Hindawi Publishing Corporation
Discrete Dynamics in Nature and Society
Volume 2012, Article ID 578064, 22 pages
doi:10.1155/2012/578064

Research Article

A Dynamic Multistage Hybrid Swarm Intelligence Optimization Algorithm for Function Optimization

Daqing Wu[1,2,3] and Jianguo Zheng[1]

[1] *Glorious Sun School of Business and Management, DongHua University, Shanghai 200051, China*
[2] *Computer Science and Technology Institute, University of South China, Hunan, Hengyang 421001, China*
[3] *Artificial Intelligence Key Laboratory of Sichuan Province, Sichuan University of Science and Engineering, Zigong 643000, China*

Correspondence should be addressed to Daqing Wu, dqw_1982@126.com

Received 24 April 2012; Revised 24 August 2012; Accepted 24 August 2012

Academic Editor: Gabriele Bonanno

A novel dynamic multistage hybrid swarm intelligence optimization algorithm is introduced, which is abbreviated as DM-PSO-ABC. The DM-PSO-ABC combined the exploration capabilities of the dynamic multiswarm particle swarm optimizer (PSO) and the stochastic exploitation of the cooperative artificial bee colony algorithm (CABC) for solving the function optimization. In the proposed hybrid algorithm, the whole process is divided into three stages. In the first stage, a dynamic multiswarm PSO is constructed to maintain the population diversity. In the second stage, the parallel, positive feedback of CABC was implemented in each small swarm. In the third stage, we make use of the particle swarm optimization global model, which has a faster convergence speed to enhance the global convergence in solving the whole problem. To verify the effectiveness and efficiency of the proposed hybrid algorithm, various scale benchmark problems are tested to demonstrate the potential of the proposed multistage hybrid swarm intelligence optimization algorithm. The results show that DM-PSO-ABC is better in the search precision, and convergence property and has strong ability to escape from the local suboptima when compared with several other peer algorithms.

1. Introduction

Optimization can be viewed as one of the major quantitative tools in network of decision making, in which decisions have to be taken to optimize one or more objectives in some prescribed sets of circumstances. Typical single objective bound constrained optimization problems can be expressed as

$$\text{Min } f(x), \quad x = [x_1, x_2, \ldots, x_D],\ x_i \in [x_{\min,} x_{\max}] \text{ with } i = 1, 2, \ldots, D, \tag{1.1}$$

where D is the number of variables (the dimension of the search space) and the $x_{\min}$ and $x_{\max}$ are the upper and lower bounds of the search space. In view of the practical utility of optimization problems, there is a need for efficient and robust computational algorithms, which can numerically solve in computers the mathematical models of medium as well as large size optimization problems arising in different fields. Evolutionary algorithms have emerged as a revolutionary approach for solving complex search and optimization problems. The success of most of the Heuristic optimization algorithms depends to a large extent on the careful balance of two conflicting goals, exploration (diversification) and exploitation (intensification). While exploration is important to ensure that every part of the solution domain is searched enough to provide a reliable estimate of the global optimum, exploitation, on the other hand, is important to concentrate the search effort around the best solutions found so far by searching their neighbourhoods to reach better solutions. The search algorithms achieve these two goals by using local search methods, global search approaches, or an integration of both global and local strategies: these algorithms are commonly known as hybrid methods. Hybrid algorithms are chosen as the topic of the present paper because they are a growing area of intelligent systems research, which aims to combine the desirable properties of different approaches to mitigate their individual weaknesses.

In the recent years, the hybridization technique based on particle swarm optimization (PSO) and artificial bee colony (ABC) was hotly researched. El-Abd [1] proposed a hybridization approach between ABC and SPSO. This is achieved by incorporating an ABC component into SPSO, which updates the pbest information of the particles in every iteration using the ABC update Equation; penalty guided support vector machines based on hybrid of particle swarm optimization and artificial bee colony algorithm to mine financial distress trend data [2]; Shi et al. [3] developed a hybrid swarm intelligent algorithm based on particle swarm optimization (PSO) and artificial bee colony (ABC). Turanoğlu and Özceylan [4] used particle swarm optimization and artificial bee colony to optimize single input-output fuzzy membership functions; the obtained results show that PSO and ABC methods are capable and effective to find optimal values of fuzzy membership functions in a reasonable time.

Honey bees are among the most closely studied social insets. Their foraging behaviour, learning, memorizing, and information sharing characteristics have recently been one of the most interesting research areas in swarm intelligence [5]. According to many researched results on the local variants of the PSO [6], PSO with small neighbourhoods performs better on complex problems. Ballerini pointed out that the interaction ruling animal collective behavior depends on topological rather than metric distance [7–9] and proposed a new model for self-organized dynamics and its flocking behaviour. In this paper, we propose a new optimization method, called dynamic multistage hybrid swarm intelligence optimization algorithm (DM-PSO-ABC).

The rest of the paper is organized as follows. Section 2 briefly introduces the original PSO algorithm and ABC algorithm and describes the related hybrid techniques of the particle swarm optimization (PSO) and artificial bee colony algorithm (ABC) in recent years. Section 3 discusses a new method named dynamic multistage DM-PSO-ABC algorithm. Section 4 tests the algorithms on the benchmarks, and the results obtained are presented and discussed. Finally, conclusions are given in Section 5.

2. The Original Algorithm

2.1. Particle Swarm Optimization (PSO)

Particle swarm optimization (PSO) is inspired by social behaviour simulation and was originally designed and developed by Kennedy and Eberhart [10]. It is a population-based search algorithm that was on the basis of the simulation of the social behaviour of birds within a flock. In the PSO, individuals are particles and are "flown" through hyperdimensional search space. They simulated birds' swarm behaviour and made each particle in the swarm move according to its experience and the best experience of particle. Each particle represents a potential solution to the problem and searches around in a multidimensional search space. All particles fly through the D-dimensional parameter space of the problem while learning from the historical information gathered during the search process. The particles have a tendency to fly towards better search regions over the course of search process. The velocity v_{id} and position x_{id} updates of the dth dimension of the ith particle are presented below:

$$v_{id}(t+1) = w \times v_d(t) + c_1 \times r_1 \times \left(pb_{id}(t) - x_{id}(t)\right) + c_2 \times r_2 \times \left(gb_d(t) - x_{id}(t)\right), \tag{2.1}$$

$$x_{id}(t+1) = x_{id}(t) + v_{id}(t), \tag{2.2}$$

where c_1 and c_2 are the acceleration constants and r_1 and r_2 are two uniformly distributed random numbers in $[0,1]$. $x_i = (x_{i1}, x_{i2}, \ldots, x_{id})$ is the position of the ith particle, $v_i = (v_{i1}, v_{i2}, \ldots, v_{id})$ represents the velocity of the ith particle, $\text{pbest}_i = (\text{pbest}_{i1}, \text{pbest}_{i2}, \ldots, \text{pbest}_{id})$ is the best previous position yielding the best fitness value for the ith particle, $gb = (gb_1, gb_2, \ldots, gb_D)$ is the best position discovered by the whole population, and w is the inertia weight used to balance between the global and local search abilities.

There are two main models of the PSO algorithm, called global model and local model. The two models differ in the way of defining the neighborhood for each particle. In the global model, the neighborhood of a particle consists of the particles in the whole swarm, which share information between each other. On the contrary, in the local model, the neighborhood of a particle is defined by several particles. The two models give different performances on different problems. van den Bergh and Engelbrecht [11] and Poli et al. [12] pointed out that the global model has a faster convergence speed but also has a higher probability of getting stuck in local optima than the local model. On the contrary, the local model is less vulnerable to the attraction of local optima but has a slower convergence speed than the global model. In order to give a standard form for PSO, Bratton and Kennedy proposed a standard version of PSO (SPSO) [13]. In SPSO a local ring population topology is used, and the experimental results have shown that the local model is more reliable than the global model on many test problems. The velocity update of the local PSO is

$$v_{id}(t+1) = w \times v_d(t) + c_1 \times r_1 \times \left(pb_{id}(t) - x_{id}(t)\right) + c_2 \times r_2 \times \left(lb_d(t) - x_{id}(t)\right). \tag{2.3}$$

The population topology has a significant effect on the performance of PSO. It determines the way particles communicate or share information with each other. Population topologies can be divided into static and dynamic topologies. For static topologies, communication structures of circles, wheels, stars, and randomly assigned edges were tested [14], showing that the performance of algorithms is different in different problems depending on the topology used. Then, Kennedy and Mendes [15] have tested a large number of aspects

of the social-network topology on five test functions. After that, a fully informed PSO (FIPS) algorithm was introduced by Mendes et al. [16]. In FIPS, a particle uses a stochastic average of pbests from all of its neighbors instead of using its own pbest position and the gbest position in the update equation. A recent study [17] showed that PSO algorithms with a ring topology are able to locate multiple global or local optima if a large enough population size is used.

For dynamic topologies, Suganthan [18] suggested a dynamically adjusted neighbor model, where the search begins with an lbest model and is gradually increased until the gbest model is reached. Janson and Middendorf [19] proposed a dynamic hierarchical PSO (HPSO) to define the neighborhood structure where particles move up or down the hierarchy depending on the quality of their pbest solutions. Liang et al. [20] developed a comprehensive learning PSO (CLPSO) for multimodal problems. In CLPSO, a particle uses different particles' historical best information to update its velocity, and for each dimension, a particle can potentially learn from a different exemplar.

