Research Article

Multiswarm Particle Swarm Optimization with Transfer of the Best Particle

Xiao-peng Wei,¹ Jian-xia Zhang,¹ Dong-sheng Zhou,² and Qiang Zhang²

¹School of Mechanical Engineering, Dalian University of Technology, Dalian 116024, China

²Key Laboratory of Advanced Design and Intelligent Computing, Ministry of Education, Dalian University, Dalian 116622, China

Correspondence should be addressed to Xiao-peng Wei; xpwei@dlu.edu.cn, Dong-sheng Zhou; zhoudongsheng@dlu.edu.cn, and Qiang Zhang; zhangq@dlu.edu.cn

Received 2 March 2015; Revised 23 June 2015; Accepted 12 July 2015

Academic Editor: Reinoud Maex

Copyright © 2015 Xiao-peng Wei et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

We propose an improved algorithm, for a multiswarm particle swarm optimization with transfer of the best particle called BMPSO. In the proposed algorithm, we introduce parasitism into the standard particle swarm algorithm (PSO) in order to balance exploration and exploitation, as well as enhancing the capacity for global search to solve nonlinear optimization problems. First, the best particle guides other particles to prevent them from being trapped by local optima. We provide a detailed description of BMPSO. We also present a diversity analysis of the proposed BMPSO, which is explained based on the Sphere function. Finally, we tested the performance of the proposed algorithm with six standard test functions and an engineering problem. Compared with some other algorithms, the results showed that the proposed BMPSO performed better when applied to the test functions and the engineering problem. Furthermore, the proposed BMPSO can be applied to other nonlinear optimization problems.

1. Introduction

Many nonlinear optimization problems are attracting increasing attention from researchers, with conflicting objectives and using various random search methods. Global optimization algorithms are employed widely to solve these problems [1]. Particle swarm optimization (PSO) is a type of random optimization method, which was inspired by the flocking behavior of birds [2, 3]. Kennedy and Eberhart were the first to propose PSO [4]. Compared with other swarm intelligence algorithms, PSO has a simple structure and rapid convergence rate, and it is easy to perform, which makes it an effective method for solving nonlinear optimization problems [5, 6].

In recent years, many researchers have tried to improve PSO to overcome its shortcomings; that is, it exhibits premature convergence and is readily trapped by local optima [7]. In order to improve the efficiency and effectiveness of multiobjective particle swarm optimization, a competitive and cooperative coevolutionary multiobjective particle swarm optimization algorithm (CCPSO) was presented by Goh et al. in 2010 [8]. A competitive and cooperative coevolution

mechanism was introduced in the proposed CCPSO, which does not handle the ZDT4 problem well, so it cannot be applied widely. Rathi and Vijay presented a modified PSO (EPSO) in 2010 [9], where two stages are used to balance the local and global search. First, the bandwidth of microstrip antenna (MSA) was modeled by a benchmark function. Second, the first output was then employed as an input to obtain the new output in the form of five parameters. The proposed EPSO is efficient and accurate. In 2011, a multiswarm self-adaptive and cooperative particle swarm optimization (MSCPSO) was proposed by Zhang and Ding [10], which employs four subswarms: subswarms 1 and 2 are basic, but subswarm 3 is influenced by subswarms 1 and 2, while subswarm 4 is affected by subswarms 1, 2, and 3. The four subswarms employ a cooperative strategy. While it achieved good performances in solving complex multimodal functions, MSCPSO was not applied to practical engineering problems. A new chaos-enhanced accelerated PSO algorithm was proposed by Gandomi et al. in 2013 [11], which delivered good performances when applied to a complex problem, but it is not easy to operate. Ding et al. developed the multiswarm cooperative chaos particle swarm optimization

algorithm in 2013 [12], which includes chaos and multiswarm cooperative strategies. This method was proposed only to optimize the parameters of a least squares support vector machine. In order to establish and optimize the alternative path, an efficient routing recovery protocol with endocrine cooperative particle swarm optimization was proposed by Hu et al. in 2015 [13], which employs a multiswarm evolution equation. Qin et al. proposed a novel coevolutionary particle swarm optimizer with parasitic behavior in 2015 [14], where a host swarm and parasite swarm exchange information. This method performs better in terms of the solution accuracy and convergence, but the structure is complex.

