, 0.05	1, 0.05
CF6	0.1, $0.1 \times 1/5$	0.2, $0.2 \times 1/5$	0.3, $0.3 \times 0.5$	0.4, $0.4 \times 0.5$	0.5, $0.5 \times 0.05$	0.6, $0.6 \times 0.05$	0.7, $0.7 \times 5/32$	0.8, $0.8 \times 5/32$	0.9, $0.9 \times 0.05$	1, $1 \times 0.05$



**Fig. 3.** Fitness landscapes for (a) Griewank's function and (b) composition function CF1 in two dimensions.

### 3.2. Experimental settings

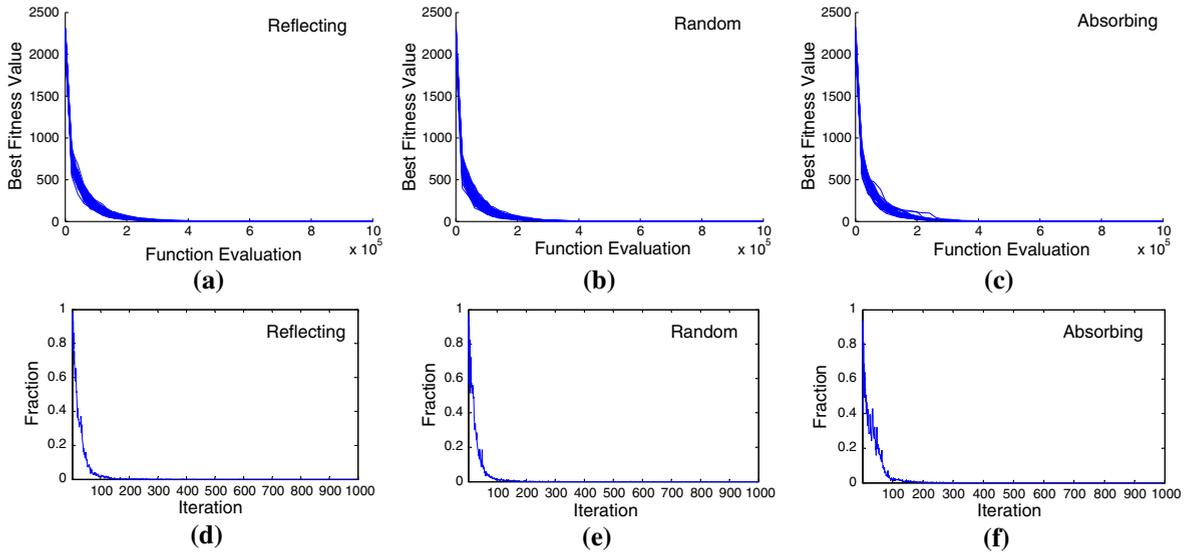
The algorithmic coefficients of PSO are set at widely used values. The inertia weight is held at a constant  $c_0 = 0.5$ ;  $c_1$  and  $c_2$  are both set at 2, and  $v_{\max}$  equals half of the search range. A population of 1000 particles is randomly generated with a uniform distribution in the search space. For Griewank's function, the lower and upper bounds of every dimension are  $-600$  and  $600$ , respectively, and for CF1, these values are  $-5$  and  $5$ , respectively. The global minimum and the major local minima of CF1 are all randomly set within the range of  $[-4.5, 4.5]$  in every dimension.

## 4. Results

For each benchmark function, PSO is implemented under three different bound-handling schemes. For each scheme, 50 randomly initialized runs were conducted; the mean and standard deviation of the final function values are listed in Table 3. Furthermore, to test if the results of runs with the reflecting scheme are significantly different from the results of runs using the other schemes, Wilcoxon rank sum tests were conducted. The null assumption is  $H_0$ : that there is no difference between the results of the best scheme and the results of the second best scheme. For each function, if one bound-handling scheme leads to a significantly better result as compared with the other two schemes at the 5% significance level, then the corresponding  $P$ -value is listed in Table 3. If no bound-handling scheme leads to a significantly better result, it is labeled as "not significant". The test reveals that, on Griewank's function, the selection of a bound-handling scheme does not affect the performance of PSO, since results from runs using different bound-handling schemes are not significantly different. In contrast, on the CF1 function, the runs using the reflecting bound-handling scheme yield significantly better results than the runs under other two schemes, as the corresponding  $P$ -value is extremely small.