2.2. Artificial Bee Colony Algorithm (ABC)

Recently, by simulating the behaviour of honey bee swarm intelligence, an efficient bee colony (ABC) algorithm is proposed [21, 22]. Due to its simplicity and ease of implementation, the ABC algorithm has gained more and more attention and has been used to solve many practical engineering problems. In the basic ABC algorithm [18–21], it classifies foraging artificial bees into three groups, namely, employed bees, onlookers, and scouts. An employed bee is responsible for flying to and making collections from the food source which the bee swarm is exploiting. An onlooker waits in the hive and decides on whether a food source is acceptable or not. This is done by watching the dances performed by the employed bees. A scout randomly searches for new food sources by means of some internal motivation or possible external clue. In the ABC algorithm, each solution to the problem under consideration is called a food source and represented by an n-dimensional real-valued vector where the fitness of the solution corresponds to the nectar amount of the associated food resource. As with other intelligent swarm-based approaches, the ABC algorithm is an iterative process. The approach begins with a population of randomly generated solutions (or food sources); then, the following steps are repeated until a termination criterion is met [23, 24].

In the employed bees phase, artificial employed bees search for new food sources having more nectar within the neighbourhoods of the food source in their memory. They find a neighbour food source as defined in (2.4), providing that its nectar is higher than that of the previous one; the bee memorizes the new position and forgets the old one. Then evaluate its fitness as defined in (2.5). After producing the new food source, its fitness is calculated, and a greedy selection is applied between it and its parent. After that, employed bees share their food source information with onlooker bees waiting in the hive by dancing on the dancing area:

$$v_{ij} = x_{ij} + \vartheta_{ij} \times \left(x_{ij} - x_{kj}\right), \tag{2.4}$$

where $k \in \{1, 2, \ldots, \mathrm{BN}\}$, BN is the number of food sources which is equal to the number of employed bees in each subgroup, and $j \in \{1, 2, \ldots, D\}$ are randomly chosen indexes. Although k is determined randomly, it has to be different from i. ϑ_{ij} is a random number between $[-1, 1]$. It controls the production of a neighbour food source position around x_{ij}

and the modification represents the comparison of the neighbour food positions visually by the bee. Equation (2.4) shows that as the difference between the parameters of the x_{ij} and x_{kj} decreases, the perturbation on the position x_{ij} decreases, too:

$$\text{Fitness}(i) = \begin{cases} \dfrac{1}{1+f(i)}, & \text{if } f(i) \geq 0, \\ 1 + \text{abs}\big(f(i)\big), & \text{otherwise.} \end{cases} \tag{2.5}$$

In the onlooker bees' phase, artificial onlooker bees probabilistically choose their food sources depending on the information provided by the employed bees as defined in (2.6) and (2.7). For this purpose, a fitness-based selection technique can be used, such as the roulette wheel selection method. After a food source for an onlooker bee is probabilistically chosen, a neighbourhood source is determined, and its fitness value is computed. As in the employed bees phase, a greedy selection is applied between two sources:

$$\text{prob}(i) = \frac{0.9 * \text{Fitness}(i)}{\max(\text{Fitness})} + 0.1, \tag{2.6}$$

$$\text{prob}(i) = \frac{\text{Fitness}(i)}{\sum_{n=1}^{SN} \text{Fitness}}. \tag{2.7}$$

In the scout bees' phase, employed bees whose solutions cannot be improved through a predetermined number of trials, called "limit", become scouts, and their solutions are abandoned. Then, the scouts start to search for new solutions, randomly using (2.8). Hence, those sources which are initially poor or have been made poor by exploitation are abandoned, and negative feedback behaviour arises to balance the positive feedback:

$$x_{ij} = x_{\min d} + \text{rand}\ (0,1) \times \left(x_{\max j} - x_{\min j}\right). \tag{2.8}$$

3. A Novel Multistage Hybrid Swarm Intelligence Optimization Algorithm

As mentioned in the previous sections, researchers confirm that the PSO algorithm should be taken into account as a powerful technique for handling various kinds of optimization problems, but it is vulnerable to premature convergence and low stability in the process of evolution. According to many researches that PSO with small neighborhoods performs better on complex problems. The particles will enhance their diversity with a randomized regrouping schedule by dynamically changing neighborhood structures. We allow maximum information exchange among the particles to enhance the diversity of the particles. Cooperative ABC algorithm has the mechanism of labor's division and cooperation, the different search strategies can cooperate together to achieve global optimization, and it has strong ability in global optimization, but when close to the global optimal solution, the search speed slowed, so the population diversity reduced and particles were apt to be trapped in local optimal solution. In order to make full use of and balance the exploration of the solution search equation of ABC and the exploitation of the proposed solution search equation of PSO, we propose a novel dynamic multistage hybrid DM-PSO-ABC based on the compensation by combining the evolution ideas of the PSO and ABC algorithm.

In the proposed hybrid DM-PSO-ABC algorithm, the different strategies in the three phases collaborate to cope with different situations in the search space. Firstly, we used local version of PSO with a new neighborhood topology of the small neighborhoods to maintain the population diversity; secondly, we adjusted the initial allocation of pheromone in the cooperative ABC algorithm based on a series of suboptimal solutions; obtained in the fore stage, we make use of these advantages of the parallel, positive feedback and high accuracy of solution of cooperation ABC to implement solving of the whole problem. In the third stage, we make use of the PSO global model that has a faster convergence speed to enhance the global convergence. In Pseudocode 1, the main steps of DM-PSO-ABC algorithm are given.

3.1. Rough Searching by the Multiswarm PSO with a Randomized Regrouping Schedule

In the first stage of the DM-PSO-ABC, small neighbourhoods are used. The population is divided into small-sized swarms. Each subswarm uses its own members to search for better regions in the search space. The small-sized swarms used their own best historical information in the searching phase; they can easily converge to a local optimum because of PSO's speedy convergence behaviour. Hence, a randomized regrouping schedule is introduced so that the particles will enhance their diversity by dynamically changing neighbourhoods' structures. For every L generation, the population is regrouped randomly and starts searching using a new configuration of small subswarm. Here L is called the regrouping period. In this way, the information obtained by each subswarm is exchanged among the whole swarms. Simultaneously the diversity of the population is also increased. In DM-PSO-ABC, in order to constrain a particle within the search range, the fitness value of a particle is calculated, and the corresponding pbest is updated only if the particle is within the search range. Since all pbests and lbests are within the search bounds, all particles will eventually return within the search bounds.

In Figure 1, we use three swarms with ten particles in each swarm to show the regrouping schedule. First, the thirty particles are divided into three swarms randomly. Then the three swarms use their own particles to search for better solutions. In this period, they may converge to near a local optimum. Then the whole population is regrouped into new swarms. The new swarms begin their search. This process is continued until a stop criterion is satisfied. With the randomly regrouping schedule, particles from different swarms are grouped in a new configuration so that each small swarms search space is enlarged, and better solutions are possible to be found by the new small swarms.

In order to achieve better results on complex problems, the dynamic multigroup particle swarm optimizer is designed to make the particles have a large diversity, and consequently the convergence speed will be slow. Even after the global region is found, the particles will not converge to the global optimization very fast in order to avoid premature convergence. How to maintain the diversity and get the good result at the same time is a problem. Hence, In order to alleviate this weakness and give a better search in the better local areas, an ABC local search is added into the dynamic multiswarm particle swarm optimizer. For every L generation, the pbests of ten randomly chosen particles will be used as the starting points of cooperation ABC local search. We calculate the fitness values of all the pbests for each refined solution and replace the nearest ones with the refined solutions if the refined solution is better.

A dynamic Multi-stage hybrid swarm intelligence optimization algorithm

m: Each swarm's population size
S: Swarms' number
L: Regrouping period
Max_gen: Max generations, stop criterion
Step 1 Generate initial $m * n$ particles and set up parameters for each particle;
Initialize the position of all particles $X = (X_1, X_2, \ldots, X_N)$, and their fitnesses, and the velocity of all particles $V = (V_1, V_2, \ldots, V_N)$; the best local position of all particles pbest = (pbest_1,pbest_2,...,pbest_N);
Step 2 Update all particles using local version PSO with Dynamic multi-group
For $i = 1 : 0.95 * \text{Max\_gen}$
Update each swarm using (2.2), (2.3) local version PSO, pbests and lbests updating
If $\text{mod}(i, L) == 0$
Regroup the swarms randomly
End
Step 3 local search carried out in each small swarm by the artificial bee colony
the population of food sources (solutions) is initialized by the current lbests in each sub-swarm
For each component $j \in (1, 2, \ldots, D)$
Employed Bees' Phase
For each employed bee i
Replace the j component of the lbest by using the j component of bee i
Calculate the [f_newlbest (lbest_1, lbest_2,...,x_{ij},...,lbest_D)]
If (f_newlbest_ better than f_lbest)
Then newlbest replaced lbest
For employed bee i produce new food source positions V_i by using (2.4)
Calculate the value fitness by using (2.5)
Apply greedy selection mechanism
End For.
End For
Calculate the probability values p_i for the solutions X_i by (2.6) and (2.7) using the roulette wheel selection rule;
Onlooker Bees' Phase
For each onlooker bee i
Chooses a food source depending on p_i
Replace the j component of the lbest by using the j component of bee i
Calculate the f[newlbest](lbest_1, lbest_2, ..., x_{ij}, ...,lbest_D)
If f(newlbest) better than f(lbest)
Then newlbest replaced lbest
For onlooker bee i produce new food source positions V_i by using (2.4)
Calculate the value fitness
Apply greedy selection mechanism
End For
End For
Scout Bees' Phase
If there is an employed bee becomes scout
Then replace it with a new random source positions by using (2.8)
Memorize the best solution achieved so far
Compare the best solution with lbest and Memorize the better one.
Step 4 Update all particles using global version PSO
For $i = 0.95 * \text{Max\_gen} : \text{Max\_gen}$
Update all particles using global version PSO, pbests and gbest updating
End

Pseudocode 1: Pseudocode of DM-PSO-ABC.