In the present study, we introduce a multiswarm particle swarm optimization with transfer of the best particle (BMPSO), in order to improve the global search capacity and to avoid trapping in local optima. The proposed algorithm employs three slave swarms and a master swarm. The best particle and worst particle are selected by the PSO from every slave swarm. The best particle is then transferred to the next slave swarm to replace the worst particle. The three best particles are then stored in the master swarm. Finally, the optimal value is obtained by PSO from the master swarm. Compared with other optimization algorithms, this parasitism strategy is easier to understand. The control parameters for the proposed algorithm do not increase and it can find the optimal solution easily.

The remainder of this paper is organized as follows. In Section 2, we provide an overview of the standard PSO. Our proposed BMPSO is explained in Section 3. We present the results of our numerical experimental simulations as well as comparisons in Section 4. In Section 5, we apply our proposed BMPSO to a practical engineering problem. Finally, we give our conclusions.

2. Standard PSO

In 1995, the standard PSO algorithm was presented by Kennedy and Eberhart [4]. It was developed further subsequently due to its easy implementation, high precision, and fast convergence. Similar to other evolutionary algorithms, it starts from random solutions and performs iterative searches to find the best solution. The quality of the solution is evaluated based on fitness.

PSO utilizes a swarm population of no quality and volume particles. Each particle moves in a D -dimensional space according to its own experience and that of neighbors while searching for the best solution. The position of the i 'th particle is expressed by the following vector: $x_i = [x_{i1}, \dots, x_{id}]$. The velocity is expressed by the following vector: $v_i = [v_{i1}, \dots, v_{id}]$. In each step, the particles move according to the following formulae:

$$v_{ij}(t+1) = v_{ij}(t) + c_1 r_1 [p_{ij} - x_{ij}(t)] + c_2 r_2 [p_{gj} - x_{ij}(t)], \quad (1)$$

$$x_{ij}(t+1) = x_{ij}(t) + v_{ij}(t+1), \quad (2)$$

where $j = 1, \dots, d$, c_1 and c_2 are acceleration constants, r_1 and r_2 represent two independent uniformly distributed random variables between 0 and 1, p_{ij} is the best previous position of the particle itself, and p_{gj} is the best global value.

In 1998, an inertia weight w was introduced into formula (1) by Shi and Eberhart [15], as shown by the following formula [18]:

$$v_{ij}(t+1) = wv_{ij}(t) + c_1 r_1 [p_{ij} - x_{ij}(t)] + c_2 r_2 [p_{gj} - x_{ij}(t)]. \quad (3)$$

It was demonstrated that the inclusion of a suitable inertia weight w allows the best solution to be searched more accurately. This weight can balance exploration and exploitation.

The standard PSO procedure is illustrated as follows.

Step 1. Initialize the position and velocity of each particle randomly in the population.

Step 2. Evaluate the value of the fitness function. Store the values in pbest and gbest.

Step 3. Update the position and velocity of each particle using formulae (2) and (3).

Step 4. Compare pbest and gbest, and then update gbest.

Step 5. Assess whether the conditions are met or not. If not, go back to Step 3.

3. General Description of BMPSO

The standard PSO is readily trapped by local optima and it is susceptible to premature convergence [19], while solving multiobjective problems with constraint conditions. Thus, we propose a new method for multiswarm PSO with transfer of the best particle called BMPSO. Our proposed algorithm can be applied to unconstrained problems but also with constraint conditions. It has the ability to escape local optima and prevent premature convergence.

3.1. Best Particle Coevolutionary Mechanism. In this section, we provide a general description of BMPSO. The proposed method employs three slave swarms and a master swarm. There is a specific relationship among the four swarms. We propose three swarms as slaves, which are designated slave-1, slave-2, and slave-3. First, the best particle-1 of slave-1 is selected, which is then stored in the master swarm. Second, the worst particle-2 from slave-2 is replaced with the best particle-1. The best particle-2 is then found, which is stored in the same manner as particle-1. The same strategy is applied to slave-3. When the three slave swarms have been processed, the master best particle is found in the master swarm as the optimal value. This strategy is not complete until various conditions have been met. The proposed BMPSO involves cooperation and competition. This evolutionary strategy is called parasitism in nature [20]. The structure of the BMPSO is shown in Figure 1.

FIGURE 1: Structure of BMPSO.

Our improved algorithm comprises three slave swarms and a master swarm, where a parasitism strategy is used to balance exploration and exploitation. The control parameters for the proposed BMPSO include the number of particles, inertia weight, dimension of particles, acceleration coefficients, and iterations. The number of particles is determined by the complexity of the problem, that is, from 5 to 100. The inertia weight decides how to inherit from current velocity. The dimension of the particles is determined by the optimization problem as the dimension of the result required. The acceleration coefficients give the particles the capacity for self-summary and learning from others, and they usually have a value of 2. The number of iterations can be determined by the experimental requirements. The pseudocodes for the BMPSO are presented as follows.

