Hindawi Publishing Corporation
The Scientific World Journal
Volume 2014, Article ID 370691, 11 pages
http://dx.doi.org/10.1155/2014/370691

Research Article

Cooperative Quantum-Behaved Particle Swarm Optimization with Dynamic Varying Search Areas and Lévy Flight Disturbance

Desheng Li

Anhui Science and Technology University, Fengyang, Anhui 233100, China

Correspondence should be addressed to Desheng Li; ldsyy2006@126.com

Received 7 October 2013; Accepted 2 January 2014; Published 3 March 2014

Academic Editors: Z. Cui and X. Yang

This paper proposes a novel variant of cooperative quantum-behaved particle swarm optimization (CQPSO) algorithm with two mechanisms to reduce the search space and avoid the stagnation, called CQPSO-DVSA-LFD. One mechanism is called Dynamic Varying Search Area (DVSA), which takes charge of limiting the ranges of particles' activity into a reduced area. On the other hand, in order to escape the local optima, Lévy flights are used to generate the stochastic disturbance in the movement of particles. To test the performance of CQPSO-DVSA-LFD, numerical experiments are conducted to compare the proposed algorithm with different variants of PSO. According to the experimental results, the proposed method performs better than other variants of PSO on both benchmark test functions and the combinatorial optimization issue, that is, the job-shop scheduling problem.

1. Introduction

PSO, originally introduced by Kennedy and Eberhart [1], has become one of the most important swarm intelligence-based algorithms. The unique information diffusion and interaction mechanism of PSO enable it to solve many problems with good performance at low computational cost. Hence, in the past decades, PSO is imported to solve the problems of numerical optimization [2], combinatorial optimization [3], controller optimization [4, 5], and software design optimization [6].

Among the applications, function optimization has been often chosen to check the performance of them, because benchmark functions are not only well described in literature such as their properties, locations, and values of the optimal solutions, but also have many different versions that can rove different capabilities of optimizer [7, 8]. However, according to the state of art in the relevant research [9] and in spite of its superior performance, PSO is even not a global optimization algorithm. To overcome this defect, some randomizing techniques are employed into the design of PSO, such as chaos [10] and quantum behavior [11, 12], to accelerate the global convergence of the algorithms. In literatures [11, 12], Sun et al. proposed a quantum-behaved PSO (QPSO) algorithm, which can be guaranteed theoretically to find optimal solution in search space. The experimental results on some widely used benchmark functions show that the QPSO works better than standard PSO and should be a promising algorithm.

Like all other intelligence algorithms [8], escaping from the local optimum and preventing premature convergences are two inevitable difficulties in implementation. Especially, as dimensionality increases, these kinds of problems become more complex and the possibility for finding global optimum sharply decreases. Nevertheless, some applications really need to probe the global optimal solutions rather than local ones, such as function optimization and clusters structure optimization. Hence, the main motivation of this research is to find a solution to make the PSO adapt to the multidimension and difficult problems, for example, NP hard ones.

This paper proposes a novel particle swarm algorithm with two different methods to reduce the search space. One is called Dynamic Varying Search Area (DVSA), which takes charge of limiting the ranges of particles' activity, and the other is cooperative strategy, which divides the candidate solution vector into small subswarms. Moreover, in order to escape the local optima, Lévy flights are used to generate the stochastic disturbance in the movement of particles.

The remainder of this article is organized as follows. Section 2 provides a brief review on the related algorithms. Section 3 proposes the CQPSO-DVSA-LFD and gives the

main flow of the algorithm. Sections 4 and 5 illustrate the inner mechanisms, Dynamic Varying Search Area (DVSA), and Lévy Flights Disturbance (LFD), respectively. Section 6 describes the experimental framework and then presents and discusses the numerical results from the trails. Finally, Section 7 offers our conclusions and future work.

2. Review on Related PSO Algorithms

2.1. PSO. PSO algorithm was first introduced by Kennedy and Eberhart [1] as a simulation of the flock's behavior but quickly evolved into one of the most powerful optimization algorithms in the computational intelligence field. The algorithm consists of a population of particles that are flown through an n-dimensional search space. The position of each particle represents a potential solution to the optimization problem and is used in determining the fitness (or performance) of a particle. In each generation of iteration, particle in swarm can be updated by the values of the best solution found by it and the one found by the whole swarm by far according to the following equation:

$$\begin{aligned} v_{\mathrm{id}}^{t+1} &= \omega \times v_{\mathrm{id}}^{t} + c_1 \times r_1 \times \left(P_{\mathrm{id}}^{\mathrm{best}} - P_{\mathrm{id}}^{t}\right) \\ &\quad + c_2 \times r_2 \times \left(P_{\mathrm{gd}}^{\mathrm{best}} - P_{\mathrm{id}}^{t}\right), \\ P_{\mathrm{id}}^{t+1} &= P_{\mathrm{id}}^{t} + v_{\mathrm{id}}^{t+1}, \end{aligned} \tag{1}$$

where v_{id}^{t} denotes the particle's velocity in t generation and P_{id}^{t}, $P_{\mathrm{id}}^{\mathrm{best}}$, and $P_{\mathrm{gd}}^{\mathrm{best}}$ are the particle's position, personal historical best position, and swarm's global best position in t generation, respectively.

2.2. CPSO. In practice, most variants of standard PSO suffer from the curse of dimensionality, which may directly reduce to the performance deterioration. Therefore, it also causes the failure of finding the global optimum of a highdimensional problem. One effective way is to decompose the large search space into smaller swarms in lower dimensional vector space, that is, to employ cooperative strategy. Based on the rationale of Cooperative Coevolutionary Genetic Algorithm (CCGA) in [13], van den Bergh and Engelbrecht introduced the Cooperative PSO that employs a kind of cooperative behavior to significantly improve the performance of the original algorithm [9]. Compared to basic single swarm PSO, both robustness and precision are improved and guaranteed. The key idea of CPSO is to divide all the n-dimension vectors into k subswarms. So the front n/k swarms are $\lceil n/k \rceil$-dimensional and the $k - (n/k)$ swarms behind have $\lfloor n/k \rfloor$-dimensional vectors. In each pass of iteration, the solution is updated based on k subswarms rather than the original one. When the particles in one subswarm complete a search along some component, their latest best position will be combined with other subswarms to generate a whole solution.