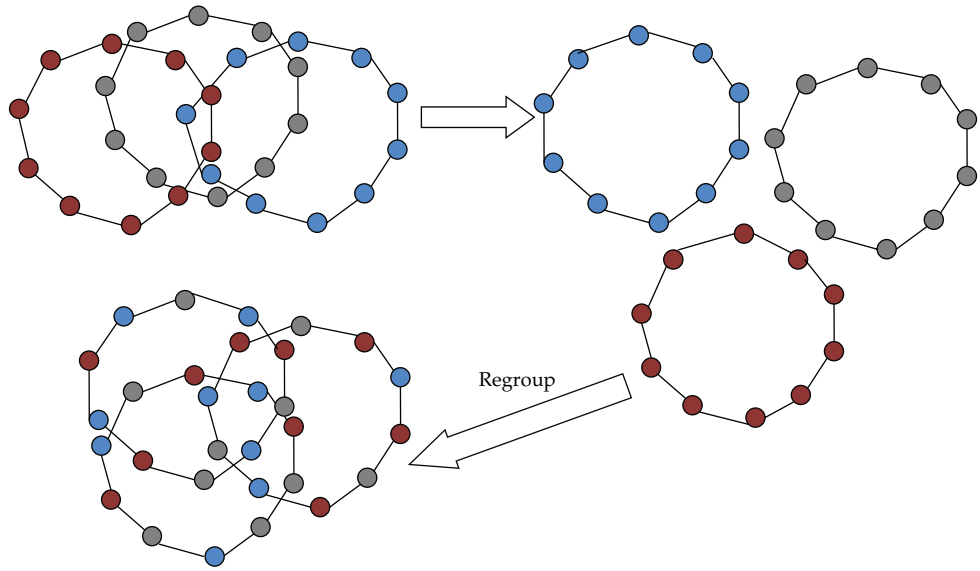

Figure 1: DM-PSO-ABC's configuration of small swarms.

3.2. *Detailed Searching in Each Small Swarm by the Cooperative Artificial Bee Colony*

In the second stage, we present an extended ABC algorithm, namely, the cooperative article bee colony, which significantly improves the original ABC in solving complex optimization problems. In the ABC algorithm, the goal of each individual bee is to produce the best solution. From expression (2.4), we can see that the new food source is produced by a random neighborhood of current food position and a random single dimension of D-dimensional vector. This will bring about a problem that an individual may have discovered a good dimension, but the fitness of the individual is computed by using D-dimensional vector; hence we know it is very probable that the individual is not the best solution in the end, and the good dimension which the individual has found will be abandoned. To produce a good solution vector, all the populations must cooperate. And the information from all the populations needs to be used. Therefore, we apply cooperative search to solve the problem in the ABC algorithm and propose the cooperative ABC algorithm. We set a super best solution vector, namely, lbest, and its each component of D-dimensional is the best in each subswarm. For lbest: (lb1, lb2$\cdots$lbd), lbi corresponds to the ith component of the lbest. In the initialization phase, we evaluate the fitness of the initial food source positions and set the position which has the best fitness as the initial lbest. In the employed bees' and onlooker bees' phase, we use the j component of each individual to replace the corresponding component of the lbest to find the best position of the j component. lbests do not influence the employed and onlooker bees finding new food sources. It is a virtual bee. It just saves the best one of each component. After all phases, the best solution achieved by all individuals and the lbest will be compared.

In this stage, the population of food sources (solutions) is initialized by each small swarm's lbest which is generated in the first stage called nectar information of the food

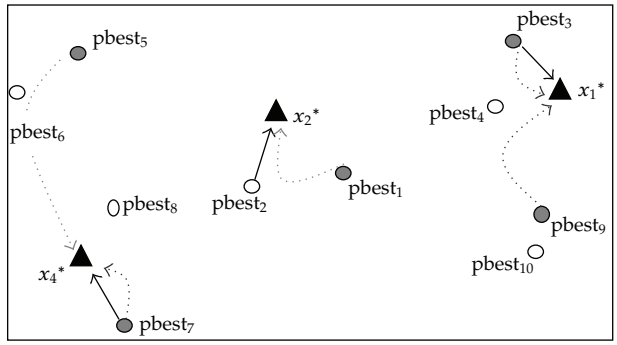

Figure 2: Illustration of local search phase for a population with 10 particles.

sources (solutions); the employed bee is to perform a neighbourhood search around a given food source. Therefore, the employed bee takes the exploitation search of the algorithm.

An illustration for the cooperative ABC local search phase for a swarm of 10 particles is given in Figure 2. Five pbests "•" pbest_1, pbest_3, pbest_5, pbest_7, and pbest_9 are randomly chosen as the start points for the local search, and 3 local optima x_1^*, x_2^*, and x_4^* are achieved after the local search. The nearest three pbests "◯" pbest_2, pbest_4, pbest_6, pbest_8, pbest_{10} are replaced by x_1^*, x_2^*, and x_4^* "▲", respectively, provided the refined solutions are better.

3.3. Rapid Convergence by the Global Version of PSO

The success of PSO in solving one specific problem crucially depends on the choice of suitable strategies; the particles can play different roles (exploitation and exploration) during the two-stage search progress before; in the third stage, we make use of the PSO global model in dealing with global optimization problems including the improved capability of high convergence speed and good generality for the whole problem.

3.4. The Framework of DM-PSO-ABC

In order to achieve better results on multimodal problems, Liang et al. [20] designed an improved algorithm in such a way that the particles have a larger diversity by sacrificing the convergence speed of the global PSO. Even after the globally optimal region is found, the particles will not converge rapidly to the globally optimal solution. Hence, maintaining the diversity and obtaining good solutions rapidly at the same time are a challenge which is tackled by integrating the neighbourhood search phase of artificial bee colony in the DMS-PSO to obtain the DM-PSO-ABC; the population of food sources (solutions) is initialized by the current lbest in each subswarm; the position of a food source represents a possible solution to the problem, and the nectar amount of a food source corresponds to the quality (fitness) of the associated solution. Thereafter, the nectar of food sources is exploited by employed bees and onlooker bees, and this continual exploitation will ultimately cause them to become exhausted. Except the ABC phase in each subswarm, the process of the paper [16] is also retained. In this way, the strong exploration abilities of the basic PSO and the exploitation abilities of the ABC can be fully exploited. The flowchart of the proposed DM-PSO-ABC is presented in Figure 3.