Algorithm BMPSO

Begin

Specify the population of each slave swarm
 Initialize the velocity and position
 Evaluate the fitness value
 Find the best particle and worst particle in each slave swarm
 Use the best particle to replace the worst particle in the next slave swarm
 Store the best particle in the master swarm
 Find the optimal value in the master swarm

Repeat

Until a terminate-condition is met

End

3.2. Diversity Analysis of BMPSO. In order to explain the proposed algorithm in detail, we illustrate the search capacity of each particle in the four swarms. The worst particle is replaced by the best particle. The best particle will lead the other particles away from a local optimum. Figure 2 shows

the evolutionary processes for the particles in slave-1, slave-2, slave-3, and the master swarm.

The evolutionary processes based on the Sphere function in 20D are shown in Figure 2. The graphs represent the results obtained by the proposed algorithm in a single run. Figure 2(a) shows that the particles in the four swarms performed their search behavior in a smooth manner. The diversity was improved when the best particle replaced the worst particle. This point is illustrated clearly by the experiments with the test function. Figure 2(b) shows the status of the particles during different generations based on the distances among four particles. The proposed algorithm made a greater effort to avoid becoming trapped by a local optimum after the 50th generation, while still considering the convergence speed. Thus, it maintained the diversity of the fitness value but not at the cost of the convergence speed.

The number of particles affects the optimization ability of BMPSO. In order to verify this, Figure 3 illustrates the convergence characteristics using the Sphere function as in the example ($N = 10, 20, 30, 50, 80$).

Figure 3 shows that the final optimized fitness value tended to improve as the number of particles increased. However, this is not as obvious in the later stage. In addition, there must be a greater communication cost when the number of particles increases. In the proposed BMPSO, $N = 10-30$ is sufficient for most problems.

In the standard PSO, the inertia weight w is very important, because it affects the balance between local and global search. It describes the influence of inertia on speed. When the inertia weight w is higher, the global optimization ability is better. In order to determine the appropriate value for w , we performed experiments with the test function and Figure 4 shows the results obtained with different values for w .

Figure 4 clearly demonstrates that the optimum fitness value is not easy to reach, when the inertia weight w is too small or too large. In the proposed BMPSO, $w = 0.5$ is the optimum value.

To a certain extent, the dimension (D) of particles represents the complexity of a problem, where the search capacity decreases as D increases. We performed experiments

FIGURE 2: Evolutionary processes based on the Sphere function.

FIGURE 3: Convergence characteristics for the Sphere function.

FIGURE 5: Results obtained using different dimensions (D).

FIGURE 4: Different inertia weights.

to determine the suitable scope for D using the Sphere function as an example, and Figure 5 illustrates the results obtained.

Figure 5 shows that the proposed BMPSO is more effective with a smaller dimension. In the proposed algorithm, $D = 5-30$ is a suitable dimension.

In this section, we considered the influence of different parameters on the proposed algorithm. This diversity analysis demonstrated the effectiveness of BMPSO, where the best particle shares more information with others and it replaces the worst particle during the evolutionary process. This

guides the other particles to prevent them from being trapped by local optima and avoids premature convergence. The proposed algorithm balances exploration and exploitation in an effective manner.

4. Numerical Experiments

In order to determine whether the proposed algorithm is effective for nonlinear optimization problems, we performed experiments using standard test functions [21, 22]. The proposed algorithm was simulated and verified on the MATLAB platform. The results were compared with those obtained using a modified particle swarm optimizer called W-PSO [15], an improved particle swarm optimization combined with chaos called CPSO [16], a completely derandomized self-adaptation in evolution strategies called CMAES [17], and a multiswarm self-adaptive and cooperative particle swarm optimization called MSCPSO [10].

4.1. Standard Test Functions. In order to validate the efficiency of BMPSO, the six standard test functions were employed in experiments to search for the optimum value of the fitness function. The six test functions were Sphere, Rastrigin, Griewank, Schwefel, Elliptic, and Rosenbrock [23]. The global optima for these six standard test functions are equal to zero. The formulae for the six functions are shown in Table 1.