**Table 3**  
Experimental results.

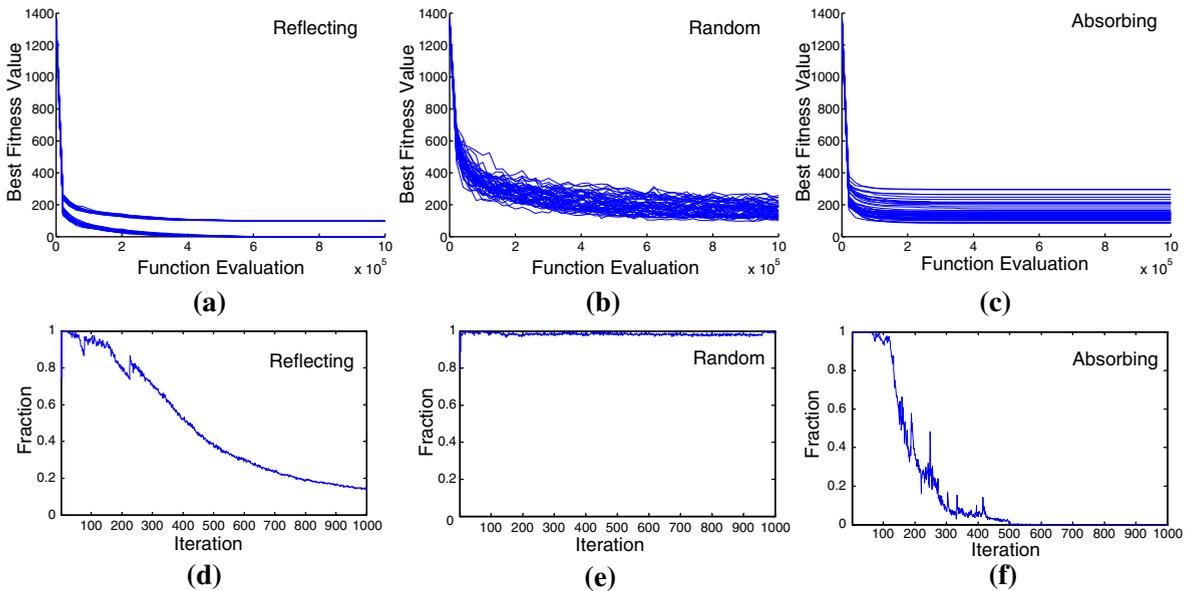
	Reflecting	Random	Absorbing	$P$ -value
Griewank's	$0.1269 \pm 0.0050$	$0.1016 \pm 0.0040$	$0.1103 \pm 0.0029$	Not significant
CF1	$32.00 \pm 47.12$	$148.2 \pm 38.09$	$156.7 \pm 54.80$	$4.550e-17$
CF2	$210.1 \pm 112.9$	$327.3 \pm 111.5$	$312.5 \pm 117.7$	$1.377e-04$
CF3	$434.9 \pm 362.5$	$676.9 \pm 379.5$	$717.3 \pm 319.5$	$1.629e-06$
CF4	$900.0 \pm 0.000$	$900.0 \pm 0.000$	$917.5 \pm 40.40$	Not significant
CF5	$74.20 \pm 82.26$	$175.7 \pm 44.43$	$185.2 \pm 95.05$	$2.682e-11$
CF6	$900.0 \pm 0.000$	$900.0 \pm 0.000$	$901.9 \pm 7.780$	Not significant



**Fig. 4.** Upper row: the best fitness value versus the number of function evaluations for all 30 runs of the 100-D Griewank's function under the (a) reflecting, (b) random, and (c) absorbing bound-handling schemes. Lower row: the fraction of particles in the swarm flying outside of the boundary in each iteration for a randomly selected run under the (d) reflecting, (e) random, and (f) absorbing bound-handling schemes.

To further illustrate the effects of each scheme on the behavior of PSO, the fitness curves for all runs are plotted in Figs. 4 and 5.

Shown in the upper row of Fig. 4, the best fitness value of the swarm reaches very close to the global minimum in all of the runs with similar convergence rates. In addition, the records of particle positions indicate that the swarm has converged into a small vicinity around the global minimum in every run. The lower row in Fig. 4 shows how the fraction of particles flying outside of the boundary changes as the evolution proceeds. For each scheme, the result from a randomly selected run is plotted. All of the other runs show similar patterns. In light of the studies in [11], it is not surprising that in the first few iterations, the majority of the particles fly outside of the boundary. However, as the particles all converge to the global minimum (that is, the center of the search domain), this fraction drops quickly to zero after around 200 iterations, regardless of which bound-handling scheme is adopted.



**Fig. 5.** Upper row: the best fitness value versus the number of function evaluations for all 30 runs of the 100-D CF1 function under the (a) reflecting, (b) random, and (c) absorbing bound-handling schemes. Lower row: the fraction of particles in the swarm flying outside of the boundary in each iteration for a randomly selected run under the (d) reflecting, (e) random, and (f) absorbing bound-handling schemes.

In contrast, in the experiments using CF1, the performance of PSO under different bound-handling schemes distinctly diverges, as seen in the upper row in Fig. 5. Under the reflecting scheme, PSO reduces the best fitness value of the swarm quickly, and eventually, the best values successfully achieve the global minimum in most runs. However, during a few runs, a major local minimum is reached instead of the global one. Under the random scheme, the improvement of the best fitness value decreases rapidly after about  $2 \times 10^5$  function evaluations, and no run discovers the global minimum or any of the major minima within the maximum number of function evaluations. Under the absorbing scheme, the best fitness value decreases as quickly as in the reflecting runs. Every run quickly arrives at its final best fitness value and stays at that value thereafter. However, none of these final best values is equal to the global minimum or any of the major local minima.

As seen in the lower row in Fig. 5, changes in the fraction of the swarm flying outside of the boundary also depends highly on the bound-handling scheme. Similar to Griewank's function experiments, the fractions are high in the first several iterations due to the high dimensionality of the search space (that is, 100-D). From there on, the patterns diverge greatly with different schemes. For the runs under the reflecting scheme, this fraction drops steadily but maintains a value substantially different from 0 throughout the evolution. For the runs under the random scheme, a very high fraction (close to one) of particles in the swarm continues to hit the boundary throughout the entire optimization process. Finally, for the runs under the absorbing bound-handling scheme, this fraction drops to zero in midway through the evolution.

The records of these runs show distinguishing processes with respect to the evolution of the swarm.

In the reflecting runs, although the global minimum has been achieved in early iterations, the population has not converged to the global minimum by the end of the last (1000th) iteration. A large number of particles in the population continue to evolve and fly outside of the boundary throughout the entire process.

In the random run, the random bound-handling process (Eq. (2)) is overwhelmingly activated through the evolution, and, therefore, the generated new points always tend to be close to the boundary, which explains why particles always tend to fly outside of the boundary.