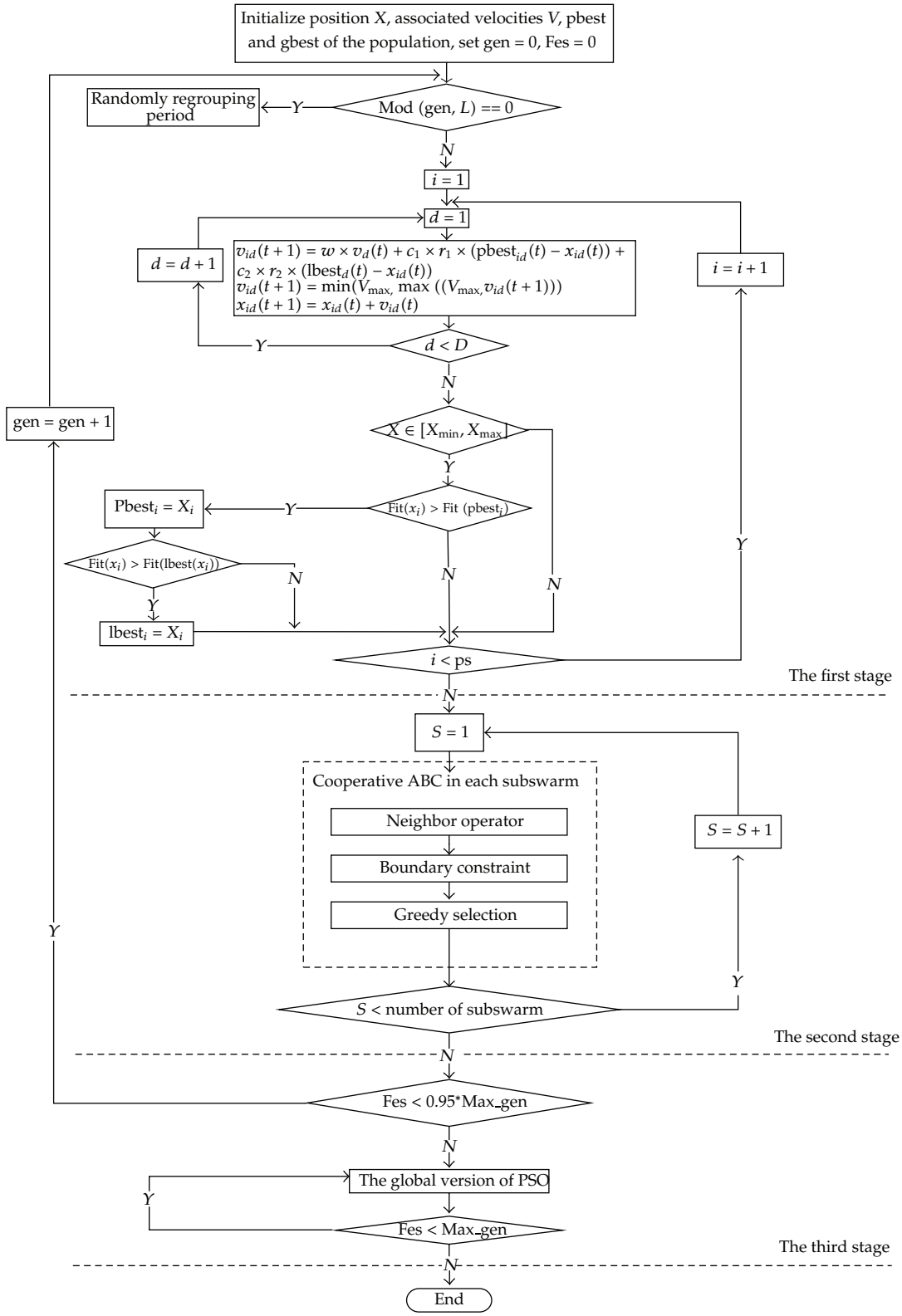

Figure 3: The flowchart of the DM-PSO-ABC.

4. Experimental Results and Discussions

4.1. Test Function

To investigate how DM-PSO-ABC performs in different environments, we chose 16 diverse benchmark problems [25–27]: 2 unimodal problems, 6 unrotated multimodal problems, 6 rotated multimodal problems, and 2 composition problems. All problems are tested with 10 and 30 dimensions. The properties and the formulas of these functions are presented briefly described in Table 1.

Note that the rotated functions are particularly challenging for many existing optimization algorithms. In case of rotations, when one dimension in the original vector $\vec{x}$ is changed, all dimensions of the rotated vector will be affected. To rotate a function, first an orthogonal matrix should be generated. The original variable x is left multiplied by the orthogonal matrix M to get the new rotated variable $y = M * x$. This variable y is used to calculate the fitness value f. IF $= \begin{bmatrix} m_{11} & m_{12} & \cdots & m_{1D} \\ m_{21} & m_{22} & \cdots & m_{2D} \\ \cdots & \cdots & \cdots & \cdots \\ m_{1D} & m_{2D} & \cdots & M_{DD} \end{bmatrix} x = [x_1, x_2, \ldots, x_D]^T$ and $y = [y_1, y_2, \ldots, y_D]^T$, then $y_i = m_{i1}x_1 + m_{i2}x_2 + \cdots + m_{iD}x_D$, $i = 1, 2, \ldots, D$.

When one dimension in x vector is changed, all dimensions in vector y will be affected. Hence, the rotated function cannot be solved by just D one-dimensional searches. In this paper, we used Salomon's method to generate the orthogonal matrix.

Composition functions are constructed using some basic benchmark functions to obtain more challenging problems with a randomly located global optimum and several randomly located deep local optima. The Gaussian function is used to combine the simple benchmark functions and blur the function's structures. The composition functions are asymmetrical multimodal problems, with different properties in different areas. The details of how to construct this class of functions and six composition functions are presented in [27]. The composition functions are characterized by nonseparable search variables, rotated coordinates, and strong multimodality due to a huge number of local optima. They blend together the characteristics of different standard benchmarks.

4.2. Parameter Settings for the Involved Algorithms

Experiments were conducted to compare five algorithms including the proposed DM-PSO-ABC algorithm on the 16 test problems with ten dimensions and 30 dimensions. The algorithms and parameters settings are listed below:

(i) DMS-PSO [25];

(ii) ABC [21];

(iii) fully informed particle swarm (FIPS) [16];

(iv) CLPSO [20];

(v) DM-PSO-ABC.

The DM-PSO-ABC parameters are set as follow: $w = 0.729$, $c_1 = c_2 = 1.49445$, $L = 5$. $V_{\max}$ restricts particles' velocities and is equal to 20% of the search range. To solve these problems, the number of subswarms is set at 10 which is also the same setting as in the DMS-PSO [25]. To tune the remaining parameters, nine selected test functions are used to investigate the impact of them. They are 10-dimensional test functions: f_1, f_2, f_3, f_4, f_5, f_6, f_9, f_{10}, f_{11}.

Table 1: Properties of the test function.

Function name	Expression	Search range	Optimization value
Sphere function	$f_1 = \sum_{i=1}^{D} x_i^2$	$[-100, 100]$	$\mathrm{Min}(f_1) = f_1(0, 0, \ldots, 0) = 0$
Rosenbrock function	$f_2 = \sum_{i=1}^{D} \left(100\left(x_{i+1} - x_i^2\right) + (x_i - 1)^2\right)$	$[-2.048, 2.048]$	$\mathrm{Min}(f_2) = f_2(1, 1, \ldots, 1) = 0$
Ackley function	$f_3(x) = -20\exp\left(-0.2\sqrt{\frac{1}{D}\sum_{i=1}^{D} x_i^2}\right) - p\left(\frac{1}{D}\sum_{i=1}^{D}\cos(2\pi x_i)\right) + 20 + e$	$[-32.768, 32.768]$	$\mathrm{Min}(f_3) = f_3(0, 0, \ldots, 0) = 0$
Griewank function	$f_4(x) = \frac{1}{400}\sum_{i=1}^{n}(x_i - 100)^2 - \prod_{i=1}^{n}\cos\left(\left(x_i - 100\sqrt{i}\right) + 1\right)$	$[-600, 600]$	$\mathrm{Min}(f_4) = f_4(0, 0, \ldots, 0) = 0$
Rastrigin function	$f_5 = \sum_{i=1}^{D}\left(x_i^2 - 10\cos(2\pi x_i) + 10\right)$	$[-5.12, 5.12]$	$\mathrm{Min}(f_5) = f_5(0, 0, \ldots, 0) = 0$
Noncontinuous Rastrigin function	$f_6(x) = \sum_{i=1}^{D}\left(y_i^2 - 10\cos(2\pi y_i) + 10\right)$, $y_i = \begin{cases} x_i & \lvert x_i\rvert < \frac{1}{2} \\ \frac{\mathrm{round}(2x_i)}{2} & \lvert x_i\rvert \geq \frac{1}{2} \end{cases}$ for $i = 1, 2, \ldots, D$.	$[-5.12, 5.12]$	$\mathrm{Min}(f_6) = f_6(0, 0, \ldots, 0) = 0$
Schwefel function	$f_7(x) = 418.9829 * D - \sum_{i=1}^{D} x_i * \sin\left(\lvert x_i\rvert^{1/2}\right)$	$[-500, 500]$	$\mathrm{Min}(f_7) = f_7(420.96, 420.96, \ldots, 420.96) = 0$
Weierstrass function	$f_8(x) = \sum_{i=1}^{D}\left(\sum_{k=0}^{k_{\max}}\left[a^k\cos\left(2\pi b^k(x_i + 0.5)\right)\right]\right) - D\sum_{k=0}^{k_{\max}}\left[a^k\cos\left(2\pi b^k * 0.5\right)\right]$, $a = 0.5$, $b = 3$, $k_{\max} = 20$	$[-0.5, 0.5]$	$\mathrm{Min}(f_8) = f_8(0, 0, \ldots, 0) = 0$
Rotated Ackley function	$f_9(x) = -20\exp\left(-0.2\sqrt{\frac{1}{D}\sum_{i=1}^{D} x_i^2}\right) - p\left(\frac{1}{D}\sum_{i=1}^{D}\cos(2\pi x_i)\right) + 20 + e, \quad y = M * x$	$[-32.768\ , 32.768]$	$\mathrm{Min}(f_9) = f_9(0, 0, \ldots, 0) = 0$

Table 1: Continued.