The function b shown in (2) performs exactly like this: it takes the best particle from each of the other subswarms and concatenates them, splicing in the current particle from the current subswarm j in the appropriate position. According to this function, the composition of $P_{\mathrm{id}}^{\mathrm{best}}$, $P_{\mathrm{gd}}^{\mathrm{best}}$, and $P_{\mathrm{cgd}}^{\mathrm{best}}$ can be calculated based on (3)–(5). Due to the employment of this component, the particles in each subswarm therefore update their global best position by (6)-(7), which is the result associated with minimal fitness value of their local best positions and global best positions of electoral swarm:

$$\begin{aligned} b(u, Z) = \big(&S_1 \cdot P_{\mathrm{gd}}^{\mathrm{best}}, \ldots, S_{u-1} \cdot P_{\mathrm{gd}}^{\mathrm{best}}, \\ &Z, S_u \cdot P_{\mathrm{gd}}^{\mathrm{best}}, \ldots, S_k \cdot P_{\mathrm{gd}}^{\mathrm{best}}\big), \quad 1 \le u \le k, \end{aligned} \tag{2}$$

$$\begin{aligned} &b\left(u, S_u \cdot P_{\mathrm{id}}^{\mathrm{best}}\right) \\ &\quad = \operatorname{argmin} \operatorname{fitness}\left(b\left(u, S_u \cdot P_{\mathrm{id}}^{\mathrm{best}}\right), b\left(u, S_u \cdot P_{\mathrm{id}}^{t}\right)\right), \\ &\qquad 1 \le \mathrm{id} \le s, \quad 1 \le u \le k, \end{aligned} \tag{3}$$

$$\begin{aligned} b\left(u, S_u \cdot P_{\mathrm{gd}}^{\mathrm{best}}\right) &= \operatorname{argmin} \operatorname{fitness}\left(b\left(u, S_u \cdot P_{\mathrm{id}}^{t}\right)\right), \\ &\qquad 1 \le \mathrm{id} \le s, \quad 1 \le u \le k, \end{aligned} \tag{4}$$

$$\begin{aligned} &b\left(u, S_u \cdot P_{\mathrm{cgd}}^{\mathrm{best}}\right) \\ &\quad = \operatorname{argmin} \operatorname{fitness}\left(b\left(u, S_u \cdot P_{\mathrm{id}}^{\mathrm{best}}\right), b\left(u, S_u \cdot P_{\mathrm{cgd}}^{\mathrm{best}}\right)\right), \\ &\qquad 1 \le \mathrm{id} \le s, \quad 1 \le u \le k, \end{aligned} \tag{5}$$

$$\begin{aligned} v_{\mathrm{id}}^{t+1} &= \omega \times v_{\mathrm{id}}^{t} + c_1 \times r_1 \times \left(P_{\mathrm{id}}^{\mathrm{best}} - P_{\mathrm{id}}^{t}\right) + c_2 \times r_2 \\ &\quad \times \left(P_{\mathrm{gd}}^{\mathrm{best}} - P_{\mathrm{id}}^{t}\right) + c_2 \times r_2 \times \left(P_{\mathrm{cgd}}^{\mathrm{best}} - P_{\mathrm{id}}^{t}\right), \end{aligned} \tag{6}$$

$$P_{\mathrm{id}}^{t+1} = P_{\mathrm{id}}^{t} + v_{\mathrm{id}}^{t+1}. \tag{7}$$

2.3. QPSO. In literature [14], Liu et al. proposed a Quantum Particle Swarm Optimization (QPSO), which discards the velocity vector of original PSO and consequently changes the updating strategy of particles' position to make the search more simple and efficient. QPSO is the integration of quantum computing and PSO. The QPSO is based on the representation of the quantum state vector. It applies the probabilistic amplitude representation of quantum to the coding of particles, which makes one particle represent the superposition of many states and uses quantum rotation gates to realize the update operation.

The iterative equation of QPSO is very different from that of PSO. Besides, unlike PSO, QPSO needs no velocity vectors for particles and also has fewer parameters to adjust, making it easier to implement. This can be seen clearly in Figure 1.

The updated strategy of particles' position of QPSO is as follows:

$$P_{\mathrm{id}}^{t+1} = \varphi \times P_{\mathrm{id}}^{\mathrm{best}} + (1 - \varphi) \times P_{\mathrm{gd}}^{\mathrm{best}} \pm \beta \times \left|C_d - P_{\mathrm{id}}^{t}\right| \times \ln\left(\frac{1}{u}\right), \tag{8}$$

where N is population of particles; D is dimension of problem; P_{id}^{t} is position of particle; $P_{\mathrm{id}}^{\mathrm{best}}$ is local best position; $P_{\mathrm{gd}}^{\mathrm{best}}$ is global best position; $C_d = (1/N) \sum_{i=1}^{N} P_{\mathrm{id}}$, $d = 1, \ldots, D$, $C = C_1, \ldots; C_d$ is mean local best positions.