In the absorbing run, the absorbing process (Eq. (3)) makes all of the particles of the swarm adhere quickly to the boundary in a single dimension, which means that the corresponding components ( $x_{i,p}$ ) of that dimension in all particles are replaced by a single value. This value is one of the boundary values, namely,  $b_l$  or  $b_u$ , depending on whether the upper or lower bound is hit. While the PSO process continues, the absorbing process drives the swarm particles to hit and stick to the boundary in more and more dimensions, until finally the population converges into a single point on the boundary of the search space. We find that all 30 absorbing runs duplicate the same process and converge to different single points on the boundary. None of these points is a stable or near-stable point in the search space (i.e., the gradient of the object function at the point is equal or close to zero).

Here, the experimental results strongly support the argument that bound-handling is critical to the success of PSO for high-dimensional complex problems. In the following section, we aim to understand the mechanisms that cause PSO to fail under the random and absorbing schemes.

## 5. Failure of PSO under the random and absorbing schemes

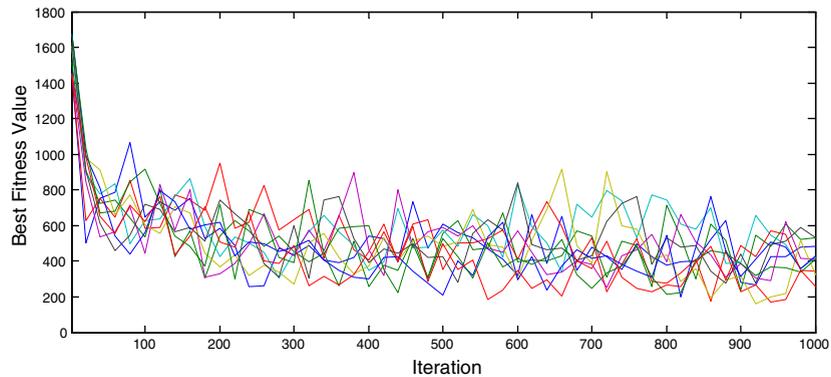
To understand why these two schemes lead PSO to failure in every run of the 100-D CF1 function, the evolution paths of the individual particles are traced.

### 5.1. The random scheme

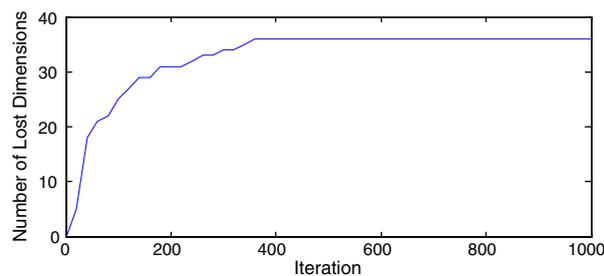
As illustrated in Fig. 5(e), almost all particles in the initial population fly outside of the boundary in the first iteration, which triggers the random scheme (Eq. (2)) to generate new positions for the particles in the search space. Therefore, the first iteration actually repeats the same random process that initiates the population. In the same manner, the subsequent iterations also repeat this random process, and random sampling dominates the entire evolution. Consequently, the PSO process (Eq. (1)) hardly has an opportunity to work. In this situation, PSO is merely acting as a random-searching algorithm. If the function has a well-defined attractive basin that can drag the swarm away from the boundary and drive particles toward the global minimum, such as in Griewank's function, this random sampling may frequently find better positions away from the boundary so that the swarm can converge. However, if the test function has a complex fitness surface like CF1, random sampling can hardly find better positions, because the effectiveness of random sampling degrades immensely with an increase in the dimensionality of the search space. Therefore, as the evolution goes on and the fitness values drop to a certain level, it becomes difficult for the swarm to make further progress. Fig. 6 illustrates the variation of fitness values during the evolution of 100 randomly selected particles in the same CF1 run already shown in Fig. 5(e). All of the particles improve significantly only within the first 200 iterations. Afterward, a particle's fitness value fluctuates due to the inefficiency of the random-sampling process, and improvements begin happening extremely slowly.

### 5.2. Absorbing scheme

Zhang et al. [36] identified premature convergence of PSO on the boundary caused by the absorbing scheme. Helwig and Wanka [11] attributed this to the fact that the particles located on the boundary generally have better fitness values as compared with the initial fitness values of other particles in the swarm; therefore, they tend to attract the whole population.



**Fig. 6.** The evolution of 100 randomly selected particles from the same run of the CF1 function as shown in Fig. 5(e) which is under the random bound-handling scheme.



**Fig. 7.** The number of lost dimensions versus the iteration number for the same run as shown in Fig. 5(f). “Lost dimension” is defined as the dimension in which all of the particles, along with their best-ever positions and the swarm’s best-ever position, are located on the same boundary.

However, we doubt this argument because it does not explain why the absorbing scheme succeeded consistently on Griewank’s function. By tracing the particle evolution paths in the CF1 runs, we discover that due to the lack of symmetry in the fitness landscape, particles are more likely to hit the boundary in certain dimensions than in the others as the swarm evolves. This, in fact, results in the entire population hitting and sticking to the boundary in certain dimensions. Very likely, the best-ever positions of individual particles,  $\hat{x}_i$ , and the best-ever position of the swarm,  $\hat{g}$ , are also attracted to the boundary in these dimensions. We name these dimensions the “lost dimensions”. If lost dimensions occur, the entire swarm, along with  $\hat{g}$  and all  $\hat{x}_i$ , can span only a subspace of the full search space, which excludes the lost dimensions because the projections of all particles as well as  $\hat{g}$  and all  $\hat{x}_i$  on the lost dimensions are actually a single point on the boundary. Because PSO is a linear process (Eq. (1)), the swarm can no longer move in these lost dimensions; instead, it can only explore in the subspace and thus eventually converge to the minimum of this subspace. Very often, this minimum point is not even a stable point in the original search space, as observed in our experimental results, which is referred to as “stagnation to non-stationary points” as graphically illustrated in [6]. Fig. 7 shows that for the same run as in Fig. 5(f), the number of lost dimensions continues to increase before the population converges to a single point. By the end of the run, 36 lost dimensions are identified. It is important to mention that PSO is not the only evolutionary algorithm suffering from lost dimensions. Chu et al. [7] invest the impact of lost dimensions on several popular algorithms, and provide a tool to solve this issue.