Function name	Expression	Search range	Optimization value
Rotated Griewank function	$f_{10}(x) = \frac{1}{400}\sum_{i=1}^{n}(x_i - 100)^2 - \prod_{i=1}^{n}\cos\left(\left(x_i - 100\sqrt{i}\right) + 1\right), \quad y = M * x$	$[-600, 600]$	$\text{Min}(f_{10}) = f_{10}(0, 0, \ldots, 0) = 0$
Rotated Rastrigin function	$f_{11}(x) = \sum_{i=1}^{D}(x_i^2 - 10\cos(2\pi x_i) + 10), \quad y = M * x$	$[-5.12, 5.12]$	$\text{Min}(f_{11}) = f_{11}(0, 0, \ldots, 0) = 0$
Rotated noncontinuous Rastrigin function	$f_{12}(x) = \sum_{i=1}^{D}(y_i^2 - 10\cos(2\pi y_i) + 10)$, $y_i = \begin{cases} x_i & \lvert x_i\rvert < \frac{1}{2} \\ \frac{\text{round}(2x_i)}{2} & \lvert x_i\rvert \geq \frac{1}{2} \end{cases}$ for $i = 1, 2, \ldots, D$, $y = M * x$	$[-5.12, 5.12]$	$\text{Min}(f_{12}) = f_{12}(0, 0, \ldots, 0) = 0$
Rotated Schwefel function	$f_{13}(x) = 418.9829 * D - \sum_{i=1}^{D} y_i$, $y_i = \begin{cases} x_i \sin\left(\lvert x_i\rvert^{1/2}\right) & \text{if } \lvert x_i\rvert \leq 500 \\ 0.001(\lvert x_i\rvert - 500)^2 & \text{if } \lvert x_i\rvert > 500 \end{cases}$ for $i = 1, 2, \ldots, D$, $x = x' + 420.96$, $x' = M * (z - 420.96)$	$[-500, 500]$	$\text{Min}(f_{13}) = f_{13}(420.96, 420.96, \ldots, 420.96) = 0$
Rotated Weierstrass function	$f_{14}(x) = \sum_{i=1}^{D}\left(\sum_{k=0}^{k_{\max}}\left[a^k \cos(2\pi b^k (x_i + 0.5))\right]\right) - D\sum_{k=0}^{k_{\max}}\left[a^k \cos(2\pi b^k * 0.5)\right]$ $a = 0.5$, $b = 3$, $k_{\max} = 20$, $y = M * x$	$[-0.5, 0.5]$	$\text{Min}(f_{14}) = f_{14}(0, 0, \ldots, 0) = 0$
Hybrid composition function 1 (CF 1)	f_{15} marked as $f_{16}(x)$ in the CEC2005 benchmark problem set [24] and is composed of ten sphere functions	$[-500, 500]$	$\text{Min(CF1)} = 0$
Hybrid composition function 2 (CF2)	f_{16} marked as $f_{16}(x)$ in the CEC2005 benchmark problem set [24] and is composed of two rotated Rastrigin functions, two rotated Weierstrass functions, two rotated Griewank functions, two rotated Ackley functions, and two sphere functions.	$[-500, 500]$	$\text{Min(CF2)} = 0$

Table 2: Parameter tuning of subswarm size (S).

f	S				
	3	5	8	10	15
f_1	$1.22E-078$	$2.33E-086$	$1.78E-093$	**$1.23E-094$**	$1.19E-098$
f_2	$2.15E-001$	$3.28E-001$	$2.35E+000$	**$7.77E-002$**	$2.11E+000$
f_3	$4.49E-015$	**$8.88E-016$**	$4.44E-015$	**$8.88E-016$**	$4.44E-015$
f_4	0	0	0	**0**	0
f_5	0	0	0	**0**	0
f_6	0	0	0	**0**	0
f_9	$4.44E-015$	$4.44E-015$	$8.88E-016$	**$4.44E-015$**	$4.44E-015$
f_{10}	$9.86E-003$	$6.96E-010$	$2.22E-007$	**$1.84E-016$**	$9.92E-003$
f_{11}	$2.98E+000$	$2.98E+000$	$1.99E+000$	**$1.01E+000$**	$2.98E+000$

Table 3: Parameter tuning of the regrouping period (L).

f	L				
	3	5	8	10	15
f_1	$1.06E-088$	**$4.59E-094$**	$2.01E-093$	$1.23E-093$	$1.00E-094$
f_2	$3.45E+000$	**$2.43E-002$**	$3.31E-001$	$7.77E-002$	$2.19E+000$
f_3	$4.44E-015$	**$8.88E-016$**	$4.44E-015$	$8.88E-016$	$8.88E-016$
f_4	0	**0**	0	0	0
f_5	0	**0**	0	0	0
f_6	0	**0**	0	0	0
f_9	$4.44E-015$	**$8.88E-016$**	$4.44E-015$	$4.44E-015$	$4.44E-015$
f_{10}	$5.16E-005$	**$4.88E-010$**	$7.39E-003$	$1.82E-007$	$9.93E-003$
f_{11}	$1.99E+000$	**$2.98E+000$**	$1.98E+000$	$1.48E+000$	$1.99E+000$

Experiments were conducted on these nine 10-dimensional test function, and the mean values of 30 runs are presented. The population size is set at 100, and the max iteration is set at 2000.

(1) Subswarm S

For each subswarm, the results of investigation on the selected test problems are shown in Table 2. In this table, the mean values of nine problems with different parameter settings are given. Based on the comparison of the results, the best setting is 10 particles for each subswarm. This is also the setting for the ABC population size. Hence, in DM-PSO-ABC the population size is 100 as there are 10 subswarms.

(2) Regrouping Iterations

For the regrouping iterations L, it should not be very small because we need to allow enough number of iterations for each subswarm to search. It should not also be too large because function evaluations will be wasted when the subswarm could not further improve. Table 3 presents the results of tuning L. Based on the results, the best value for L is 5.

4.3. Experimental Results and Discussions

4.3.1. Comparison Regarding Mean and Variance Values

For each function, the DMS-PSO-ABC, the DMS-PSO, the FIPS, the CLPSO, and the ABC are run 30 times. The maximum function evaluations Max_Fes are set at 100,000 for 10D, and 200,000 for 30D. The computer system is Windows XP (SP1) with Pentium (*R*) 4 3.00 GHz CPU, 4 GB RAM running the Matlab 7.1. For each function, we present the mean (the standard deviation) of the 30 runs in Tables 4 and 5.

Table 4 presents the means and variances of the 30 runs of the five algorithms on the 10D 16 test functions. The best results among the five algorithms are shown in bold. From the results, we observe that for the Group A unimodal problems, since DM-PSO-ABC has local search by ABC (exploit), it converged faster than other algorithms; we observe that DM-PSO-ABC has good performance in the multimodal groups. The f_4 is a good example, as it traps all other algorithms in local optima. The DM-PSO-ABC successfully avoids falling into the deep local optimum which is far from the global optimum. And the DM-PSO-ABC achieved the same best result as the CLPSO and the DMS-PSO on functions 5, 6, 7, and 8. The DM-PSO-ABC performs much better on rotated multimodal problems than others. On the two composition functions with randomly distributed local and global optima, DM-PSO-ABC performs the best.

From the results in Table 5, all 30D functions become more difficult than their 10D counterparts, and the results are not as good as in 10D cases, although we increased the maximum number of iterations from 2000 to 5000. The results of composition functions are not affected much. DM-PSO-ABC surpasses all other algorithms on functions 1, 2, 3, 5, 6, 7, 10, 12, 13, 15, and 16 and especially significantly improves the results on functions 5, 6, and 7. This implies that the DM-PSO-ABC is more effective in solving problems with dynamic multiswarm, and a local selection process of ABC may learn from different exemplars. Due to this, the DM-PSO-ABC explores a larger search space than the DMS-PSO and ABC algorithms. The larger search space is not achieved randomly. Instead, it is based on the historical search experience. Because of this, the DM-PSO-ABC performs comparably to or better than many algorithms on most of the problems experimented in this paper.

Comparing the results and the convergence graphs in Figure 4, among these five algorithms, DMS-PSO converges fast. But DM-PSO-ABC's local search ability is better than DMS-PSO. The ABC's performance is seriously affected after rotation, especially in the end of evolution; the population's diversity is rapidly reduced. CLPSO does not perform the best for unimodal f_1 and simple multimodal problems f_2; FIPSO with a U-ring topology of local versions; it presents good performance on some unrotated multimodal problems and converges faster when compared to DM-PSO-ABC. However, although DM-PSO-ABC's performance is also affected by the rotation, it still performs the best on four rotated problems. It can be observed that all PSO variants and ABC failed on the rotated Schwefel function, as it becomes much harder to solve after applying rotation.

4.3.2. Comparison Regarding the t-Test Results

By analyzing the results on 10D and 30D problems, one can conclude that the DM-PSO-ABC benefits from both the DMS-PSO and the ABC and the global PSO algorithms by integrating the faster convergent speed of the ABC as well as the stronger exploration ability of the DMS-PSO to tackle diverse problems in 16 functions. Therefore, it performs significantly better than

Table 4: Results for 10D problems.