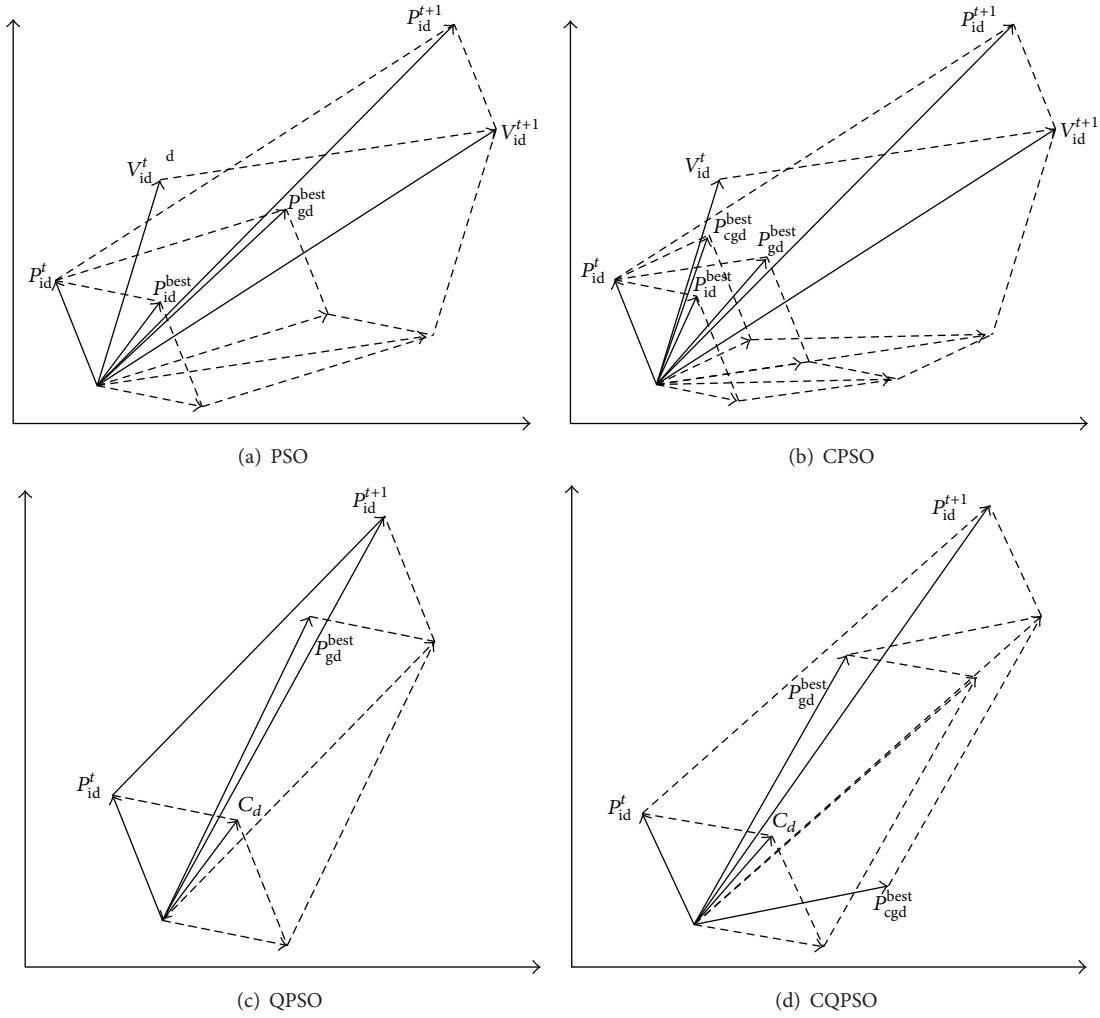


Figure 1: Particle movement principle of PSO, CPSO, QPSO, and CQPSO.

2.4. CQPSO. In CQPSO algorithm proposed in [15], an electoral swarm is generated by the voting of primitive subswarms and also participates in evolution of swarm, whose candidate particles come from primitive subswarms with variable votes. In reverse, the number of selected particles could also impact the voting of the primitive subswarms, such as the total number of candidates and quota of selected ones. The selected candidates could share their components with best segments of position, which are then composed into a new particle position to participate in the combining of positions. Like the treatment in our previous work, a new component of particle's position is also imported, that is, P_{cgd}^{best}, denoting the electoral best position composed by the dimensions of elected candidates.

The updating strategy of particles' position of CQPSO is as follows:

$$P_{id}^{t+1} = \varphi \times P_{id}^{best} + \psi \times P_{gd}^{best} + (1 - \varphi - \psi) \times P_{cgd}^{best} \pm \beta \times \left| C_d - P_{id}^{t} \right| \times \ln\left(\frac{1}{u}\right). \quad (9)$$

3. Proposed Algorithm: CQPSO-DVSA-LFD

In this paper a novel particle swarm algorithm with two different methods is proposed to reduce the search space. One is called Dynamic Varying Search Area (DVSA), which takes charge of limiting the ranges of particles' activity and the other is cooperative strategy, which divides the candidate solution vector into small subswarms. The illustration of DVSA will be introduced in Section 4.

On the other hand, theoretically, CQPSO algorithm could solve any problem by QPSO. However, due to the possibility of trapping in a case that all subswarm could not find the optimum, CQPSO could also reach the current minimum. To avoid this kind of stagnation, we employ a stochastic disturbance method, that is, Lévy flights disturbance, to generate a random movement of the stagnant subswarms. The details of Lévy flights disturbance will be introduced in Section 5.

Differently with other similar methods, we use the output parameters of Lévy flights to intervene the position change directly, which can be seen in (10) as follow, where $\text{Angle}_{\text{Lévy}}$

Initiation;
Label 1: Generation primitive sub-swarms;
Foreach sub-swarm-i **In** sub-swarms **Do**
  Calculate the fitness value;
  If (run==first-time)
  Then Update the personal and global optimal position as in QPSO;
  Else Update the personal and global optimal position with;
  Calculate the best particles;
  Check the condition of DVSA, if not satisfied, and then go to the second step.
  Calculate the reduced search area.
End Foreach
Calculate the fitness value;
Foreach dimension-i **In** D **Do**
  Update the personal and global optimal position;
  Update the particles based on quantum behavior with the Lévy disturbance;
End Foreach
Calculate the electoral best position;
Test whether satisfy the condition of termination;
If (Meet terminal condition) **Then** ends
Else repeat from **Label 1**;
End If
End.

Algorithm 1: Pseudocode of CQPSO-DVSA-LFD.

and $\text{Step}_{\text{Lévy}}$ are the output parameters of Lévy flights which are randomly generated, while $\varepsilon_1, \varepsilon_2$, and ε_3 are the parametric empirical coefficient:

$$\begin{aligned} P_{\text{id}}^{t+1} &= \varphi \times P_{\text{id}}^{\text{best}} + \psi \times \left(P_{\text{gd}}^{\text{best}} + \varepsilon_1 \times \text{Angle}_{\text{Lévy}}\right) \\ &\quad + (1 - \varphi - \psi) \times \left(P_{\text{cgd}}^{\text{best}} + \varepsilon_2 \times \text{Angle}_{\text{Lévy}}\right) \\ &\quad \pm \beta \times \left|\left(C_d + \varepsilon_3 \times \text{Step}_{\text{Lévy}}\right) - P_{\text{id}}^{t}\right| \times \ln\left(\frac{1}{u}\right). \end{aligned} \tag{10}$$

Based on the above introduction, we can now present the proposed CQPSO-DVSA-LFD algorithm in the following steps in Algorithm 1.

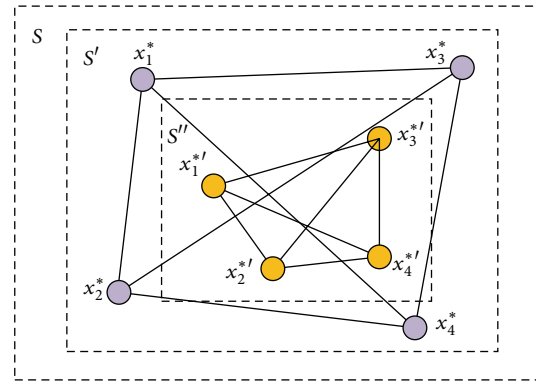


Figure 2: A case of DVSA.

4. Dynamic Varying Search Area (DVSA)

4.1. Rationale of Dynamic Varying Search Area (DVSA). As we know, complexity of optimization problem does not only rely heavily on the objective/constraint function but also relates to its search area. Simply speaking, subjected to the same objective/constraint function, the larger the search area is, the harder it can find the solution [16]. Based on this idea, to change the search area dynamically, or say it reduces, is necessary to accelerate the processing of algorithm. On the other hand, when the search area reduced, the populations of subswarms are unnecessary as big as previous ones.