However, why does the absorbing scheme work well on Griewank’s function? The answer is twofold.

- (1) For a symmetric response surface, particles have the same probability of flying outside of the boundary in all dimensions so that the possibility of the entire population hitting the boundary in the same dimension is very small. We examine the records of all 30 runs on Griewank’s function and do not find any occurrence of the entire swarm flying outside of the boundary in the same dimension.
- (2) Due to the well-defined attractive region of the global minimum at the center, the best-ever positions ( $\hat{g}$  and  $\hat{x}_i$ ) usually are not located on the boundary because better positions can easily be found away from the boundary. Therefore, even if all particles hit the boundary in the same dimension, the best-ever positions can still drag them away from the boundary through the update of the velocity (Eq. (1)).

## 6. Experiments using more composition benchmark functions

The other five composition functions (namely, CF2 through CF6) in [19] were also used as benchmark functions to test if the above phenomena occur in more general cases. All of these composition functions are constructed in the same manner as

CF1. In addition to the sphere function, CF2–CF6 also chose from more sophisticated standard functions, which include the Ackely, Griewank, Rastrigin, and Weierstrass functions, as listed in Table 1. Values for parameters  $\lambda_i$  and  $\sigma_i$  are listed in Table 2.

Apparently, CF2–CF6 are more complex than CF1, as demonstrated by the two-dimensional response surfaces plotted in Fig. 8. Like real-world problems, these functions are highly irregular and pose major difficulties for direct-search algorithms, including a hidden global minimum region, numerous local minima, and noisy or deceptive fitness landscapes. Similar to CF1, the global minimum and all of the major local minima of these functions are randomly set within the range of  $[-4.5, 4.5]$  in every dimension.

Using the same settings as in Section 3.2, PSO is implemented under the three bound-handling schemes on 100-D CF2–CF6 with 50 independent runs for each function; Wilcoxon rank sum tests were also conducted to test the significance of differences between runs with different bound-handling schemes. Results are presented in Table 3. For functions CF2, CF3, and CF5, runs with the reflecting scheme yield much smaller final function values as compared with the other runs, and this difference is significant at the 5% significance level, given the extreme low  $P$ -values in the Wilcoxon test. In contrast, for functions CF4 and CF6, the differences between the schemes are insignificant at the 5% significance level.