Algorithm	F			
	f_1	f_2	f_3	f_4
DM-PSO-ABC	**7.98E − 102 (4.37E − 102)**	**2.60E − 002 (1.43E − 001)**	**1.48E − 016 (1.48E − 016)**	**0.00E + 000 (0.00E + 000)**
DMS-PSO	$5.69E-100$ $(3.24E-100)$	$1.49E+000$ $(1.32E+00)$	$8.95E-14$ $(4.26E-014)$	$1.64E-003$ $(3.56E-002)$
ABC	$4.45E-017$ $(1.13E-017)$	$4.63E+001$ $(1.56E+01)$	$7.81E-004$ $(1.83E-004)$	$8.37E-004$ $(1.38E-003)$
CLPSO	$5.15E-029$ $(2.16E-028)$	$2.46E+000$ $(1.70E+00)$	$4.32E-014$ $(2.55E-014)$	$4.56E-011$ $(4.81E-011)$
FIPS	$3.15E-030$ $(4.56E-030)$	$2.78E+000$ $(2.26E-001)$	$3.75E-015$ $(2.13E-014)$	$1.31E-001$ $(9.32E-002)$
	f_5	f_6	f_7	f_8
DM-PSO-ABC	**0.00E + 000 (0.00E + 000)**	**0.00E + 000 (0.00E + 000**	**0.00E + 000 (0.00E + 000)**	**0.00E + 000 (0.00E + 000)**
DMS-PSO	**0.00E + 000 (0.00E + 000)**	**0.00E + 000 (0.00E + 000)**	$1.73E+002$ $(5.64E+002)$	**0.00E + 000 (0.00E + 000)**
ABC	$4.36E+001$ $(1.44E+001)$	$2.19E+000$ $(3.42E+000)$	$1.26E+002$ $(3.36E+002)$	$3.56E-003$ $(6.41E-003)$
CLPSO	**0.00E + 000 (0.00E + 000)**	**0.00E + 000 (0.00E + 000)**	**0.00E + 000 (0.00E + 000)**	**0.00E + 000 (0.00E + 000)**
FIPS	$7.51E+001$ $(3.05E+001)$	$4.35E+000$ $(2.80E+000)$	$7.10E+001$ $(1.50E+002)$	$2.02E-003$ $(6.40E-003)$
	f_9	f_{10}	f_{11}	f_{12}
DM-PSO-ABC	**1.48E − 16 (8.11E − 016)**	**4.82E − 08 (2.64E − 007)**	**3.40E − 002 (1.8E − 001)**	**6.67E − 001 (0.36 + 000)**
DMS-PSO	$3.31E-015$ $(1.21E-015)$	$1.99E-002$ $(5.84E-002)$	$3.23E+000$ $(2.84E+000)$	$4.32E+000$ $(3.41E+000)$
ABC	$2.44E-014$ $(1.06E-014)$	$2.19E-001$ $(4.84E-001)$	$5.14E+001$ $(6.43E+001)$	$7.12E+002$ $(3.24E+002)$
CLPSO	$3.56E-005$ $(1.57E-004)$	$4.50E-002$ $(3.08E-002)$	$5.79E+000$ $(2.88E+000)$	$5.44E+000$ $(1.39E+000)$
FIPS	$2.25E-015$ $(1.54E-015)$	$1.7E-001$ $(1.26E-001)$	$1.20E-001$ $(6.22E+000)$	$8.84E+000$ $(3.27E+000)$
	f_{13}	f_{14}	f_{15}	f_{16}
DM-PSO-ABC	**1.58E + 000 (8.64E + 000)**	**0.00E + 000 (0.00E + 000)**	**3.07E + 000 (5.16E + 000)**	**8.04E − 001 (4.26E − 001)**
DMS-PSO	$3.66E+002$ $(1.31E+002)$	**0.00E + 000 (0.00E + 000)**	$1.33E+001$ $(2.01E+001)$	$1.64E+001$ $(3.14E+001)$
ABC	$1.23E+003$ $(2.13E+003)$	$1.07E-003$ $(8.61E-003$	$1.04E+003$ $(2.93E+003)$	$1.48E+002$ $(2.93E+002)$
CLPSO	$1.14E+002$ $(1.28E+002)$	$3.72E-010$ $(4.40E-010)$	$1.64E+001$ $(2.93E+001)$	$1.98E+001$ $(2.93E+001)$
FIPS	$2.89E+002$ $(2.00E+002)$	$5.93E-014$ $(1.86E-013)$	$6.00E+001$ $(5.16E+001)$	$4.21E+001$ $(6.37E+001)$

Table 5: Results for 30D problems.

Algorithm	F			
	f_1	f_2	f_3	f_4
DM-PSO-ABC	**1.10E − 74 (6.04E − 074)**	**7.01E − 001 (3.86 − E001)**	**1.48E − 016 (8.11E − 016)**	**0.00E + 000 (0.00E + 000)**
DMS-PSO	$1.26E-066$ $(2.14E-066)$	$1.98E+001$ $(1.23E+001)$	$6.51E-014$ $(8.26E-014)$	**0.00E + 000 (0.00E + 000**
ABC	$7.57E-004$ $(2.48E-004)$	$9.63E+000$ $(1.76E+000)$	$7.81E-004$ $(1.83E-004)$	$8.37E-004$ $(1.38E-003)$
CLPSO	$6.25E-019$ $(2.77E-010)$	$1.76E+001$ $(3.62E+000)$	$3.54E-010$ $(1.00E-010)$	$4.56E-010$ $(4.81E-010)$
FIPS	$2.18E-012$ $(5.87E-013)$	$2.41E+001$ $(2.19e-001)$	$4.81E-007$ $(9.17E-008)$	$1.16E-006$ $(1.87E-006)$
	f_5	f_6	f_7	f_8
DM-PSO-ABC	**0.00E + 000 (0.00E + 000)**	**0.00E + 000 (0.00E + 000)**	**0.00E + 000 (0.00E + 000)**	**0.00E + 000 (0.00E + 000)**
DMS-PSO	$1.40E+001$ $(2.03E+001)$	$1.87E+001$ $(2.03E+001)$	$1.72E+002$ $(0.00E+003$	**0.00E + 000 (0.00E + 000)**
ABC	$4.66E+000$ $(3.44E+000)$	$8.19E+001$ $(3.42E+002)$	$1.26E+003$ $(2.36E+003)$	$1.06E-002$ $(2.40E-003)$
CLPSO	$3.74E-009$ $(2.84E-009)$	$5.96E-009$ $(6.78E-009)$	$1.27E-012$ $(8.79E-013)$	$4.34E-012$ $(2.02E-012)$
FIPS	$7.30E+001$ $(1.24E+001)$	$6.08E+001$ $(8.35E+000)$	$1.18E-001$ $(8.35E-001)$	$1.14E-001$ $(1.48E-001)$
	f_9	f_{10}	f_{11}	f_{12}
DM-PSO-ABC	$1.99E-016$ $(1.21E-016)$	**0.00E + 000 (0.00E + 000)**	**6.30E − 001 (1.44E + 000)**	**5.97E − 001 (3.11E + 000)**
DMS-PSO	$4.85E-015$ $(1.21E-015)$	$2.31E-002$ $(4.84E-002)$	$2.81E+001$ $(2.84E+001)$	$3.20E+001$ $(2.41E+001)$
ABC	**3.14E − 017 (1.06E − 017)**	$4.11E+000$ $(4.84E+001)$	$3.18E+002$ $(6.23E+001)$	$5.12E+002$ $(7.44E+002)$
CLPSO	$3.56E-005$ $(1.57E-004)$	$4.50E-002$ $(3.08E-002)$	$5.79E+000$ $(2.88E+000)$	$5.44E+000$ $(1.39E+000)$
FIPS	$2.25E-015$ $(1.54E-015)$	$1.7E-001$ $(1.26E-001)$	$1.20E-001$ $(6.22E+000)$	$8.84E+000$ $(3.27E+000)$
	f_{13}	f_{14}	f_{15}	f_{16}
DM-PSO-ABC	**6.66E + 001 (3.65E + 002)**	$1.08E-013$ $(1.47E-013)$	**1.08E + 000 (4.13E + 000)**	**6.26E + 000 (4.92E + 000)**
DMS-PSO	$3.21E+003$ $(1.31E+003)$	$1.11E-002$ $(2.31E-002)$	$1.33E+001$ $(1.01E+001)$	$1.93E+001$ $(2.04E+001)$
ABC	$3.01E+003$ $(3.31E+003)$	$1.01E-002$ $(2.31E-002)$	$2.04E+002$ $(2.93E+002)$	$1.63E+002$ $(2.61E+002)$
CLPSO	$1.14E+002$ $(1.28E+002)$	$3.72E-010$ $(4.40E-010)$	$1.64E+001$ $(2.93E+001)$	$5.06E+001$ $(2.93E+001)$
FIPS	$2.89E+002$ $(2.00E+002)$	**5.93E − 014 (1.86E − 013)**	$6.00E+001$ $(5.16E+001)$	$4.21E+001$ $(6.37E+001)$