Sphere, Schwefel, and Elliptic are typical unimodal functions, and thus it is relatively easy to search for the optimum value, which is considered to be the simple single mode state. Rastrigin and Griewank are typical nonlinear multimodal functions, with a wide search space. The peak shape emerges in high and low volatile hops, and it is usually considered difficult to handle complex multimodal problems using optimization algorithms. The global optimum value

TABLE 1: Benchmark functions.

	Test function	Search range	$f_{\min}$
Sphere	$f_1(x) = \sum_{i=1}^D x_i^2$	$[-100, 100]^D$	0
Rastrigin	$f_2(x) = \sum_{i=1}^D [x_i^2 - 10 \cos(2\pi x_i) + 10]$	$[-10, 10]^D$	0
Griewank	$f_3(x) = \frac{1}{4000} \sum_{i=1}^D x_i^2 - \prod_{i=1}^D \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	$[-50, 50]^D$	0
Schwefel	$f_4(x) = \sum_{i=1}^D x_i + \prod_{i=1}^D x_i $	$[-10, 10]^D$	0
Elliptic	$f_5(x) = \sum_{i=1}^D 10^{6(i-1)/(D-1)} x_i^2$	$[-100, 100]^D$	0
Rosenbrock	$f_6(x) = \sum_{i=1}^{D-1} [100(x_i^2 - x_{i+1})^2 + (x_i - 1)^2]$	$[-100, 100]^D$	0

for Rosenbrock is in a smooth and narrow parabolic valley. Thus, it is difficult to search for the optimum value because the function supplies little information for optimization algorithms. The six standard test functions comprise a class of complex nonlinear problems [24].

4.2. Results and Comparative Study. In order to compare the results in a standard manner, all of the experiments were performed on the same computer and using MATLAB R2013b. The parameters for the algorithms included the number of particles N , inertia weight w , dimension of particles D , acceleration coefficients c_1 and c_2 , and iterations M , which were set as follows. For W-PSO and the proposed BMPSO, the parameters were set as $N = 30$, $c_1 = c_2 = 2$, $w = 0.5$, $M = 1000$, and $D = 10, 30$. For CPSO, the parameters were set as $N = 30$, $c_1 = c_2 = 2$, $w_{\max} = 0.9$, $w_{\min} = 0.4$, $M = 1000$, $x_{\max} = 10^* \text{ones}(1, D)$, $x_{\min} = -10^* \text{ones}(1, D)$, and $D = 10, 30$. For CMAES, the parameters were set as described in [16]. For MSCPSO, the parameters were $N = 30$, $c_1 = c_2 = 2$, $w_{\max} = 0.9$, $w_{\min} = 0.4$, $M = 1000$, $D = 10, 30$, $a_1 = 1/6$, $a_2 = 1/3$, and $a_3 = 1/2$. In this study, all of the experiments using the algorithms were replicated independently 50 times. The best, worst, mean, and standard deviations of the fitness values were recorded, so the results summarized the performance of the algorithms [25]. The simulation results are presented in Table 2 ($D = 10$) and Table 3 ($D = 30$).

In Tables 2 and 3, the best results are highlighted in bold. Table 2 shows that the proposed BMPSO performed better compared with W-PSO, CPSO, CMAES, and MSCPSO for Rosenbrock on “Std.” The results obtained by BMPSO were closest to the theoretical value. It should be noted that the optimal value of 0 could be found for Rosenbrock. Table 3 shows that our proposed BMPSO performed better for Sphere, Griewank, Schwefel, Elliptic, and Rosenbrock compared with the other algorithms, whereas W-PSO performed the best for Rastrigin. According to this analysis, we conclude that the proposed BMPSO has a greater capacity for

FIGURE 6: A 3D model of a gearbox.

handling most nonlinear optimization problems compared with W-PSO, CPSO, CMAES, and MSCPSO.

5. Engineering Application

We also solved an engineering problem based on lightweight optimization design for a gearbox to confirm the efficiency of the proposed BMPSO. In the development of the autoindustry, lightweight optimization design for gearboxes is attracting much attention because of energy and safety issues. There are three standard types of lightweight methods [26, 27]. In order to solve the problem effectively, Yang et al. proposed a method that establishes a simplified model first, before analyzing it using ANSYS and then establishing an approximate model, with the response surface methodology, followed by final optimization with a genetic algorithm (GA) [28]. A simplified 3D model of a gearbox is shown in Figure 6. This simplified principle has no effect on the finite element analysis.

TABLE 2: Results for 10D problems.