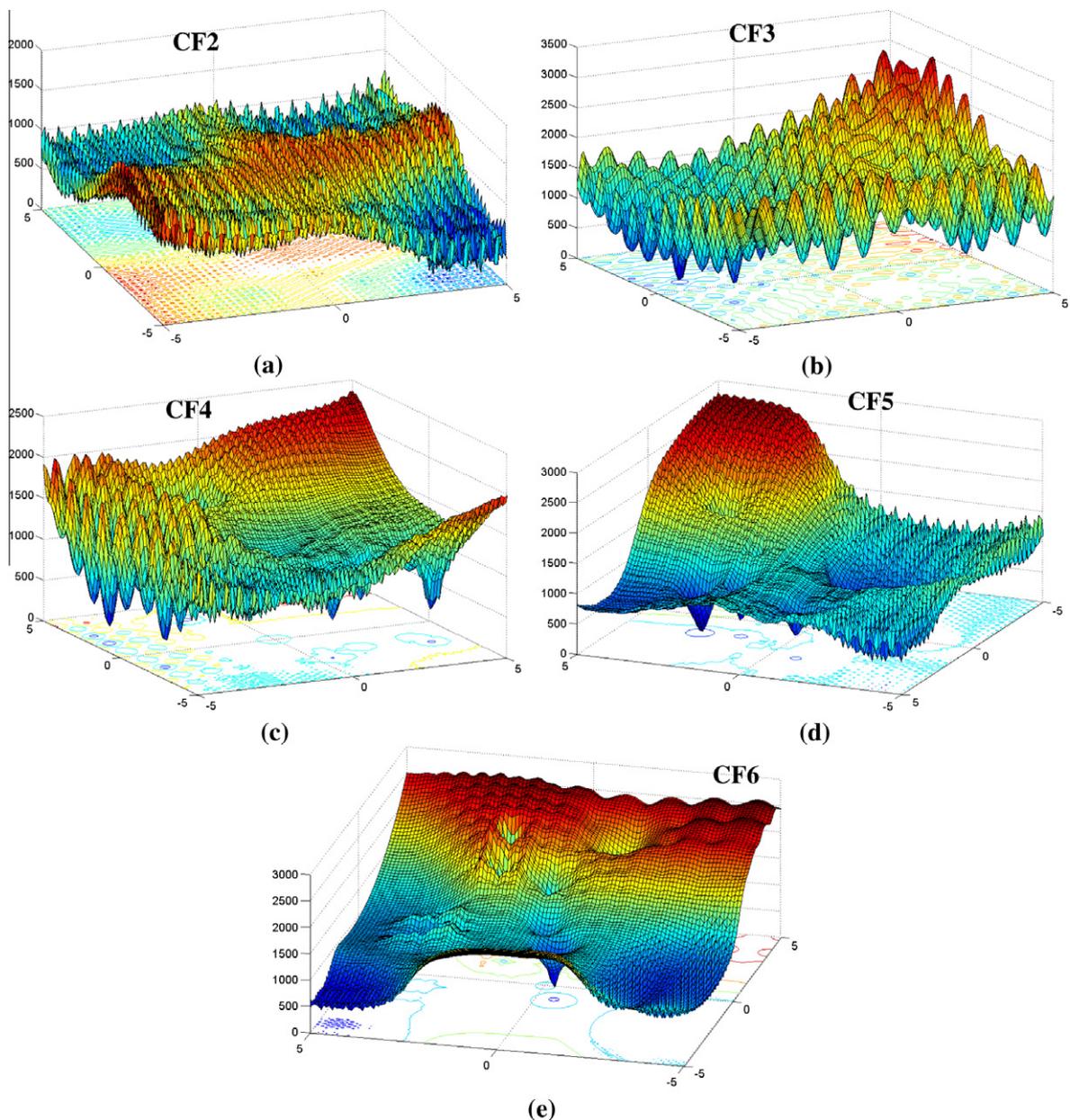


Fig. 8. The two-dimensional fitness landscapes of the composition functions CF2–CF6 ((a)–(e), respectively).

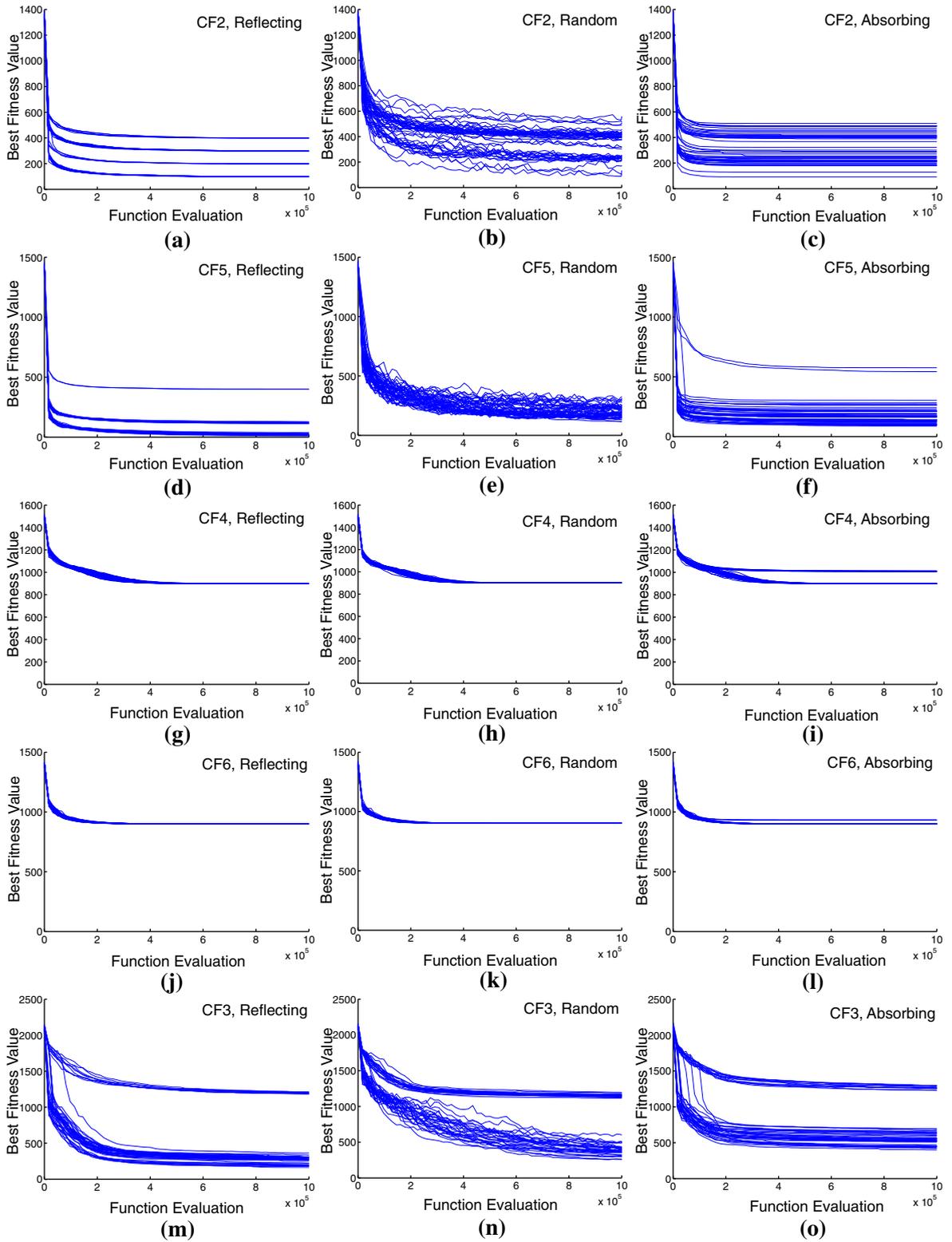


Fig. 9. The best fitness value versus the number of function evaluations for 30 independent PSO runs on 100-D CF2–CF6 under the reflecting bound-handling (left column), random bound-handling (middle column), and absorbing (right column) bound-handling schemes.