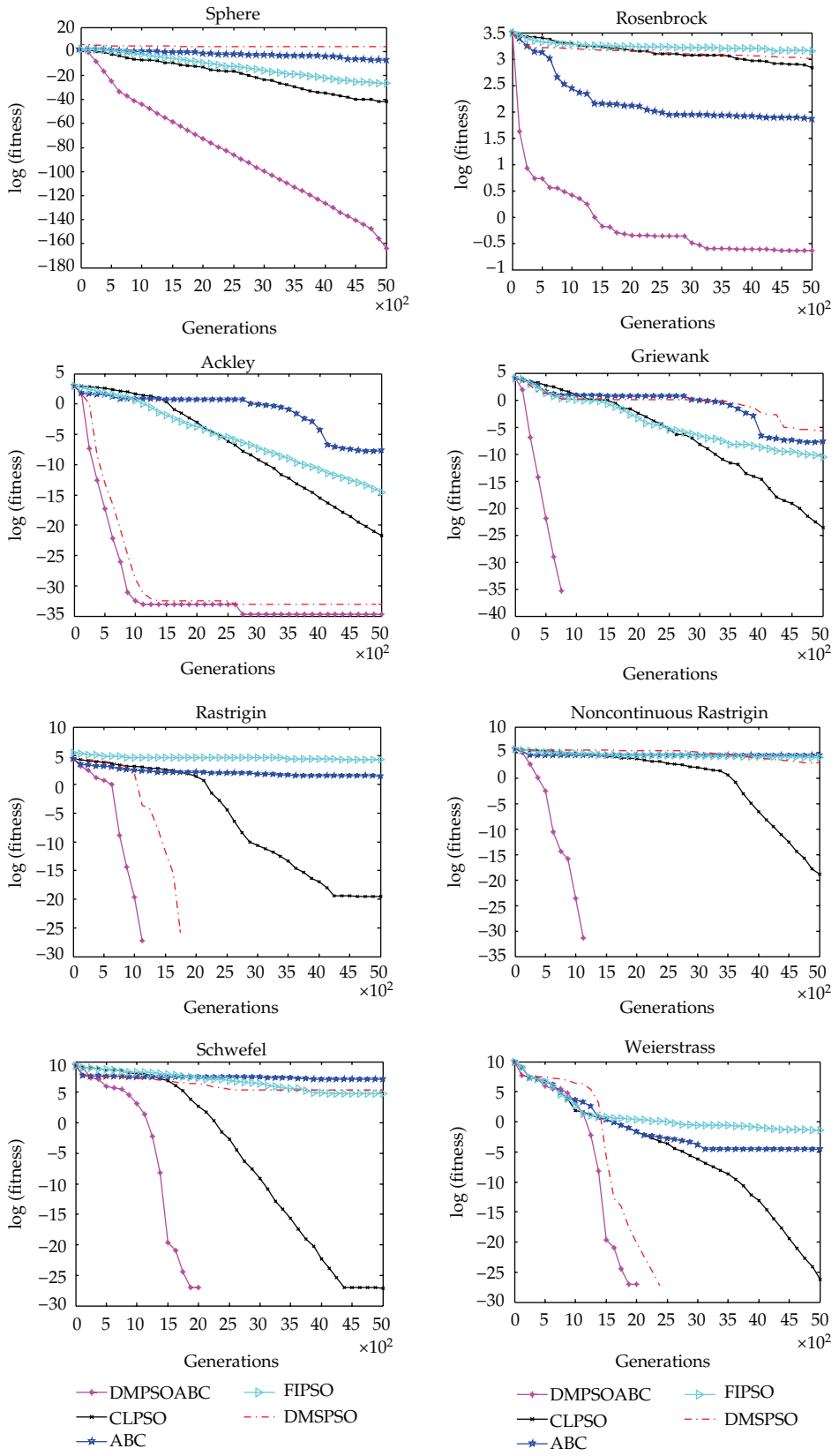

Figure 4: Continued.

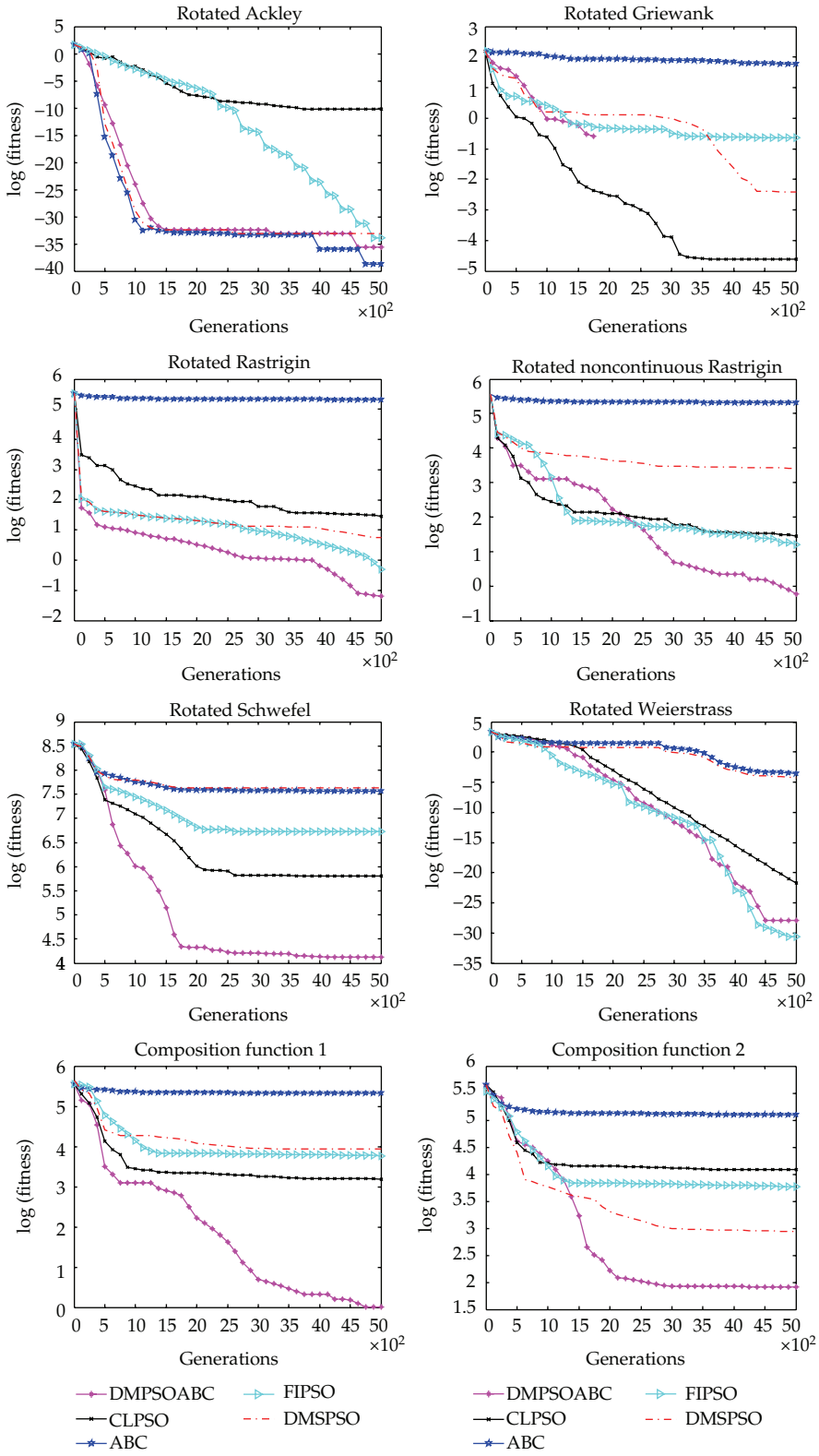

Figure 4: The median convergence characteristics of 30-D test functions.

Table 6: DM-PSO-ABC differs from other algorithms with bilateral t-testing method.

Algorithm	F							
	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8
DMS-PSO	5.559 (+)	1.231 (−)	1.046 (−)	1.235 (−)	23.530 (+)	7.903 (+)	8.406 (+)	0.642 (−)
ABC	7.484 (+)	2.035 (−)	2.517 (+)	3.017 (+)	8.826 (+)	7.912 (+)	11.071 (+)	5.463 (+)
CLPSO	2.903 (+)	0.737 (−)	1.445 (−)	2.932 (+)	1.956 (−)	1.620 (−)	1.935 (−)	1.679 (−)
FIPS	4.032 (+)	1.94 (−)	1.874 (−)	4.728 (+)	3.476 (+)	6.946 (+)	4.632 (+)	3.336 (+)
	f_9	f_{10}	f_{11}	f_{12}	f_{13}	f_{14}	f_{15}	f_{16}
DMS-PSO	0.521 (−)	5.034 (+)	2.152 (−)	2.634 (+)	3.445 (+)	3.078 (+)	1.563 (−)	1. 643 (−)
ABC	1.624 (−)	9.525 (+)	2.845 (+)	2.967 (+)	2.907 (+)	3.864 (+)	2.086 (+)	2.069 (+)
CLPSO	3.743 (+)	4.974 (+)	2.004 (−)	1.735 (−)	1.764 (−)	1.863 (−)	1.964 (−)	1.896 (−)
FIPS	0.864 (−)	7.452 (+)	0.042 (−)	1.466 (−)	1.644 (−)	0.975 (−)	2.097 (+)	2.184 (+)

the two constituent approaches. Furthermore, most results are markedly improved by the proposed DM-PSO-ABC. The DMS-PSO-ABC can perform robustly with respect to scaling of the dimensions, rotation, and composition of difficult multimodal test problems.