Test functions	MSCPSO [10]	W-PSO [15]	CPSO [16]	CMAES [17]	BMPSO Present
f_1					
Best	$1.628e^{-001}$	$1.205e^{-005}$	$5.260e^{-002}$	$2.430e^{-002}$	$1.115e^{-132}$
Worst	$3.301e^{-000}$	$3.589e^{-002}$	$1.983e^{-000}$	$6.471e^{-001}$	$1.967e^{-121}$
Mean	$1.741e^{-000}$	$4.048e^{-003}$	$5.484e^{-001}$	$1.555e^{-001}$	$2.383e^{-122}$
Std.	$6.880e^{-001}$	$7.038e^{-003}$	$4.282e^{-001}$	$1.461e^{-001}$	$6.449e^{-122}$
f_2					
Best	$3.263e^{+001}$	$2.011e^{-000}$	$8.603e^{-000}$	$4.358e^{+001}$	$1.990e^{-000}$
Worst	$6.669e^{+001}$	$1.696e^{+001}$	$4.498e^{+001}$	$1.212e^{+002}$	$9.950e^{-000}$
Mean	$5.471e^{+001}$	$7.750e^{-000}$	$2.432e^{+001}$	$8.182e^{+001}$	$4.759e^{-000}$
Std.	$7.737e^{-000}$	$3.677e^{-000}$	$7.737e^{-000}$	$1.596e^{+001}$	$1.894e^{-000}$
f_3					
Best	$6.319e^{-002}$	$1.008e^{-006}$	$2.900e^{-003}$	$4.324e^{-000}$	$9.807e^{-008}$
Worst	$3.469e^{-001}$	$2.046e^{-003}$	$2.156e^{-001}$	$1.557e^{+001}$	$2.314e^{-004}$
Mean	$1.945e^{-001}$	$4.139e^{-004}$	$6.938e^{-002}$	$8.689e^{-000}$	$2.985e^{-005}$
Std.	$7.117e^{-002}$	$4.641e^{-004}$	$5.029e^{-002}$	$2.209e^{-000}$	$4.811e^{-005}$
f_4					
Best	$2.331e^{-000}$	$7.700e^{-003}$	$3.759e^{-001}$	$5.834e^{-000}$	$8.807e^{-078}$
Worst	$4.909e^{-000}$	$5.131e^{-001}$	$3.616e^{-000}$	$1.992e^{+014}$	$5.982e^{-070}$
Mean	$3.749e^{-000}$	$1.415e^{-001}$	$1.650e^{-000}$	$1.399e^{+000}$	$2.574e^{-071}$
Std.	$5.918e^{-001}$	$1.053e^{-001}$	$6.698e^{-001}$	$4.010e^{+013}$	$9.904e^{-071}$
f_5					
Best	$2.746e^{-000}$	$1.251e^{-000}$	$5.386e^{+001}$	$5.734e^{+005}$	$3.412e^{-133}$
Worst	$9.176e^{+003}$	$3.357e^{+002}$	$1.877e^{+004}$	$1.267e^{+007}$	$2.431e^{-117}$
Mean	$1.340e^{+003}$	$6.198e^{+001}$	$3.090e^{+003}$	$4.678e^{+006}$	$1.030e^{-118}$
Std.	$1.652e^{+003}$	$7.274e^{+001}$	$4.007e^{+003}$	$3.376e^{+006}$	$4.393e^{-118}$
f_6					
Best	$7.326e^{-000}$	$5.662e^{-000}$	$1.184e^{+001}$	$1.785e^{+001}$	0
Worst	$6.411e^{+001}$	$1.274e^{+001}$	$1.620e^{+002}$	$4.574e^{+005}$	$8.595e^{-000}$
Mean	$1.783e^{+001}$	$8.486e^{-000}$	$6.249e^{+001}$	$1.993e^{+004}$	$5.465e^{-001}$
Std.	$1.319e^{+001}$	$1.206e^{-000}$	$3.946e^{+001}$	$6.735e^{+004}$	$1.704e^{-000}$

In the optimization process, the variables are the bottom thickness x_1 , axial thickness x_2 , and lateral thickness x_3 . Detailed descriptions of the establishment of the fitness function can be found in [28]. The optimization problem for the lightweight optimization design of a gearbox can be represented by the following formula:

$$\begin{aligned}
& \min f(x_1, x_2, x_3), \\
& f(x_1, x_2, x_3) = 4.6896 + 3.3676 * x_1 + 0.5282 * x_2 \\
& \quad + 1.0110 * x_3, \\
& 87.2571 + 24.4741 * x_1 - 1.6680 * x_2 + 5.1004 \\
& \quad * x_3 > 120, \\
& 0.1715 - 0.0121 * x_1 - 0.0011 * x_2 - 0.0026 \\
& \quad * x_3 < 0.09, \\
& 73.1417 - 3.7565 * x_1 + 0.0754 * x_2 - 1.0626
\end{aligned}$$

$$* x_3 < 200,$$

$$x_1 \in [3, 6], x_2 \in [14, 20], x_3 \in [3, 8].$$

(4)