The fitness curves of all runs are plotted in Fig. 9 and are summarized as follows:

- (1) The results of CF2 and CF5 resemble the results of CF1 (Fig. 5), except that the global minimum is not achieved in any run. Records of particle positions reveal that for CF5 under the reflecting scheme, some runs discover the attractive region of the global minimum but are trapped by local minima within it. Furthermore, for runs under the random and absorbing schemes, the processes that are responsible for the failures of PSO in the CF1 runs are all observed.
- (2) The results of CF4 and CF6 resemble the results of Griewank’s function (Fig. 4), except that most runs converge into the vicinity of the origin, whereas only eight runs on CF4 and three runs on CF6 converge to points on the boundary under the absorbing scheme. This observation results from the fact that the global and the first eight (that is,  $i = 2-9$  in Eq. (5)) major local minima of CF4 and CF6 have relatively small attractive regions, and the last major local minimum ( $i = 10$  in Eq. (5)) has its attractive region that dominates the search space – see the values of related  $\lambda_i$  and  $\sigma_i$  in Table 2. From Table 1, it is known that for CF4 and CF6, the last standard functions are both sphere functions, with the optimum at the origin. In 100-D space, it is extremely difficult for the swarm to succeed on deceptive fitness landscapes, and hence, the dominant local minimum often traps the swarm.
- (3) For CF3, the results are not so straightforward. As illustrated in the last row in Fig. 9, runs diverge into two groups based on their final best fitness values, including Group I, with value  $>1000$ , and Group II, with value  $<1,000$ . For Group I, swarms are all trapped by the attractive region of a local minimum that is close to the origin. Therefore, particles behave similarly as in runs of Griewank’s function, and no evident difference resulting from different bound-handling schemes is observed. However, for Group II, swarms succeed in escaping from the attractive region that traps Group I and, therefore, make better progress in evolving particles. Similar to the results of CF2 and CF5, the processes that cause failed runs on CF1 under the random and absorbing schemes are all identified in the runs of Group II.

Observations from experiments on CF2–CF6 provide further support to our inference regarding the effects of bound-handling schemes, although more challenges still exist for the success of PSO (none of the runs achieve the global minimum).

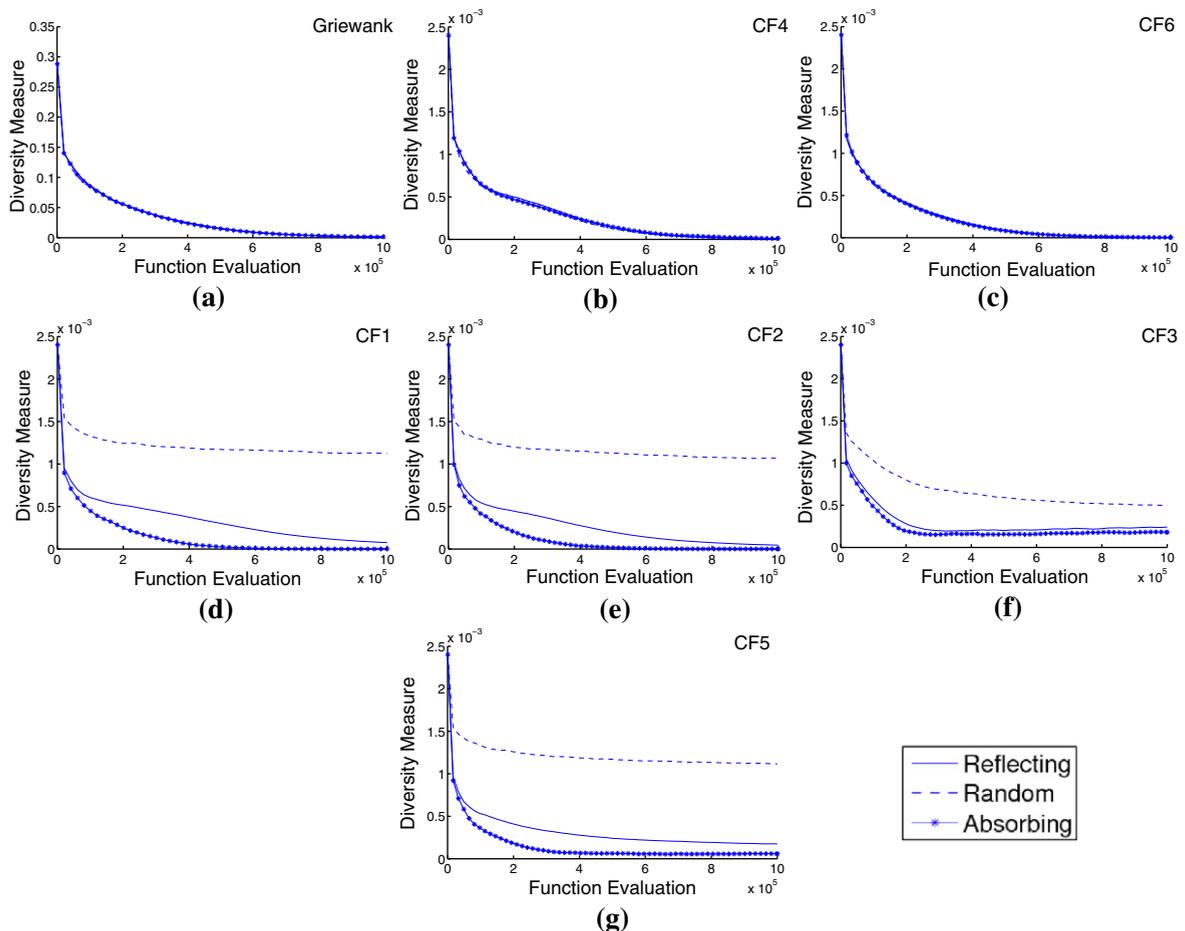


Fig. 10. Average swarm diversity changes during the evolution process.