Given an optimization function:

$$\min\ f(x), \quad x = \left(x_1, x_2, \ldots, x_{N_d}\right)^T \in S \subseteq R^{N_d}, \tag{11}$$

where $S = [a_1, b_1] \times [a_2, b_2] \times \cdots \times [a_{N_d}, b_{N_d}]$, the basic rationale of Dynamic Varying Search Area (DVSA) could be illustrated as the following description. Firstly, assume that N_p cooperative subswarms probe is in the search space. When the minimal distances between optimal individuals of each subswarm reached a threshold, according to the maximum likelihood estimation, the hypothesis that the real optimal solution is in the area surrounded by these particles was established. Then reduce the previous search area S to S', generate a new swarm with same subswarms on S', and decrease the populations meanwhile. Finally, repeat the above procedures until satisfying the end condition.

Consider the vector x before the rth reduce, where the ith component x_i ranges over $[a_i^{r-1}, b_i^{r-1}]$. Then x could be expressed as $x^{r-1} \in [a_i^{r-1}, b_i^{r-1}]$. Figure 2 exemplifies the case that four cooperative subswarms reduce their search area. First, they probe the solution in S and get the best particles x_1^*, x_2^*, x_3^*, and x_4^* included in S'. So the search area becomes S'. The next time of reduction to S'' is the same procedure.

4.2. Condition of DVSA. In this part, we will give the condition when the DVSA occurs. Suppose there exist N_p subswarms, the best particles set found is written as

$$\begin{aligned} x_b^{r-1} &= \left\{x_b^{r-1,1}, x_b^{r-1,2}, \ldots, x_b^{r-1,N_p}\right\}, \\ x_b^{r-1,p} &= \left(x_{b1}^{r-1,p}, x_{b2}^{r-1,p}, \ldots, x_{bN_d}^{r-1,p}\right), \\ & p = 1, 2, \ldots, N_p. \end{aligned} \tag{12}$$

Now, let us consider the distance among them:

$$\begin{aligned} D\left(x_b^{r-1,i}, x_b^{r-1,j}\right) &= \left\|x_b^{r-1,i}, x_b^{r-1,j}\right\|_2, \\ D^{r-1} &= \max_{x_b^{r-1,i}, x_b^{r-1,j} \in x_b^{r-1}} D\left(x_b^{r-1,i}, x_b^{r-1,j}\right), \end{aligned} \tag{13}$$

where $\|\cdot\|_2$ is the 2-norm on corresponding search area.

When D^{r-1} reached a small threshold, according to the maximum likelihood estimation, the hypothesis that the real optimal solution is in the area surrounded by these particles was established. So the latter search can be performed around these particles.

In light of this, we can give the condition of DVSA as shown below

$$D^{r-1} < \lambda \cdot \left\|a^{r-1} - b^{r-1}\right\|_2, \quad \lambda \in \left(0, 1/N_p\right]. \tag{14}$$

In other words, if the above equation is satisfied, then change the search area of the next generation of subswarms until the DVSA occurs again. λ can be a fixed number, but more often, it is a parameter that can be changed adaptively according to the results of evolution.

Let us consider the search area after reduction. Note that after the rth reduce, x_i ranges over $[a_i^r, b_i^r]$. Then the upper/lower bounds are defined by the following equation:

$$\begin{aligned} a_i^r &= \min\left\{x_{bi}^{r-1,p}\right\} - \xi \cdot \left(b_i^{r-1} - a_i^{r-1}\right), \\ b_i^r &= \min\left\{x_{bi}^{r-1,p}\right\} + \xi \cdot \left(b_i^{r-1} - a_i^{r-1}\right), \\ & \xi \in (0, 1]. \end{aligned} \tag{15}$$

To guarantee that the new search area is not larger than the previous area, the above equation should be modified as follows:

$$\begin{aligned} a_i^r &= \begin{cases} a_i^{r-1}, & a_i^r < a_i^{r-1}, \\ a_i^r, & \text{otherwise}, \end{cases} \\ b_i^r &= \begin{cases} b_i^{r-1}, & b_i^r < b_i^{r-1}, \\ b_i^r, & \text{otherwise}. \end{cases} \end{aligned} \tag{16}$$

4.3. Policy of Population Scale Adjustment. The computational complexity also relies heavily on the scale of the population of the swarm/subswarm. In general, the more take time about particle evaluation is, the more computation takes place. Hence, under the permission of optimization performance, it is necessary to cut down the population of subswarms.

In this article, we will follow a traditional method called search granularity. Take the particle after the rth reduce for instance, whose ith component x_i ranges over $[a_i^r, b_i^r]$. The distance of this interval can be written as (17) which reflects the refined effort of search. If the distance among the solutions is small, we can say that search granularity is small and vice versa. From the real experience, the bigger the swarm is, the less distance exists among the particles, which also lessen the search granularity:

$$d_i^r = b_i^r - a_i^r. \tag{17}$$

Furthermore, if it is asked that the search granularity on $[a_i^r, b_i^r]$ should be $1/N_{ik}$, the population scale of subswarm can be determined by the below equation:

$$N_i^r = \left\lfloor \prod_{k=1}^{N_d} N_{ik} \cdot d_i^r \right\rfloor, \tag{18}$$

where $\lfloor\cdot\rfloor$ is the floor function. When the search area decreases, the population of the related subswarm also becomes small.

4.4. Theoretical Analysis. In this subsection, an analysis of the convergence of CQPSO-DVSA-LFD is provided. We discuss it from two perspectives, that is, search area and population of swarms.

Firstly, we analyze the variance of interval measure caused by two neighboring reduces. According to the policy of DVSA, it can be described as follows according to (19), (20):

$$b_i^r - a_i^r \le b_i^{r-1} - a_i^{r-1}. \tag{19}$$

Without loss of generality, let

$$b_i^r - a_i^r = k_i^r \cdot \left(b_i^{r-1} - a_i^{r-1}\right), \quad k_i^r \in (0, 1], \tag{20}$$

then

$$\begin{aligned} b_i^r - a_i^r &= k_i^r \cdot \left(b_i^{r-1} - a_i^{r-1}\right) \\ &= k_i^r \cdot k_i^{r-1}\left(b_i^{r-2} - a_i^{r-2}\right) = \cdots = K_i^r \cdot (b_i - a_i), \\ K_i^r &= \prod_{j=1}^{r} k_i^r \le \min\left\{k_i^1, k_i^2, \ldots, k_i^r\right\}. \end{aligned} \tag{21}$$

From formula (21), we can see that when search area varies, the reduced area becomes the k_i^r times of origin area. So when several generations of this procedure happen, the final area could be heavily reduced with the considerable promotion of efficiency.