For the proposed BMPSO, the parameters were set as follows: $N = 30$, $c_1 = c_2 = 2$, $w = 0.5$, $M = 1000$, and $D = 3$. The parameters for the GA were set as described in [28]. This experiment was performed with 50 independent runs. The results obtained were expressed as the average of 50 runs.

The optimization results presented in Table 3 show that the weight of the gearbox is 31.3458 kg according to the proposed BMPSO, which is 8.4682 kg less than the original and 0.6368 kg less compared with GA. Thus, it can be concluded that the proposed BMPSO performed better compared with the GA. Therefore, the proposed BMPSO is effective in handling complex problems (see Table 4).

TABLE 3: Results for 30D problems.

Test functions	MSCPSO [10]	W-PSO [15]	CPSO [16]	CMAES [17]	BMPSO Present
f_1					
Best	$8.344e^{-000}$	$2.803e^{-001}$	$2.582e^{-000}$	$1.643e^{+004}$	$1.490e^{-050}$
Worst	$1.766e^{+001}$	$1.486e^{-000}$	$1.481e^{+001}$	$3.302e^{+004}$	$4.029e^{-042}$
Mean	$1.377e^{+001}$	$8.158e^{-001}$	$9.139e^{+000}$	$2.448e^{+004}$	$1.279e^{-043}$
Std.	$1.904e^{-000}$	$2.639e^{-001}$	$2.710e^{-000}$	$4.240e^{+003}$	$5.823e^{-043}$
f_2					
Best	$1.888e^{+002}$	$1.354e^{-000}$	$1.092e^{+002}$	$1.467e^{+004}$	$5.474e^{-000}$
Worst	$2.543e^{+002}$	$2.289e^{+001}$	$2.293e^{+002}$	$3.989e^{+004}$	$5.234e^{+001}$
Mean	$2.292e^{+002}$	$8.902e^{-000}$	$1.595e^{+002}$	$2.575e^{+004}$	$2.219e^{+001}$
Std.	$1.442e^{+001}$	$4.894e^{-000}$	$2.714e^{+001}$	$4.326e^{+003}$	$8.802e^{-000}$
f_3					
Best	$4.156e^{-001}$	$1.200e^{-002}$	$9.400e^{-002}$	$2.881e^{+001}$	$3.600e^{-003}$
Worst	$8.146e^{-001}$	$8.402e^{-002}$	$7.377e^{-001}$	$4.071e^{+001}$	$1.715e^{-002}$
Mean	$6.618e^{-001}$	$3.750e^{-002}$	$4.080e^{-001}$	$3.416e^{+001}$	$8.690e^{-003}$
Std.	$9.181e^{-002}$	$1.379e^{-002}$	$1.088e^{-001}$	$2.636e^{-000}$	$2.938e^{-003}$
f_4					
Best	$1.225e^{+001}$	$1.798e^{-000}$	$8.544e^{-000}$	$2.790e^{+032}$	$9.794e^{-034}$
Worst	$1.826e^{+001}$	$4.753e^{-000}$	$1.595e^{+001}$	$4.676e^{+045}$	$2.946e^{-029}$
Mean	$1.626e^{+001}$	$3.334e^{-000}$	$1.243e^{+001}$	$2.789e^{+044}$	$3.206e^{-030}$
Std.	$1.114e^{-000}$	$6.914e^{-001}$	$1.680e^{-000}$	$8.220e^{+044}$	$7.794e^{-030}$
f_5					
Best	$1.584e^{+004}$	$1.892e^{+003}$	$1.649e^{+004}$	$1.875e^{+008}$	$1.657e^{-048}$
Worst	$4.792e^{+005}$	$1.983e^{+004}$	$2.789e^{+005}$	$1.363e^{+009}$	$4.245e^{-040}$
Mean	$2.103e^{+005}$	$6.978e^{+003}$	$1.269e^{+005}$	$7.935e^{+008}$	$2.546e^{-041}$
Std.	$1.185e^{+005}$	$3.620e^{+003}$	$6.848e^{+004}$	$2.416e^{+008}$	$8.401e^{-041}$
f_6					
Best	$1.944e^{+002}$	$5.434e^{+001}$	$4.955e^{+002}$	$2.323e^{+009}$	$2.632e^{+001}$
Worst	$2.259e^{+003}$	$2.054e^{+002}$	$2.649e^{+003}$	$1.057e^{+010}$	$3.016e^{+001}$
Mean	$6.547e^{+002}$	$1.008e^{+002}$	$1.586e^{+003}$	$5.284e^{+009}$	$2.907e^{+001}$
Std.	$3.655e^{+002}$	$2.963e^{+001}$	$5.188e^{+002}$	$1.723e^{+009}$	$7.100e^{-001}$