## 7. Swarm diversity

The diversity of the swarm during the evolution is also examined. To quantify diversity, we adopt the diversity measure suggested in [27].

$$\text{diversity}(S) = \frac{1}{|S| \cdot |L|} \cdot \sum_{i=1}^{|S|} \sqrt{\sum_{j=1}^N (p_{ij} - \bar{p}_j)^2}. \quad (6)$$

Note that  $S$  is the swarm,  $|S|$  is the swarm size,  $|L|$  is the length of the longest diagonal in the search space,  $N$  is the dimensionality,  $p_{ij}$  is the  $j$ th value of the  $i$ th particle, and  $\bar{p}_j$  is the  $j$ th value of the average point  $\bar{p}$ . This measure is independent of swarm size, dimensionality, and search range.

The mean diversity of 50 runs through the evolution process was calculated for every experiment and is illustrated in Fig. 10. The plots of diversity correspond to the plots of fitness curves very well. For Griewank's function, CF4, and CF6, for which fitness curves are very similar across runs under different bound-handling schemes, there is no evident difference between the average diversity curves. This is due to the fact that the symmetry of the response surfaces obscures the significance of bound-handling. However, for the other functions, the fitness curves of runs under the random scheme all drop much slower than those of runs under the other two schemes. This is because the random scheme causes the swarm to evolve very slowly due to the inefficiency of the random-sampling process. Fitness curves for runs under the reflecting and absorbing schemes demonstrate similar trends, but diversity drops slower in the reflecting runs than in the absorbing runs, on average.

## 8. Conclusions

A series of numerical experiments is conducted to investigate the effects of bound-handling schemes on the application of PSO to high-dimensional (100-D) complex problems. Three basic bound-handling strategies, namely, the random, absorbing, and reflecting schemes, are tested using standard and composition benchmark functions. The results highlight certain facts, problems, and insights regarding the performance of PSO in terms of research and practical applications. The main conclusions are summarized as follows:

- (1) The role that the bound-handling scheme plays in PSO applications becomes critical when dealing with high-dimensional complex problems. This is because as the dimensionality increases, the majority of particles in a swarm tend to fly outside of the boundary of the search space in the beginning of the PSO search.
- (2) There is a potential risk that PSO applications may fail when the random and absorbing bound-handling schemes are used for high-dimensional and complex problems. With the failure of the random scheme, the population evolution becomes extremely slow, and the fitness values of the particles tend to oscillate strongly (Fig. 6). Regarding failure under the absorbing scheme, the population converges quickly to an unstable point on the boundary. Only the reflecting scheme allows the PSO process (Eq. (1)) to work normally in all of our experiments.
- (3) We now understand that the failures of PSO under the random and absorbing bound-handling schemes are accompanied with the occurrence of the following two distinct processes.
  - (a) In the failed runs under random bound-handling, the random-sampling (Eq. (2)) procedure becomes the dominant process, and the PSO process (Eq. (1)) is depressed.
  - (b) In the failed runs under absorbing bound-handling, the combined action of the absorbing process (Eq. (3)) and the PSO process (Eq. (1)) causes all of the particles in the population and all of the best-ever positions to adhere to the boundary of the same dimensions and thus finally converge to a single point on the boundary of the search space.
- (4) We recognize that in order to explore and examine the possible problems of applying PSO to complicated real-world applications, composition benchmark functions with irregular fitness landscapes are appropriate for analysis. The widely used standard benchmark functions that have symmetric fitness landscapes and global optima at the origin can sometimes obscure problems.

These results are of great importance to PSO algorithm developers, and PSO users.

## Acknowledgements

The authors gratefully acknowledge the valuable and constructive suggestions provided by the anonymous reviewers. The authors also thank Dr. P. N. Suganthan for kindly providing codes for the composition benchmark functions and the PSO algorithm used in this study. Computation support was provided by Dan Braithwaite. Corrie Thies is also appreciated for her careful and professional editing of the manuscript.

## References

- [1] S. Aine, R. Kumar, P.P. Chakrabarti, Adaptive parameter control of evolutionary algorithms to improve quality-time trade-off, *Applied Soft Computing* 9 (2009) 527–540.