Secondly, in consideration of swarm populations, we can get the result from

$$\begin{aligned} N_j^r &= \left\lfloor \prod_{i=1}^{N_d} N_{ji} \cdot d_i^r \right\rfloor = \left\lfloor \prod_{j=1}^{N_d} N_{ji} \cdot \left(b_i^r - a_i^r\right) \right\rfloor \\ &= \left\lfloor \prod_{j=1}^{N_d} N_{ji} \cdot K_i^r \left(b_i^r - a_i^r\right) \right\rfloor < \left\lfloor \prod_{j=1}^{N_d} N_{ji} \cdot (b_i - a_i) \right\rfloor. \end{aligned} \tag{22}$$

The above inference shows that as the search area decreases, the related populations of swarms can also be cut down with a certain rate.

5. Lévy Flights Disturbance

The technique of random disturbance is often imported to improve the performance of PSO or QPSO. When QPSO was proposed, the Gaussian and Cauchy probability distribution disturbances have been used to avoid premature convergence. In [17], the random sequences in QPSO were generated using the absolute value of the Gaussian probability distribution with zero mean and unit variance. Based on the characteristic of QPSO, the variables of the global best and mean best positions are mutated with Cauchy distribution, and an adaptive QPSO version was proposed in [14].

In this paper, another random method, Lévy flights, is employed to do this work. Lévy flights, named after the French mathematician Paul Pierre Lévy, are Markov processes. After a large number of steps, the distance from the origin of the random walk tends to a stable distribution. Lévy flights, which can be characterized by an inverse square distribution of step length, may optimize the random search process when targets are scarce and at scarcity of resources. In contrast, Brownian motion is usually suitable for the case when there is a need to locate abundant prey or targets. These traits of two random walks inspired us to improve our swarm intelligence optimization, where Lévy flights can improve the ability of "exploration" while Brownian motion benefits the "exploitation."

Mathematically, Lévy flights are a kind of random walk whose step lengths meet a heavy-tailed Lévy alpha-stable distribution, often in terms of a power-law formula, $L(s) \sim |s|^{-1-\beta}$, where $0 < \beta \leq 2$ is an index. A typical version of Lévy distribution can be defined as [18]

$$L(s,\gamma,\mu) = \begin{cases} \sqrt{\dfrac{\gamma}{2\pi}} \exp\left[-\dfrac{\gamma}{2(s-\mu)}\right] \dfrac{1}{(s-\mu)^{3/2}}, & 0 < \mu < s < \infty; \\ 0, & s \leq 0. \end{cases} \tag{23}$$

At the change of β, this can evolve into one of Lévy distributions, normal distributions; and Cauchy distributions.

Taking the 2D-Lévy flights for instance, the steps follow a Lévy distribution as in Figure 3(b), while the directions of its movements meet a uniform distribution as in Figure 3(a). As shown in Figure 3(c), an instance of the trajectory of 500 steps of random walks is obeying a Lévy distribution. Note that the Lévy flights are often more efficient in exploring unknown and large-scale search space than Brownian walks. One reason for this argument is that the variance of Lévy flights $\delta^2(t) \sim t^{3-\beta}$, $1 \leq \beta \leq 2$ increases faster than that of Brownian random walks, that is, $\delta^2(t) \sim t$. Also, compared to Gaussian distribution, Lévy distribution is advantageous since the probability of returning to a previously visited site is smaller than that for a Gaussian distribution, irrespective of the value of μ chosen.

From the update strategy of CQPSO-DVSA-LFD, we can draw a conclusion that all particles in CQPSO-DVSA-LFD will converge to a common point, leaving the diversity of the population extremely low and particles stagnated without further search before the iterations are over. To overcome the problem, we exert a disturbance generated by Lévy flights on the mean best position, global best position, and electoral best position when the swarm is evolving as shown in the following equation (24). To the local attractor, the hop steps in Lévy flights promise the random traversal in the search space. However, to the global and electoral best locations, they only need a slightly disturbance; that is, the angles meet a uniform distribution, to exploit the particles nearby

$$\begin{aligned} C_d' &= C_d + \varepsilon_3 \times \mathrm{Step}_{\mathrm{Lévy}}, \\ P_{\mathrm{gd}}^{\mathrm{best}'} &= P_{\mathrm{gd}}^{\mathrm{best}} + \varepsilon_1 \times \mathrm{Angle}_{\mathrm{Lévy}}, \\ P_{\mathrm{cgd}}^{\mathrm{best}'} &= P_{\mathrm{cgd}}^{\mathrm{best}} + \varepsilon_2 \times \mathrm{Angle}_{\mathrm{Lévy}}, \end{aligned} \tag{24}$$

where ε_1, ε_2, and ε_3 are a prespecified parameter, $\mathrm{Step}_{\mathrm{Lévy}}$ is a number in a sequence by Lévy flights, angle is the angles of directions in Lévy flights.

6. Experimental Studies

6.1. Experiments on Continuous Optimization Benchmarks. To study the search behavior and its performance of CQPSO-DVSA-LFD with other versions of PSO, such as plain PSO, CPSO, and CQPSO, some typical benchmark functions of continuous optimization are selected as the examples [19, 20].

Rastrigin's function is frequently used as a test function to test the performance of optimization algorithms. Based on Sphere function, it uses cosine function to generate lots of local optimal points. It is a complex multimodal function, and optimization falls into the local optimum easily. Griewank function is a spin, inseparable, variable-dimension, multi-mode function as shown in Figure 4. At the increase of its dimension, the scope of local optimum gets narrower so that searching global optimum becomes easy relatively. Therefore, for Griewank function, it is harder to get solution in low dimension than in high dimension. Michalewicz function is a multimodal function with parameter m which changes the steepness of valleys. The Lévy number 8 function has one global minimum and, approximately, 125 local minima.

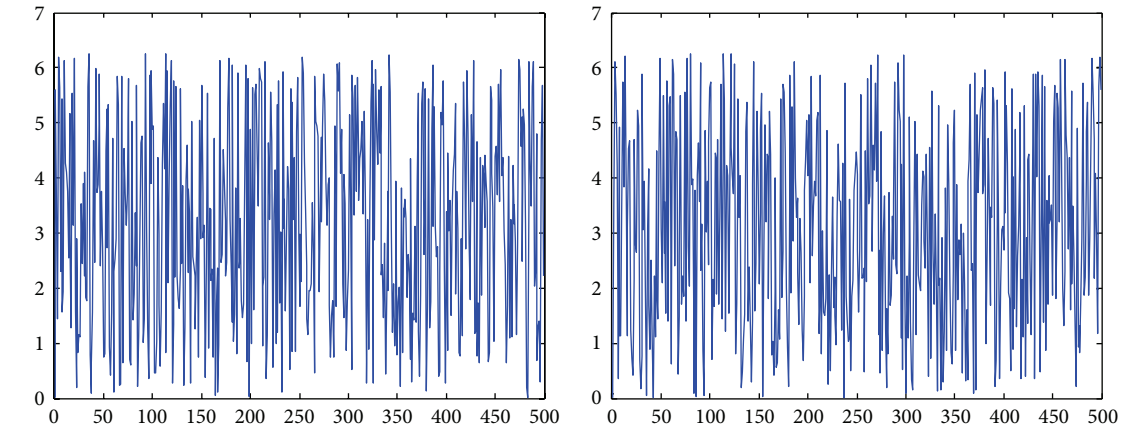
Computational results of variants of PSO used in the paper are qualitatively ranked in Table 1. From it, we can clearly get that the proposed CQPSO-DVSA-LFD algorithm performed greatly better than the plain PSO and QPSO. Also, compared to the basic Cooperative PSO (CPSO), the convergence property has been enhanced by the proposed techniques in the paper.