TABLE 4: Optimization results.

Method	X_1 (mm)	X_2 (mm)	X_3 (mm)	M (kg)
Original	4	20	7	39.8140
GA	4.8182	14.0024	3.0029	31.9826
BMPSO	4.8185	14.0001	3.0017	31.3458

6. Conclusions

In this study, we proposed an improved multiswarm PSO method with transfer of the best particle called BMPSO, which utilizes three slave swarms and a master swarm that share information among themselves via the best particle. All of the particles in the slave swarms search for the global optimum using the standard PSO. The best particle replaces the worst in order to guide other particles to prevent them from becoming trapped by local optima. We performed a diversity analysis of BMPSO using the Sphere function.

We also applied the proposed algorithm to standard test functions and an engineering problem.

We introduced parasitism into the standard PSO to develop a multiswarm PSO that balances exploration and exploitation. The advantages of the proposed BMPSO are that it is easy to understand, has a low number of parameters, and is simple to operate. In the proposed algorithm, the strategy of using the best particle to replace the worst enhances the global search capacity when solving nonlinear optimization problems. The optimum solution is obtained from the master swarm. The diversity analysis also demonstrated the obvious improvements obtained using the proposed algorithm. Compared with previously proposed algorithms, our BMPSO delivered better performance. The result obtained by the BMPSO in an engineering problem demonstrated its efficiency in handling a complex problem.

In further research, convergence analysis of the method should be performed in detail. Our further research will also focus on extensive evaluations of the proposed BMPSO by

solving more complex and discrete practical optimization problems.

Conflict of Interests

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

Acknowledgment

This work is supported by the National Natural Science Foundation of China (Grant no. 61425002).