Because a lot of experimental data were produced in this group contrast test, resulting in analyzing the performance of these algorithms with many difficulties, so we adopt bilateral t-testing method to analyze the experimental data; hence, we can objectively evaluate the difference between DM-PSO-ABC algorithm and the other algorithms. Reference [28] is conducted at the 5% significance level in order to judge whether the results obtained with the best performing algorithm differ from the final results of the rest of the competitors in a statistically significant way. Set the significance level $\alpha = 0.05$, because each function run 30 times, so the number of degrees of freed $d_f = 29$. From the t-distribution table, we can get $t_{0.05(29)} = 2.045$. $t = \overline{d}/S_{\overline{d}}$, $\overline{d}$ Is the mean of Inspection data differential, $S_{\overline{d}}$ the standard error of $\overline{d}$, the result of t in Table 6, $|t| < t_{0.05(29)}$, and mark "+" indicates that DM-PSO-ABC performs statistically better than the corresponding algorithm. On the other hand, the "−" mark indicates that the corresponding algorithm is better than DM-PSO-ABC. The last line is the sum of benchmark function whose optimization results have significant differences.

As revealed by Table 6, DM-PSO-ABC performs statistically better than the DMS-PSO in f_1, f_5, f_6, f_7, f_{10}, f_{12}, f_{13}, f_{14}; DM-PSO-ABC performs statistically better than the ABC f_1, f_3, f_4, f_5, f_6, f_7, f_8, f_{10}, f_{14}, f_{11}, f_{12}, f_{13}, f_{14}, f_{15}, f_{16}; DM-PSO-ABC performs statistically better than the CLPSO f_1, f_4, f_9, f_{10}. There are 7 functions DM-PSO-ABC performs statistically better than the FIPS in f_1, f_4, f_5, f_6, f_7, f_8, f_{10}, f_{15}, f_{16}. We also note that, in all benchmark instances, the performance of DM-PSO-ABC is statistically superior to the performance of the four other state-of-the-art algorithms. This difference in performance must be attributed to multistage search mechanisms, a fact that substantiates the usefulness of the modifications incorporated in DM-PSO-ABC.

5. Conclusion

This paper proposes a hybridization of dynamic multigroup particle swarm optimizer with the artificial bee colony (DM-PSO-ABC). Using the dynamic multi-swarm particle swarm optimizer with a randomized regrouping schedule and a local search of ABC algorithm, we periodically generate the nectar information of the food sources (solutions) based on the current pbests in each subswarm after particles' positions have been updated. The nearest

pbest is replaced by the new nectar if the new nectar vector has better fitness by ABC algorithm. The DM-PSO-ABC attempts to take merits of the DMS-PSO and the ABC in order to avoid all particles getting trapped in inferior local optimal regions. The DM-PSO-ABC enables the particles to have more diverse exemplars to learn from as we frequently regroup the subswarms. From the analysis of the experimental results, we observe that the proposed DM-PSO-ABC makes good use of the information in past solutions more effectively to generate frequently better quality solutions when compared to the other peer algorithms. The novel configuration of DM-PSO-ABC does not introduce additional complex operations beyond the original PSO and ABC. In fact, the DM-PSO-ABC eliminates the parameters in the original PSO, which normally need to be adjusted according to the properties of the test problems. In addition, the DM-PSO-ABC is simple and easy to implement.

Acknowledgments

The authors would like to thank all the reviewers for their constructive comments. The authors thank Mr. Suganthan for his thoughtful help and constructive suggestion. This research was supported by the National Natural Science Foundation of China (no. 70971020), the Open Project Program of Artificial Intelligence Key Laboratory of Sichuan Province (Sichuan University of Science and Engineering), China (no. 2012RYJ03), the fund Project of Hunan province, China (no. 11C1096).

References

[1] M. El-Abd, "A hybrid ABC-SPSO algorithm for continuous function optimization," in *Proceedings of the IEEE Symposium on Swarm Intelligence (SIS '11)*, pp. 96–101, April 2011.

[2] T.-J. Hsieh and W.-C. Yeh, "Mining financial distress trend data using penalty guided support vector machines based on hybrid of particle swarm optimization and artificial bee colony algorithm," *Neurocomputing*, pp. 196–206, 2012.

[3] X. Shi, Y. Li, H. Li, R. Guan, L. Wang, and Y. Liang, "An integrated algorithm based on artificial bee colony and particle swarm optimization," in *Proceedings of the 6th International Conference on Natural Computation (ICNC '10)*, pp. 2586–2590, August 2010.

[4] E. Turanoğlu and E. Özceylan, "Particle swarm optimization and artificial bee colony approaches tooptimize of single input-output fuzzy membership function," in *Proceedings of the 41st International Conference on Computers and Industrial Engineering*, pp. 542–547, 2011.

[5] D. Teodorović, P. Lučić, G. Marković, and M. Dell'Orco, "Bee colony optimization.principles and applications," in *Proceedings of the 8th Seminar on Neural Network Applications in Electrical Engineering (Neurel '06)*, pp. 151–156, September 2006.

[6] J. Kennedy, "Small worlds and mega-minds: effects of neighborhood topology on particle swarm performance," in *Proceedings of the IEEE Congress on Evolutionary Computation*, pp. 1931–1938, Piscataway, NJ, USA, 1999.

[7] M. Ballerini, N. Cabibbo, R. Candelier et al., "Interaction ruling animal collective behavior depends on topological rather than metric distance: evidence from a field study," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 105, no. 4, pp. 1232–1237, 2008.

[8] H. Hildenbrandt, C. Carere, and C. K. Hemelrijk, "Self-organized aerial displays of thousands of starlings: a model," *Behavioral Ecology*, vol. 21, no. 6, pp. 1349–1359, 2010.

[9] S. Motsch and E. Tadmor, "A new model for self-organized dynamics and its flocking behavior," *Journal of Statistical Physics*, vol. 144, pp. 923–947, 2011.

[10] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of the 4th IEEE International Conference on Neural Networks*, pp. 1942–1948, December 1995.

[11] F. van den Bergh and A. P. Engelbrecht, "A cooperative approach to participle swam optimization," *IEEE Transactions on Evolutionary Computation*, vol. 8, no. 3, pp. 225–239, 2004.

[12] R. Poli, J. Kennedy, and T. Blackwell, "Particle swarm optimization: an overview," *Swarm Intelligence*, vol. 1, no. 1, pp. 33–58, 2007.

[13] D. Bratton and J. Kennedy, "Defining a standard for particle swarm optimization," in *Proceedings of the IEEE Swarm Intelligence Symposium (SIS '07)*, pp. 120–127, April 2007.
[14] J. Kennedy, "Small worlds and mega-minds: effects of neighborhood topology on particle swarm performance," in *Proceedings of the Congress on Evolutionary Computation*, pp. 1931–1938, 1999.
[15] J. Kennedy and R. Mendes, "Population structure and particle swarm performance," in *Proceedings of the Congress on Evolutionary Computation.*, pp. 1671–1676, 2002.
[16] 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.
[17] X. Li, "Niching without niching parameters: particle swarm optimization using a ring topology," *IEEE Transactions on Evolutionary Computation*, vol. 14, no. 1, pp. 150–169, 2010.
[18] P. N. Suganthan, "Particle swarm optimizer with neighborhood operator," in *Proceedings of the Congress on Evolutionary Computation*, pp. 1958–1962, 1999.
[19] S. Janson and M. Middendorf, "A hierarchical particle swarm optimizer and its adaptive variant," *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, vol. 35, no. 6, pp. 1272–1282, 2005.
[20] 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.
[21] D. Karaboga, "An idea based on honey bee swarm for numerical optimization," Tech. Rep. TR06, Computer Engineering Department, Erciyes University, 2005.
[22] D. Karaboga and C. Ozturk, "A novel clustering approach: Artificial Bee Colony (ABC) algorithm," *Applied Soft Computing Journal*, vol. 11, no. 1, pp. 652–657, 2011.
[23] T. J. Hsieh, H. F. Hsiao, and W. C. Yeh, "Forecasting stock markets using wavelet transforms and recurrent neural networks: an integrated system based on artificial bee colony algorithm," *Applied Soft Computing Journal*, vol. 11, no. 2, pp. 2510–2525, 2011.
[24] M. Sonmez, "Artificial Bee Colony algorithm for optimization of truss structures," *Applied Soft Computing Journal*, vol. 11, no. 2, pp. 2406–2418, 2011.
[25] J. J. Liang and P. N. Suganthan, "Dynamic multi-swarm particle swarm optimizer," in *Proceedings of the IEEE Swarm Intelligence Symposium (SIS '05)*, pp. 124–129, June 2005.
[26] X. Yao, Y. Liu, and G. Lin, "Evolutionary programming made faster," *IEEE Transactions on Evolutionary Computation*, vol. 3, no. 2, pp. 82–102, 1999.
[27] Y. W. Shang and Y. H. Qiu, "A note on the extended Rosenbrock function," *Evolutionary Computation*, vol. 14, no. 1, pp. 119–126, 2006.
[28] S. García, D. Molina, M. Lozano, and F. Herrera, "A study on the use of non-parametric tests for analyzing the evolutionary algorithms' behaviour: a case study on the CEC'2005 Special Session on Real Parameter Optimization," *Journal of Heuristics*, vol. 15, no. 6, pp. 617–644, 2009.