In Figure 5, the black cycles denote the distribution of particles of 2-d Griewank function in QPSO under DVSA and LFD, while the red ones express that of CQPSO-DVSA-LFD with only two cooperative subswarms. It can be clearly seen that in CQPSO-DVSA-LFD, the search area in each generation of iteration is reduced dynamically into the

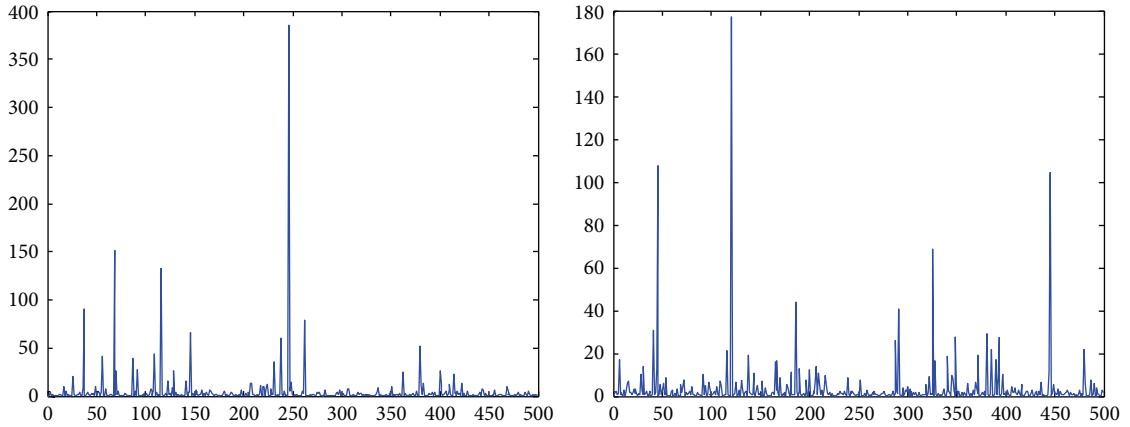
Table 1: Results of functions optimization in benchmark.

Functions	PSO			QPSO			CQPSO			CQPSO-DVSA-LFD		
	Minimum	Maximum	Average	Minimum	Maximum	Average	Minimum	Maximum	Average	Minimum	Maximum	Average
Rastrigin's	$6.12E-06$	$9.00E+00$	$4.67E+00$	$3.01E+00$	$8.90E+01$	$4.91E+01$	$\mathbf{2.50E+01}$	$1.25E+02$	$7.52E+01$	$\mathbf{2.50E+01}$	$\mathbf{2.50E+01}$	$\mathbf{2.50E+01}$
Griewank	$9.87E-03$	$1.50E-01$	$5.04E-02$	$9.12E-06$	$8.68E-02$	$3.86E-04$	$2.58E-06$	$4.95E-02$	$1.56E-02$	**0**	$8.04E-10$	$6.83E-11$
Michalewicz	−9.284715	−7.846484	−8.387755	−9.375576	−8.195449	−9.006959	−9.613477	−8.394369	−9.237160	**−9.660151**	**−9.660151**	**−9.660151**
Lévy	0.6663	4.6048	2.3496	0.1019	11.5187	2.2827	$3.27E-05$	$2.94E-04$	$1.48E-04$	$\mathbf{2.01E-05}$	$6.37E-04$	$2.38E-04$

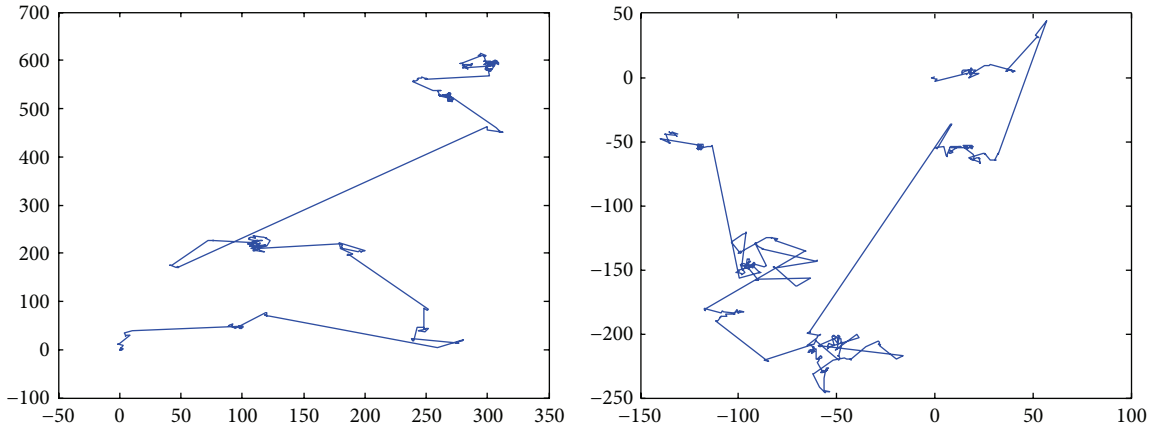
The bold values denote the stable approximate optimal solution compared to other algorithms.



(a) Angle values of 500 random turns in Lévy flights



(b) Step lengths of 500 random walks in Lévy flights



(c) Distance of 500 random walks in Lévy flights

Figure 3: 2D Lévy flights in 500 steps.

potential rectangles along two red lines on horizontal/vertical directions. In addition, we can also find that the populations of the latter generations has been reduced obviously, which means the lower computational complexity meanwhile.

Moreover, the convergence ability is also investigated in our experiment. Figure 3 illustrates the typical convergence of PSO, CPSO, QPSO, CQPSO, and CQPSO-DVSA-LFD on the benchmark Michalewicz function. From the figure, it can be seen that the varying curves of objective values using the CQPSO-DVSA-LFD descend much faster than that when using plain PSO and QPSO. In addition, the fitness values descent to lower level by using CQPSO-DVSA-LFD than CPSO due to the different mechanisms of simulated annealing and DVSA.