References

- [1] X.-S. Yang, *Engineering Optimization: An Introduction with Metaheuristic Applications*, vol. 7, John Wiley & Sons, 2010.
- [2] X. Chu, M. Hu, T. Wu, J. D. Weir, and Q. Lu, "AHPS²: an optimizer using adaptive heterogeneous particle swarms," *Information Sciences*, vol. 280, pp. 26–52, 2014.
- [3] S. Saini, N. Zakaria, D. R. A. Rambli, and S. Sulaiman, "Markerless human motion tracking using hierarchical multi-swarm cooperative particle swarm optimization," *PLoS ONE*, vol. 10, no. 5, Article ID e0127833, 2015.
- [4] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of the IEEE International Conference on Neural Networks*, vol. 4, pp. 1942–1948, IEEE, Perth, Australia, November–December 1995.
- [5] E. Talbi, *Metaheuristics, From Design to Implementation*, vol. 6, John Wiley & Sons, 2009.
- [6] Y. Wang, B. Li, T. Weise, J. Wang, B. Yuan, and Q. Tian, "Self-adaptive learning based particle swarm optimization," *Information Sciences*, vol. 181, no. 20, pp. 4515–4538, 2011.
- [7] Q. Qin, S. Cheng, Q. Zhang, L. Li, and Y. Shi, "Biomimicry of parasitic behavior in a coevolutionary particle swarm optimization algorithm for global optimization," *Applied Soft Computing*, vol. 32, no. 7, pp. 224–240, 2015.
- [8] C. K. Goh, K. C. Tan, D. S. Liu, and S. C. Chiam, "A competitive and cooperative co-evolutionary approach to multi-objective particle swarm optimization algorithm design," *European Journal of Operational Research*, vol. 202, no. 1, pp. 42–54, 2010.
- [9] A. Rathi and R. Vijay, "Expedite particle swarm optimization algorithm (EPSO) for optimization of MSA," *Swarm, Evolutionary, and Memetic Computing*, vol. 6466, pp. 163–170, 2010.
- [10] J. Zhang and X. Ding, "A multi-swarm self-adaptive and cooperative particle swarm optimization," *Engineering Applications of Artificial Intelligence*, vol. 24, no. 6, pp. 958–967, 2011.
- [11] A. H. Gandomi, G. J. Yun, X.-S. Yang, and S. Talatahari, "Chaos-enhanced accelerated particle swarm optimization," *Communications in Nonlinear Science and Numerical Simulation*, vol. 18, no. 2, pp. 327–340, 2013.
- [12] G. Ding, L. Wang, P. Yang, P. Shen, and S. Dang, "Diagnosis model based on least squares support vector machine optimized by multi-swarm cooperative chaos particle swarm optimization and its application," *Journal of Computers*, vol. 8, no. 4, pp. 975–982, 2013.
- [13] Y.-F. Hu, Y.-S. Ding, L.-H. Ren, K.-R. Hao, and H. Han, "An endocrine cooperative particle swarm optimization algorithm for routing recovery problem of wireless sensor networks with multiple mobile sinks," *Information Sciences*, vol. 300, pp. 100–113, 2015.
- [14] Q. Qin, S. Cheng, Q. Zhang, L. Li, and Y. Shi, "Biomimicry of parasitic behavior in a coevolutionary particle swarm optimization algorithm for global optimization," *Applied Soft Computing*, vol. 32, pp. 224–240, 2015.
- [15] Y. H. Shi and R. Eberhart, "A modified particle swarm optimizer," in *Proceedings of the IEEE International Conference on Evolutionary Computation*, pp. 69–73, IEEE, Anchorage, Alaska, USA, May 1998.
- [16] B. Liu, L. Wang, Y.-H. Jin, F. Tang, and D.-X. Huang, "Improved particle swarm optimization combined with chaos," *Chaos, Solitons and Fractals*, vol. 25, no. 5, pp. 1261–1271, 2005.
- [17] N. Hansen and A. Ostermeier, "Completely de-randomized self-adaptation in evolution strategies," *Evolutionary Computation*, vol. 9, no. 2, pp. 159–195, 2001.
- [18] Y. V. Pehlivanoglu, "A new particle swarm optimization method enhanced with a periodic mutation strategy and neural networks," *IEEE Transactions on Evolutionary Computation*, vol. 17, no. 3, pp. 436–452, 2013.
- [19] J. J. Liang, A. K. Qin, P. N. Suganthan, and S. Baskar, "Comprehensive learning particle swarm optimizer for global optimization of multimodal functions," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 3, pp. 281–295, 2006.
- [20] A. E. Douglas, *Symbiotic Interactions*, Oxford University Press, Oxford, UK, 1994.
- [21] X. Yao, Y. Liu, and G. Lin, "Evolutionary programming made faster," *IEEE Transactions on Evolutionary Computation*, vol. 3, no. 2, pp. 82–102, 1999.
- [22] S. C. Esquivel and C. A. C. Coello, "On the use of particle swarm optimization with multimodal functions," in *Proceedings of the Congress on Evolutionary Computation (CEC '03)*, pp. 1130–1136, IEEE, Canberra, Australia, December 2003.
- [23] K. T. Li, M. N. Omidvar, Z. Yang, and K. Qin, "Benchmark functions for the CEC' 2013 special session and competition on large-scale global optimization," in *Proceedings of the IEEE Congress on Evolutionary Computation (CEC '13)*, Cancun, Mexico, June 2013.
- [24] T. Xiang, X. Liao, and K.-W. Wong, "An improved particle swarm optimization algorithm combined with piecewise linear chaotic map," *Applied Mathematics and Computation*, vol. 190, no. 2, pp. 1637–1645, 2007.
- [25] G. J. Koehler, "Conditions that obviate the no-free-lunch theorems for optimization," *INFORMS Journal on Computing*, vol. 19, no. 2, pp. 273–279, 2007.
- [26] Y. Shi, P. Zhu, L. Shen, and Z. Lin, "Lightweight design of automotive front side rails with TWB concept," *Thin-Walled Structures*, vol. 45, no. 1, pp. 8–14, 2007.
- [27] Y. Zhang, X. M. Lai, P. Zhu, and W. R. Wang, "Lightweight design of automobile component using high strength steel based on dent resistance," *Materials and Design*, vol. 27, no. 1, pp. 64–68, 2006.
- [28] G. Yang, J. Zhang, Q. Zhang, and X. Wei, "Research on lightweight optimization design for gear box," in *Proceedings of the 7th International Conference on Intelligent Robotics and Applications*, pp. 576–585, Guangzhou, China, 2014.