- [2] F. van den Bergh, A.P. Engelbrecht, A cooperative approach to particle swarm optimization, *IEEE Transactions on Evolutionary Computation* 8 (3) (2004) 225–239.
- [3] F. van den Bergh, A.P. Engelbrecht, A study of particle swarm optimization particle trajectories, *Information Sciences* 176 (8) (2006) 937–971.
- [4] A. Chatterjee, P. Siarry, Nonlinear inertia weight variation for dynamic adaptation in particle swarm optimization, *Computers and Operations Research* 33 (3) (2006) 859–871.
- [5] C. Chen, F. Ye, Particle swarm optimization algorithm and its application to clustering analysis, in: *Proceedings of IEEE International Conference on Networking, Sensing and Control*, 2004, pp. 789–794.
- [6] W. Chu, X. Gao, S. Sorooshian, Improving the shuffled complex evolution scheme for optimization of complex nonlinear hydrological systems: Application to the calibration of the Sacramento soil-moisture accounting model, *Water Resources Research* 46 (2010) W09530, doi:10.1029/2010WR009224.
- [7] W. Chu, X. Gao, S. Sorooshian, A solution to the crucial problem of population degeneration in high-dimensional evolutionary optimization, *IEEE Systems Journal*, submitted for publication.
- [8] M. Clerc, J. Kennedy, The particle swarm-explosion, stability, and convergence in a multidimensional complex space, *IEEE Transactions on Evolutionary Computation* 6 (1) (2002) 58–73.
- [9] W. Du, B. Li, Multi-strategy ensemble particle swarm optimization for dynamic optimization, *Information Sciences* 178 (15) (2008) 3096–3109.
- [10] R.C. Eberhar, S. Yuhui, Tracking and optimizing dynamic systems with particle swarms, in: *Proceedings of the Congress on Evolutionary Computation*, 2001, pp. 94–100.
- [11] S. Helwig, R. Wanka, Particle swarm optimization in high-dimensional bounded search spaces, in: *Proceedings of IEEE Swarm Intelligence Symposium*, 2007, pp. 198–205.
- [12] X. Hu, R.C. Eberhart, Multiobjective optimization using dynamic neighborhood particle swarm optimization, in: *Proceedings of the Congress on Evolutionary Computation*, Honolulu, HI, 2002, pp. 1677–1681.
- [13] T. Huang, A.S. Mohan, A hybrid boundary condition for robust particle swarm optimization, *IEEE Antennas and Wireless Propagation Letters* 4 (2005) 112–117.
- [14] C. Juang, A hybrid of genetic algorithm and particle swarm optimization for recurrent network design, Part B: *Cybernetics* 34 (2) (2004) 997–1006.
- [15] Q. Kang, L. Wang, Q.D. Wu, A novel ecological particle swarm optimization algorithm and its population dynamics analysis, *Applied Mathematics and Computation* 205 (2008) 61–72.
- [16] J. Kennedy, R.C. Eberhart, *Swarm Intelligence*, Morgan Kaufman Academic Press, San Matteo, CA, 2001.
- [17] J. Kennedy, R.C. Eberhart, Particle swarm optimization, in: *Proceedings of IEEE International Conference on Neural Networks*, 1995, pp. 1942–1948.
- [18] X. Li, J. Wang, A steganographic method based upon JPEG and particle swarm optimization algorithm, *Information Sciences* 177 (15) (2007) 3099–3109.
- [19] J.J. Liang, P.N. Suganthan, K. Deb, Novel composition test functions for numerical global optimization, in: *Proceedings of IEEE Swarm Intelligence Symposium*, 2005, pp. 68–75.
- [20] J.J. Liang, A.K. Qin, P.N. Suganthan, S. Baskar, Comprehensive learning particle swarm optimizer for global optimization of multimodal functions, *IEEE Transactions on Evolutionary Computation* 10 (3) (2006) 281–295.
- [21] B. Liu, L. Wang, Y.H. Jin, F. Tang, D.X. Huang, Improved particle swarm optimization combined with chaos, *Chaos, Solitons and Fractals* 25 (2005) 1261–1271.
- [22] M. Lozano, F. Herrera, J.R. Cano, Replacement strategies to preserve useful diversity in steady-state genetic algorithms, *Information Sciences* 178 (2008) 4421–4433.
- [23] R. Mendes, J. Kennedy, J. Neves, The fully informed particle swarm: simpler, maybe better, *IEEE Transactions on Evolutionary Computation* 8 (3) (2004) 204–210.
- [24] Y. Marinakis, M. Marinaki, M. Doumpos, C. Zopounidis, Ant colony and particle swarm optimization for financial classification problems, *Expert Systems with Applications* 36 (7) (2009) 10604–10611.
- [25] J. Park, K. Lee, J. Shin, K.Y. Lee, A particle swarm optimization for economic dispatch with nonsmooth cost functions, *IEEE Transactions on Power Systems* 20 (1) (2005) 34–42.
- [26] A. Ratnaweera, S.K. Halgamuge, H.C. Watson, Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients, *IEEE Transactions on Evolutionary Computation* 8 (3) (2004) 240–255.
- [27] J. Riget, J.S. Vesterstroem, A diversity-guided particle swarm optimizer—the ARPSO, Technical Report 2002-02, EVA Life, Department of Computer Science, University of Aarhus, 2002, pp. 1–13.
- [28] J. Robinson, Y. Rahmat-Samii, Particle swarm optimization in electromagnetics, *IEEE Transactions on Antennas and Propagation* 52 (2) (2004) 397–407.
- [29] A. Salman, I. Ahmad, S. Al-Madani, Particle swarm optimization for task assignment problem, *Microprocessors and Microsystems* 26 (8) (2002) 363–371.
- [30] Y. Shi, R.C. Eberhart, Particle swarm optimization with fuzzy adaptive inertia weight, in: *Proceedings of Workshop Particle Swarm Optimization*, Indianapolis, IN, 2001, pp. 101–106.
- [31] P.N. Suganthan, Particle swarm optimizer with neighborhood operator, in: *Proceedings of the Congress on Evolutionary Computation*, Washington, DC, 1999, pp. 1958–1962.
- [32] P.K. Tripathi, S. Bandyopadhyay, S.K. Pal, Multi-objective particle swarm optimization with time variant inertia and acceleration coefficients, *Information Sciences* 177 (22) (2007) 5033–5049.
- [33] M.P. Wachowiak, R. Smolikova, Y. Zheng, J.M. Zurada, A.S. Elmaghraby, An approach to multimodal biomedical image registration utilizing particle swarm optimization, *IEEE Transactions on Evolutionary Computation* 8 (3) (2004) 289–301.
- [34] Y. Wang, Y. Yang, Particle swarm optimization with preference order ranking for multi-objective optimization, *Information Sciences* 179 (12) (2009) 1944–1959.
- [35] S. Xu, Y. Rahmat-Samii, Boundary conditions in particle swarm optimization revisited, *IEEE Transactions on Antennas and Propagation* 55 (3) (2007) 760–765.
- [36] W. Zhang, X. Xie, D. Bi, Handling boundary constraints for numerical optimization by particle swarm flying in periodic search space, in: *Proceedings of Congress on Evolutionary Computation*, 2004, pp. 2307–2311.
- [37] Y. Zhao, W. Zu, H. Zeng, A modified particle swarm optimization via particle visual modeling analysis, *Computers and Mathematics with Applications* 57 (2009) 2022–2029.