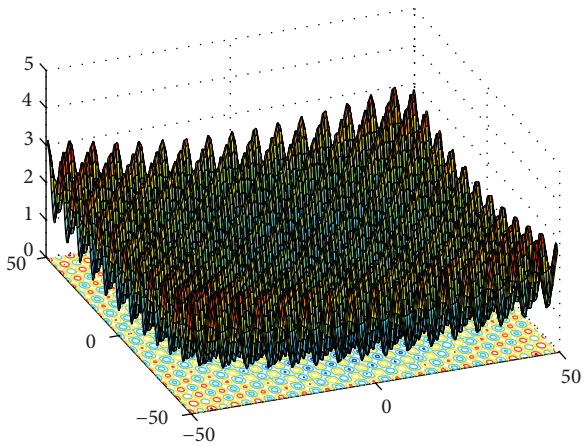


Figure 4: Surface landscape of Griewank function.

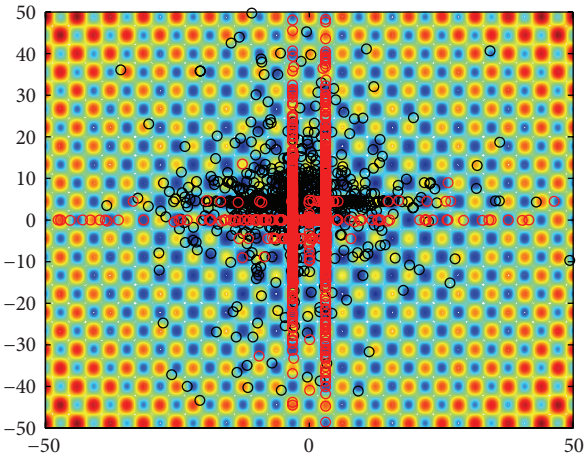


Figure 5: Particles in QPSO and CQPSO-DVSA-LFD.

From Figure 6, the results of the experiments indicated that the proposed CQPSO-DVSA-LFD can lead to more efficiency and stability than plain PSO, QPSO, CPSO, and CQPSO.

6.2. Experiments on Combinatorial Optimization Problem. For the combinatorial optimization problem, we choose job-shop scheduling problem (JSSP) to test the performance of our algorithm. Job-shop scheduling problem (JSSP) is well known that it is in the family of NP-hard and NP-complete and has proven to be an impossible task for human schedulers. In the job-shop scheduling problem, finite jobs are to be processed by fix-numbered machines. Each job consists of a predetermined sequence of task operations, each of which needs to be processed without preemption for a given period of time on a given machine. But, tasks of the same job could not be processed concurrently and each job must be on each machine exactly once. Moreover, each operation cannot be commenced until the processing is completed, if the precedent operation is still being processed. A schedule is an assignment of operations to time slots on a machine. The makespan is the maximum completion time of

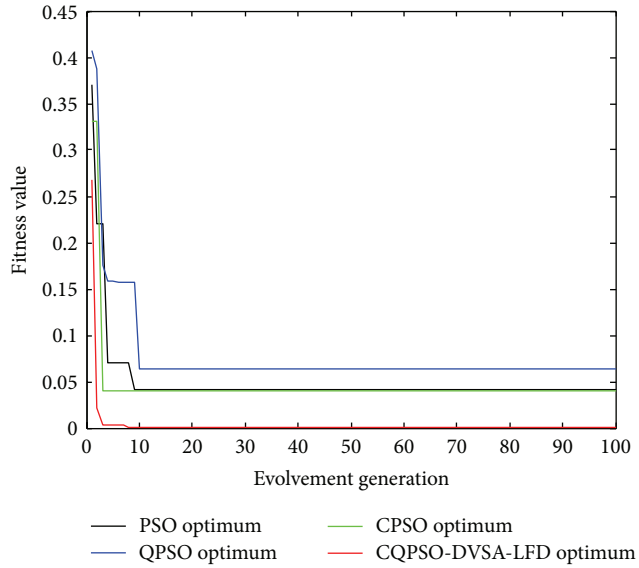


Figure 6: Evolution curves of Griewank function.

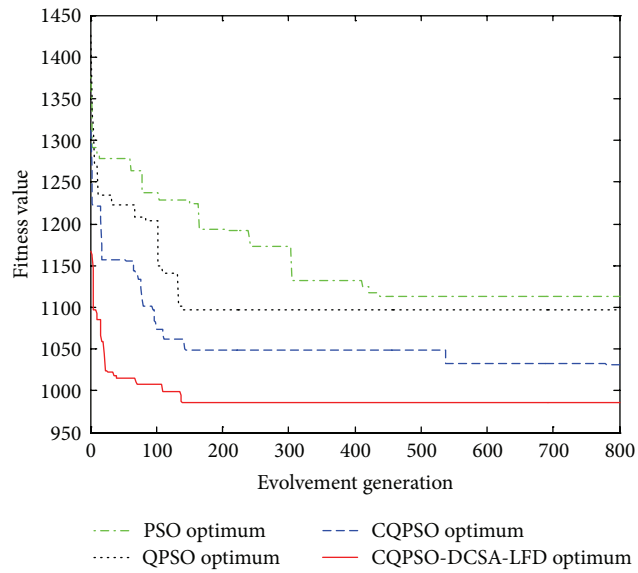


Figure 7: Evolution curves of JSSP.

the jobs and the objective of the JSSP is to find a schedule that minimizes the makespan.

The experiments on JSSP are performed on $6 * 6$, $10 * 10$, and $20 * 5$ instances, respectively, and its convergence efficiency and GANT are shown in Figures 7 and 8. We can see that the convergence rate of CQPSO-DVSA-LFD is clearly faster than other PSO algorithms from our simulation solution.

7. Conclusions and Future Work

In this paper, we proposed a CQPSO-DVSA-LFD algorithm which is a combination of QPSO, cooperative mechanisms which are used to find better particles in shorter time, and

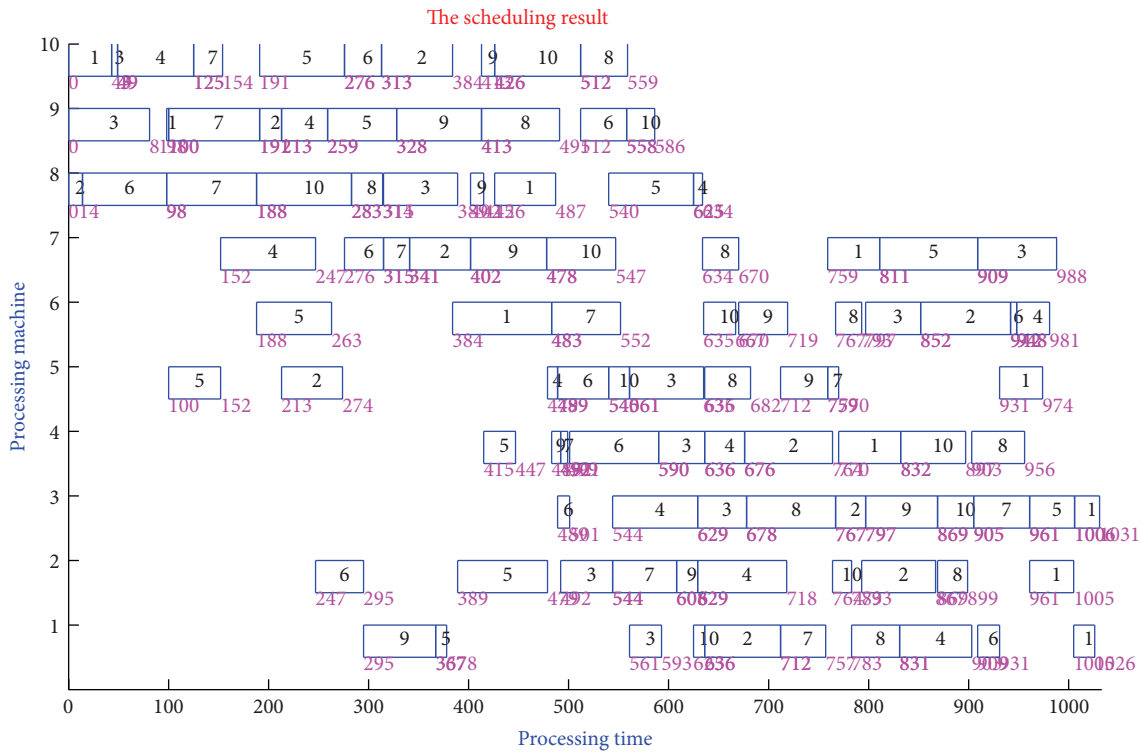


Figure 8: GANT graph of 6 ∗ 6 JSSP generated by CQPSO-DVSA-LFD.

two ways to reduce the search space. One is called Dynamic Varying Search Area (DVSA), which takes charge of limiting the ranges of particles' activity; the other is cooperative strategy, which divides the candidate solution vector into small subswarms. Moreover, to help escape from local optima, a disturbance generated by Lévy flights is embedded as a hybrid strategy. Computational results and comparisons on both continuous optimization benchmarks and JSSP problem show that it outperforms other related algorithms.

Future research may include a further investigation of the algorithm to solve other problems. It is also worthwhile to tune Lévy flights disturbance approaches and analyse the performance about the improvement by this random disturbance. The parameter setting optimally and adjustment of algorithm termination conditions are also one of our focuses.

Conflict of Interests

The author declares that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

The research was supported by the Natural Science Foundation of Anhui Province (no. 1308085QF103), Natural Science Foundation of Educational Government of Anhui Province (no. KJ2013B073), Science, and Technology Plan Project of Chuzhou City (no. 201236), Talent Introduction Special Fund of Anhui Science and Technology University (no. ZRC2011304). The author sincerely thanks the anonymous reviewers for their constructive comments and helpful suggestions.

References

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

[2] Z. Cui and X. Cai, "Integral particle swarm optimization with dispersed accelerator information," *Fundamenta Informaticae*, vol. 95, no. 4, pp. 427–447, 2009.

[3] I. Budinská, T. Kasanický, and J. Zelenka, "Production planning and scheduling by means of artificial immune systems and particle swarm optimisation algorithms," *International Journal of Bio-Inspired Computation*, vol. 4, no. 4, pp. 237–248, 2012.

[4] Z. Cui, X. Cai, J. Zeng, and Y. Yin, "PID-controlled particle swarm optimization," *Journal of Multiple-Valued Logic and Soft Computing*, vol. 16, no. 6, pp. 585–609, 2010.

[5] C. Priya and P. Lakshmi, "Particle swarm optimisation applied to real time control of spherical tank system," *International Journal of Bio-Inspired Computation*, vol. 4, no. 4, pp. 206–216, 2012.

[6] H. Derrar, M. Ahmed-Nacer, and O. Boussaid, "Particle swarm optimisation for data warehouse logical design," *International Journal of Bio-Inspired Computation*, vol. 4, no. 4, pp. 249–257, 2012.

[7] A. Lazinica, *Particle Swarm Optimization*, IN-TECH, 2009.

[8] X. S. Yang, Z. H. Cui, R. B. Xiao, A. H. Gandomi, and M. Karamanoglu, *Swarm Intelligence and Bio-Inspired Computation: Theory and Applications*, Elsevier, Waltham, Mass, USA, 2013.

[9] 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.

[10] 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.

[11] J. Sun, B. Feng, and W. Xu, "Particle swarm optimization with particles having quantum behavior," in *Proceedings of the Congress on Evolutionary Computation (CEC '04)*, pp. 325–331, June 2004.

[12] J. Sun, W. Xu, and B. Feng, "A global search strategy of Quantum-behaved Particle Swarm Optimization," in *Proceedings of the IEEE Conference on Cybernetics and Intelligent Systems*, pp. 111–116, December 2004.

[13] N. Keerativuttitumrong, N. Chaiyaratana, and V. Varavithya, "Multi-objective co-operative co-evolutionary genetic algorithm," in *Parallel Problem Solving from Nature-PPSN VII*, pp. 288–297, Springer, Berlin, Germany, 2002.

[14] J. Liu, J. Sun, and W. B. Xu, "Design IIR digital filters using quantum-behaved particle swarm optimization," in *Advances in Natural Computation*, vol. 4222 of *Lecture Notes in Computer Science*, pp. 637–640, 2006.

[15] D. Li and N. Deng, "An electoral quantum-behaved PSO with simulated annealing and Gaussian disturbance for permutation flow shop scheduling," *Journal of Information & Computational Science*, vol. 9, 2012.

[16] A. El Dor, M. Clerc, and P. Siarry, "A multi-swarm PSO using charged particles in a partitioned search space for continuous optimization," *Computational Optimization and Applications*, vol. 11, pp. 1–25, 2011.

[17] L. S. Coelho, "Novel Gaussian quantum-behaved particle swarm optimiser applied to electromagnetic design," *IET Science, Measurement and Technology*, vol. 1, no. 5, pp. 290–294, 2007.

[18] X. S. Yang, *Engineering Optimization: An Introduction with Metaheuristic Applications*, John Wiley & Sons, 2010.

[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. R. Hedar, "Test functions for unconstrained global optimization," http://www-optima.amp.i.kyoto-u.ac.jp/member/student/hedar/Hedar_files/TestGO.htm.